

# Adminca.com

The Interface for your Backend Data

HCDE Capstone, Winter 2016

Lukas Eiermann  
Frank Kaiyuan Xu  
Jordan Vincent

# TABLE OF CONTENTS

## INTRODUCTION

- Summary ..... 4
- Meet the Team ..... 5
- Problem Space ..... 6
- Design Question ..... 7
- Project Goal ..... 8
- Process ..... 9

## INVESTIGATION

- Research Overview ..... 11
- Research Insights ..... 12
- Design Principles and Implications ..... 14

## FINAL DESIGN OVERVIEW

- The Idea ..... 16
- Admin Workflow ..... 17
- Contributor Workflow ..... 23

## DESIGN PROCESS & EVOLUTION

- Iterative Design Process ..... 27
- Design Evolution ..... 28
- Challenges ..... 34

## IMPACT & OUTLOOK

- Growth ..... 36
- User Feedback ..... 37
- Final Reflection ..... 38
- Future Directions ..... 40
- Thank you ..... 41

# INTRODUCTION

---



# SUMMARY

Mobile app publishers need to manage app content and user information to keep the data up to date. An admin panel is the interface that allows them to view and manage this backend data. But admin panels are broken! Developers waste weeks setting it up, and the interface is often so complicated that nobody wants to use it.

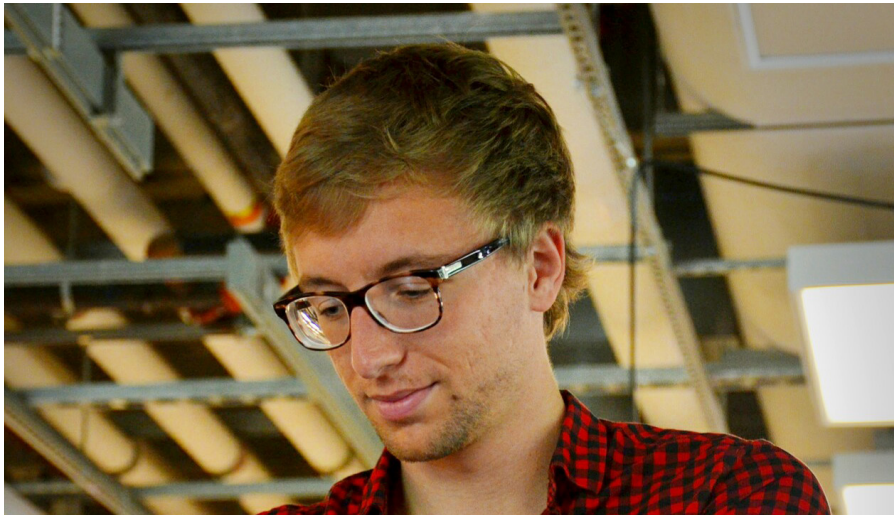
To solve this problem we created Adminca—the interface for your backend data. Adminca provides an Admin Panel as a Service which allows developers to get a powerful admin panel without writing a single line of code. Creating and customizing a panel takes seconds instead of weeks. It comes with roles and permissions for the team or clients that will manage the data. Adminca was designed specifically for these non-technical users so they can view and manage their data intuitively.

We designed and built Adminca over the last five months following the user-centered design process. Two months ago we launched Adminca to the public, and so far we welcomed over 400 users.



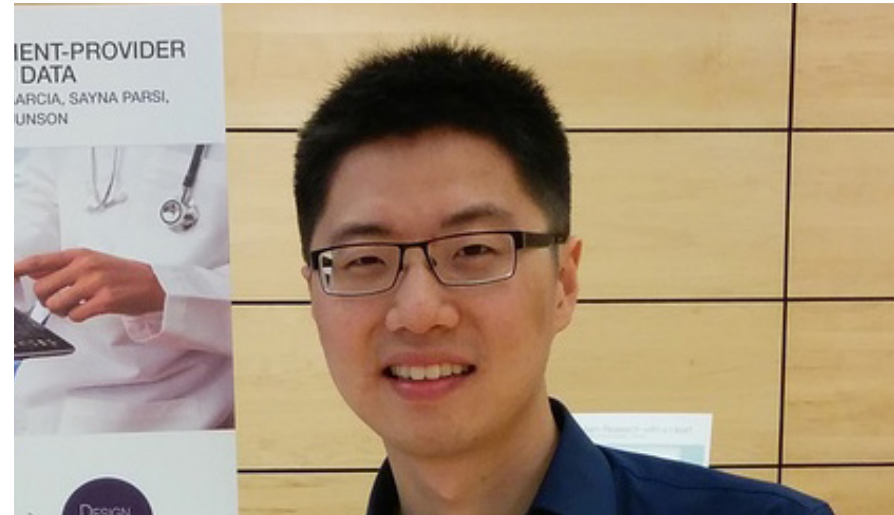


# MEET THE TEAM



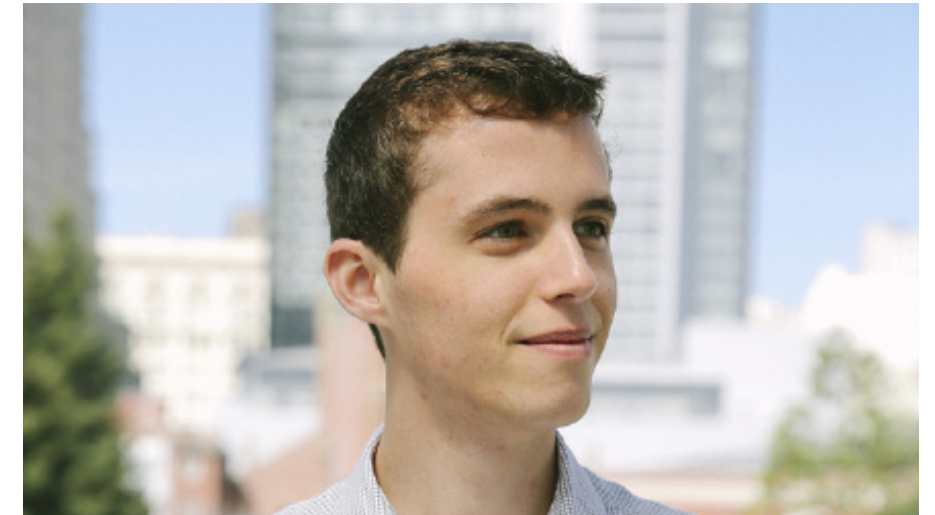
## Lukas Eiermann

Lukas is a UX Designer and is passionate about creating products that simplify the complex and serve the user. He holds a Bachelor's and Master's degree in Engineering Management from the Karlsruhe Institute of Technology, Germany. He is interested in solving challenging design problems that translate into meaningful digital experiences.



## Frank Kaiyuan Xu

Frank is a UX researcher, striving to understand people and advocate for their needs. With his background in psychology, he has years of experience in research design, data analysis, and finding synthesis. He has worked in multiple Agile teams, and improved the user experience for web, mobile, and hardware products.



## Jordan Vincent

Jordan is a UX Designer with a strong background in Software Engineering. He got his Master's degree in Computer Science at INSA of Lyon, France. He has a passion for innovation and entrepreneurship that led him to join startups in France and San Francisco. He focusses on using technology and design to empowers humans.

# PROBLEM SPACE

## What is Backend Data?

*Data generated through or for mobile/web apps that is stored in a database.*

Most mobile or web applications do not save all their data locally. The app communicates with a backend server to save or access data from the backend database.

### Backend data can be:

- User generated (e.g. a user's profile picture)
- System generated (e.g. a user's current location)
- Publisher generated (e.g. a new article)

## What is Backend Data?

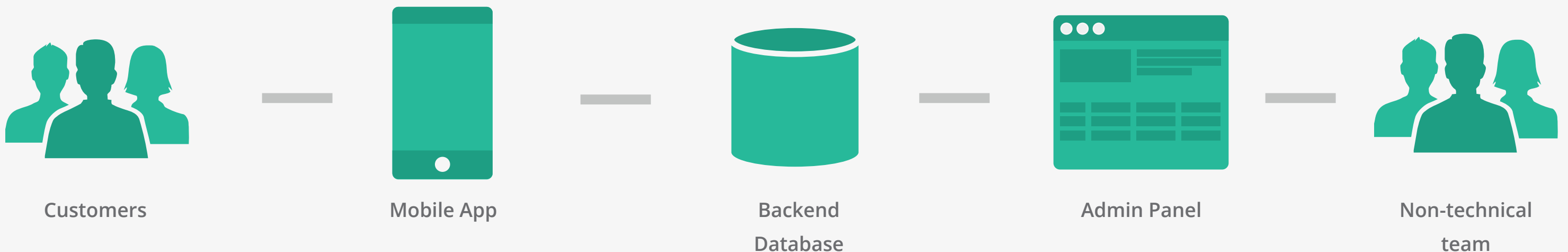
*An internal tool allowing the publisher's team to view and manage all the backend data relevant to the app.*

The team that owns the app needs a way to access the app data. Developers can access the backend database directly but non-technical users need a tool to access the data: an admin panel.

### An Admin Panel can be used to:

- Create new data
- View and analyze existing data
- Edit or delete existing data

▼ Data, from the end-users to the internal team



## DESIGN QUESTION

---

# How can we improve the experience of creating and using administrative interfaces to manage backend data?

All three of us have used, built and maintained admin panels in the past, and we believe it's fair to say that **admin panels are broken**.

### Developer Perspective

Most admin panels are built from scratch which means developers are basically reinventing the wheel. Developers waste time building and maintaining a tool that it is not core to the actual product or business.

### Non-technical User Perspective

Non-technical users have to work with a poorly designed admin panel that was built on the premise of not being important since it is not customer facing. The poorly designed interface often lowers the staff efficiency.

# PROJECT GOAL

## Focus on a specific backend

There is a wide variety of backends and the admin panel problem exists for all of them. For the scope of this project we decided to **focus on a specific backend called Parse** which is backed by Facebook.

We picked Parse because with over 400,000 active developers, it is the largest Mobile Backend as a Service provider. Parse was particularly interesting because it is mostly used by developers who focus on mobile apps. Creating an admin panel is even more time-consuming for them because they are less familiar with web technologies needed for admin panels.

## Build and Launch an actual product

A frustrating part of design projects can be that they don't go beyond the prototyping phase. Our goal for this project was to **include the implementation phase** and build a product. Building, launching and marketing a product would allow us to test and iterate our idea with actual customers in a real world scenario.

Parse community  
event in Seattle



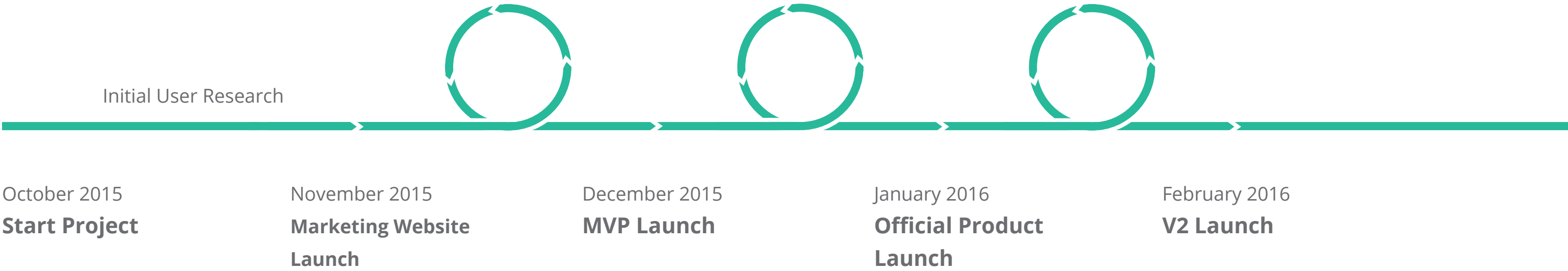
More than 400,000 developers are  
building apps with Parse.  
In the last year, more than 85% of the top apps  
on Parse were built outside of the US.



# PROCESS

After our initial research phase we started our rapid iterative design process. During each iteration, we went through the process, and improved Adminca based on the findings from the previous round.

The process book describes Adminca as of March 2016.



# INVESTIGATION

---

# RESEARCH OVERVIEW

We applied three research methods to better understand the Parse app backend data management problem space and user needs.

## 27

### THREADS ANALYSED

We started with secondary research to learn what people asked and demanded online regarding admin panels for Parse.

We found 27 questions and comments posted in recent years, which were either related to improving the current Parse data browser, or asked about building a separate tool for non-technical users. This helped us to identify different user groups and individual needs.

## 14

### INTERVIEWS

We conducted three rounds of interviews with different groups of users: Developers building apps on Parse, Mobile app development agencies and Admin panel template users.

Our interview findings showed us a number of pain points for current admin panel developers and users. We created three personas for our product, and got a better and clearer understanding of our target users.

## 13

### COMPETITORS ANALYSED

We reviewed 13 existing admin panel solutions and evaluated them by comparing them along 28 features and 16 design categories.

We highlighted the strengths and usability issues of the competitors, and gained a clearer understanding of existing solutions for admin panels.



# RESEARCH INSIGHTS

## Developers

*“Parse offers the data browser for data management, which is quite easy to use for me. But it’s too complicated for my client. Also, if I invite my client to the Parse account, he may accidentally delete the whole class of data.”*

— Parse developer building apps for clients

### Goals

- Develop and maintain an admin panel for team or clients to access and manage data.
- Make sure team/clients know how to manage data on Parse or through the admin panel.
- Backup the database to ensure the app doesn’t crash if team/clients mistakenly delete things.

### Pain points

- Designing and building admin panels is usually not a developer’s expertise or passion.
- Depending on the complexity of data management needs, building an admin panel can take weeks even with admin panel templates.
- If the team/clients manage data through the Parse backend, they may accidentally delete crucial data or even cloud code saved in the Parse.

# RESEARCH INSIGHTS

## Non-technical people

*“I don’t know much about app development, but I do want access to the backend data. I need an admin tool so that I don’t have to bother my devs every time.”*

— Co-founder of a Seattle startup with a business background

### Goals

- View data in a straightforward layout
- Manage data quickly and easily without asking developers for help.
- Export data for business analytics or marketing campaign.

### Pain points

- Parse data browser is developer-oriented, but hard for non-technical users to view or manage.
- Non-technical users are often afraid of changing or updating data in the Parse data browser because they don’t want to mess anything up in the app.
- Parse only export data in JSON format, which is hard for non-technical users to use. It’s also not easy to convert JSON data to CSV format.

# DESIGN PRINCIPLES AND IMPLICATIONS

Based on our findings, we created the three S. We used them as guidance for our design decisions and as company values for Adminca as a company.

## Simple

### EFFORTLESS SETUP + INTUITIVE INTERFACE

Save developers weeks of time, and streamline the admin panel setup process. Provide an intuitive interface that non-technical users can get their hands on without training.

## Secure

### DATA PERMISSION + ENCRYPTED CREDENTIALS

Empower developers to manage data permission to ensure data integrity. Use highest encryption and security measures for Adminca to protect customer's data.

## Smart

### POWERFUL FEATURES + SMART DEFAULTS

Provide solutions for diverse data management problems with a comprehensive feature set. Understand what users want to achieve and provide them with smart default that make their live easier.



# FINAL DESIGN OVERVIEW

---

## THE IDEA

---

# Adminca is an Admin Panel as a Service

### **For Developers:**

#### **Admin View**

Adminca enables developers to easily set up a fully customizable admin panel where they can then invite their collaborators.

### **For Non-technical Users:**

#### **Contributor View**

Adminca empowers non-technical users to intuitively view and manage their data.

# ADMIN WORKFLOW

A developer has the role of an admin in Adminca. After signing up he is guided through the admin workflow to set up and modify the admin panel for the contributor.



**Create a  
Panel**



**View the  
Panel**



**Customize  
the Panel**



**Invite  
Collaborators**



# ADMIN WORKFLOW

## Create a Panel



### A Name & Keys

Assign a name and enter the app credentials.

### B Tooltips

Help users find their app credentials on Parse and explain how Adminca uses and protects the credentials

### C Create

Once everything is entered correctly, the admin panel is just 5 seconds away.

A screenshot of a web application window titled "Create an Admin Panel". The window has a grey title bar with red, orange, and green window control buttons. The form is white and contains the following elements: a title "Create an Admin Panel", a label "Admin Panel Name" above a text input field with placeholder text "Admin Panel Name" (marked with a teal 'A'), a label "Enter your Parse App Credentials. ?" (marked with a teal 'B') above the "Application ID" label, a text input field for "Application ID" with placeholder text "Application ID", a label "Master Key ?" above a text input field for "Master Key" with placeholder text "Master Key", and two buttons at the bottom: a "Cancel" button and a dark blue "Create" button (marked with a teal 'C').

# ADMIN WORKFLOW

## View the Panel



- A Panels List**  
Switch between multiple panels or create new panel.
- B Data Table**  
Table view of data in this class.
- C Edit Class/Field**  
Customize contributor permissions and define data access for the class and its fields.
- D Class Navigation**  
Open other classes within this panel.
- E Panel Settings**  
Modify general panel properties, and invite other admins or contributors to this panel.

Yoga Poses

Members

Poses

Orders

Users

Transactions

Schedules

Studios

Instructors

Panel Settings

Yoga Seattle

Poses

Pose Name

Search...

Search

Pose Name	Picture	Pose Description	Pose Difficulty	Tags
Ardha-Chandrasana		Ardha Chandrasana or Hal	Easy	back breath upper body
Utkatasana		Utkatasana, Chair Pose, Fi	Easy	leg feet arm back
Garurasana		Garurasana or Eagle Pose.	Medium	breath leg knee arm
Dandayamana-Janushiras		Standing Head to Knee Po	Hard	back leg feet neck head
Dadayamana-Dhanurasan		Like camel posture, standi	Extreme	hip leg arm back
Svanasana		Svanasana, also known as t	Medium	leg arm back feet
Dandayamana-Bibhaktapa		Dandayamana-Bibhaktapa	Hard	hip back shoulder arm
Trikanasana		Trikanasana is also known	Medium	leg arm shoulder head
Dandayamana-Bibhaktapa		Dandayamana Bibhaktapa	Hard	back arm upper body feet
Tadasana		Mountain Pose: Step-by-St	Medium	leg knee arm
Padangustasana		Bikram describes this post	Extreme	leg hip knee feet
Savasana		Savasana it's essential that	Easy	relax breath full body

17 records

Refresh

Export

# ADMIN WORKFLOW

## Customize the Panel — Class Settings

- A Class Settings**  
Open class settings page.
- B Visibility**  
Hide or show this class to contributors.
- C Label and Name**  
Replace Parse class name with easy-to-understand label.
- D Name Field Settings**  
Change the name field of the class, which is used to identify records in this class. It is also the first column in the table, and the default search field.
- E User Permissions**  
Change contributor permissions for this class.

Yoga Poses

Members

**Poses** Edit

Orders

Users

Transactions

Schedules

Studios

Instructors

Panel Settings

Yoga Seattle

Poses

Edit class

Class settings **A**

Edit fields Create Field

Pose Name Name Field

Picture

Pose Description

Pose Difficulty

Tags

Offered?

Created At

Updated At

ACL

objectId ID Field

Instructor

Belongstoshop

Class Settings

General

**B** Visibility

☒ Show class ☐ Hide class

**C** Label

Poses

Class Name

Products

Description

Properties

**D** Name Field

Pose Name

Allow users to

**E** ☒ Create new records in this class

☒ Edit existing records in this class

☒ Delete existing records in this class

Delete Class

Cancel

Save



# ADMIN WORKFLOW

## Customize the Panel — Field Settings



- A Create New Fields**  
Directly create new fields in Adminca without going to Parse.
- B Change Field Order**  
Drag field up and down to reorder the fields.
- C Visibility**  
Hide or show this field to contributors.
- D Modify Data Format**  
Define data format for each field (e.g. Email Address), add a default value or create predefined values (e.g. 50 US States).
- E User Permissions**  
Change contributor permissions for this field.

Yoga Poses

Members

Poses

Orders

Users

Transactions

Schedules

Studios

Instructors

Panel Settings

Yoga Seattle

Poses

Edit class

Class settings

Edit fields

Pose Name

Picture

Pose Description

Pose Difficulty

Tags

Offered?

Created At

Updated At

ACL

objectId

Belongstoshop

Instructor

Pose Difficulty Settings

General

Visibility

Label

Field Name

Type and format

Type

Format

Predefined values

Properties

Allow users to

Delete Field

Cancel

Save

# ADMIN WORKFLOW

## Invite Collaborators



- A Change Panel Name**  
Directly create new fields in Adminca without going to Parse.
- B Sync**  
Drag field up and down to reorder the fields.
- C Create New Class**  
Hide or show this field to contributors.
- D Invite Admin and Contributor**  
Invite the team and clients to the panel. Admins can edit the panel settings while contributors can only view and manage the data.
- E Delete the Panel**  
Delete this panel from Adminca.

Yoga Poses

Members

Poses

Orders

Users

Transactions

Schedules

Studios

Instructors

Panel Settings

Yoga Seattle

Panel Settings

Name

A

Yoga Poses

Rename

Parse Connector

B

Sync Schema

Edit

New Class

C

Create new Class

Creates a new class in Parse too!

Admins & Contributors

Admins can edit all project settings. Contributors can only view and edit data that you defined.

Admin

Yoga Seattle

Edit

×

Contributor

Yoga Seattle

Edit

×

Contributor

eva@seattle-yoga.com

Edit

Invitation pending

×

D

Add Admin

Add Contributor

Delete Panel

E

Delete Panel

# CONTRIBUTOR WORKFLOW

A non-technical user has the role of a contributor in Adminca. After being invited by the admin he is guided through the contributor workflow that allows him to view and manage data.



**View  
Data**



**Manage  
Data**

# CONTRIBUTOR WORKFLOW

## View Data

1 — 2

### A Create

Create a new record in this class.

### B Search

Do a fuzzy search on any field in the table and combine multiple criterias.

### C Filter and Sort

Filter by a specific value or date range. Sort by clicking on the header to sort of a field.

### D Detail View

Click anywhere in the table area to open the details.

### E Account

Update email and password or log out. Access the help section.

### F Export

Download the current view as CSV.

Yoga Poses

Members

Poses

Orders

Users

Transactions

Schedules

Studios

Instructors

Panel Settings

Yoga Seattle

E

Poses

A

Create Pose

Pose Name

Search...

B

Search

Standing - Pose Description










×

asana - Pose Name

×

desc - Pose Description

×

Pose Name	Picture	Pose Description	Pose Difficulty	Tags
Utkatasana		Utkatasana, Chair Pose, Fi	Easy	<div>back</div>
Trikanasana		Trikanasana is also known	Medium	<div>shoulder</div> <div>head</div>
Dandayamana-Janushiras		Standing Head to Knee Po	Hard	<div>back</div> <div>leg</div> <div>feet</div> <div>neck</div> <div>head</div>
Tadasana		Mountain Pose: Step-by-Si	Medium	<div>leg</div> <div>knee</div> <div>arm</div>
Dadayamana-Dhanurasana		Like camel posture, standi	Extreme	<div>hip</div> <div>leg</div> <div>arm</div> <div>back</div>
Garurasana		Garurasana or Eagle Pose.	Medium	<div>breath</div> <div>leg</div> <div>knee</div> <div>arm</div>
Dandayamana-Bibhaktapa		Dandayamana-Bibhaktapa	Hard	<div>hip</div> <div>back</div> <div>shoulder</div> <div>arm</div>
Dandayamana-Bibhaktapa		Dandayamana Bibhaktapa	Hard	<div>back</div> <div>arm</div> <div>upper body</div> <div>feet</div>
Ardha-Chandrasana		Ardha Chandrasana or Hal	Easy	<div>back</div> <div>breath</div> <div>upper body</div>

9 records

F

Refresh

Export

# CONTRIBUTOR WORKFLOW

## Edit Data

1 — 2

- A Duplicate**  
Duplicate the current record to the class.
- C Read-only fields**  
Some fields are read only, so contributors cannot edit it.
- B Editable Fields**  
Update fields that are editable.
- D Relational Data**  
Field linked to a record in another class. Contributors can easily see and search by the name of the record rather than just the object ID.
- E Open a Record**  
Open another record by clicking on the row.
- F Delete**  
Delete the record on Parse.

Yoga Poses

Members

Poses

Orders

Users

Transactions

Schedules

Studios

Instructors

Panel Settings

Yoga Seattle











Poses

Search by Pose Description

Search

2 filters

sorted

Pose Name	Picture
Ulkatasana	
Trikanasana	
Svanasana	
Dandayamana-Janushiras	
Tadasana	
Dadayamana-Dhanurasan	
Garurasana	
Dandayamana-Bibhaktapa	
Dandayamana-Bibhaktapa	
Ardha-Chandrasana	

Svanasana Pose

created 4 months ago - updated a few seconds ago

A

Duplicate


×

Pose Name\*

Svanasana

B

Picture



Choose File

Delete

Pose Difficulty

Medium

C

×

▼

Studio

Ravenna Park (Oz4Q4aj2Xu)

D

view

×

▼

Tags

leg

×

arm

×

back

×

feet

×

Delete

F

Cancel

Save



# DESIGN PROCESS & EVOLUTION

---

# ITERATIVE DESIGN PROCESS

## Ideation

Based on our initial user research findings, each of us sketched ideas individually to meet the developer and non-technical users' needs. The sketches helped us to ideate different workflows and layouts for each individual view.

## Protootyping

We consolidate our initial designs by combining the best aspects of individual sketches. The wireframe we created together enabled us to share a shared vision for each view or feature. It also served as the baseline for later iterations.

## Implementation

We set up connectable and interactive prototypes by leveraging Ruby on Rails and EmberJS technology. This architecture allowed us to be very flexible and to support more services than the Parse data browser.

## Usability Testing

During our project, we have been constantly doing usability testings with our users or participants recruited through different channels. Based on our user feedback, we kept iterating on the design to fix usability issues as well as adding new features.

# DESIGN EVOLUTION

## Detail View

Through competitive analysis and user interviews, we discovered that viewing and editing details of a record (one row) is the main task for non-technical users. Unfortunately, this was also the most problematic interaction with various usability problems.

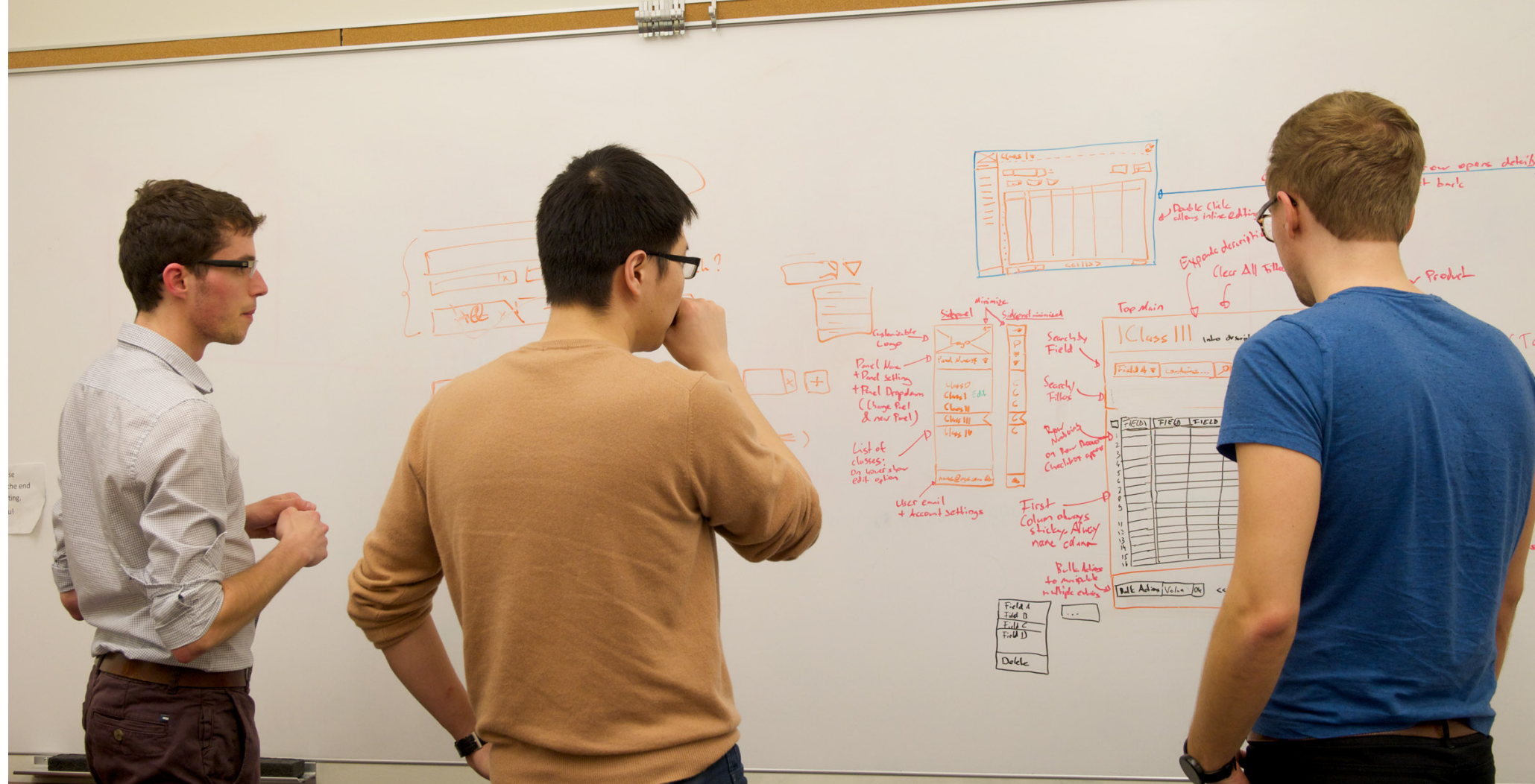
### Limitations of current solutions

**No Detail View.** The Parse data browser and some admin panels only support inline editing but don't allow users to view a record as whole.

**Different pages for View and Edit.** Users have to decide if they want to just view a record or edit it.

**Opening is difficult.** In most admin panels the icon to open the detail view is extremely small and located far away from the name of the record.

**Loss of context.** Once the detail view is open, the relationship to the original table is lost. .



### Our approach

Adminca provides a detail view that can be opened from anywhere in table, because the whole row is a click target.

The Detail View opens in a side panel. This way the user keeps the context of the original table and can easily navigate to other records

The Detail View allows users to edit fields but still presents it in a clear way that makes a view-only page redundant.

# DESIGN EVOLUTION

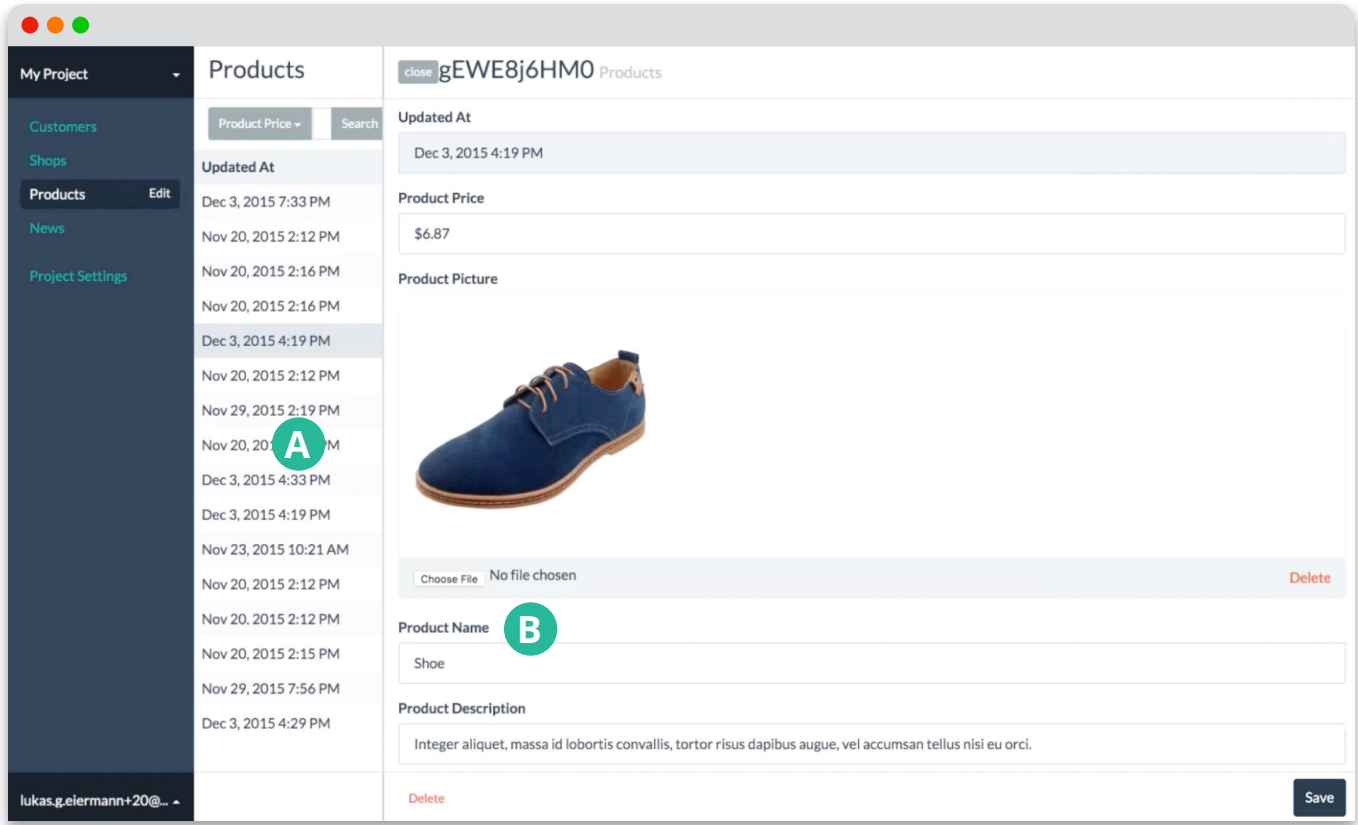
## Detail View (continued)

Based on the user feedback from our early prototypes we launched the detail view with the first version of Adminca. Responding to the usability test results we further iterated the design for version two.

*“I like it that it allows me to directly go to other records when I’m done viewing the details of one record. But why can I only see one column?”*

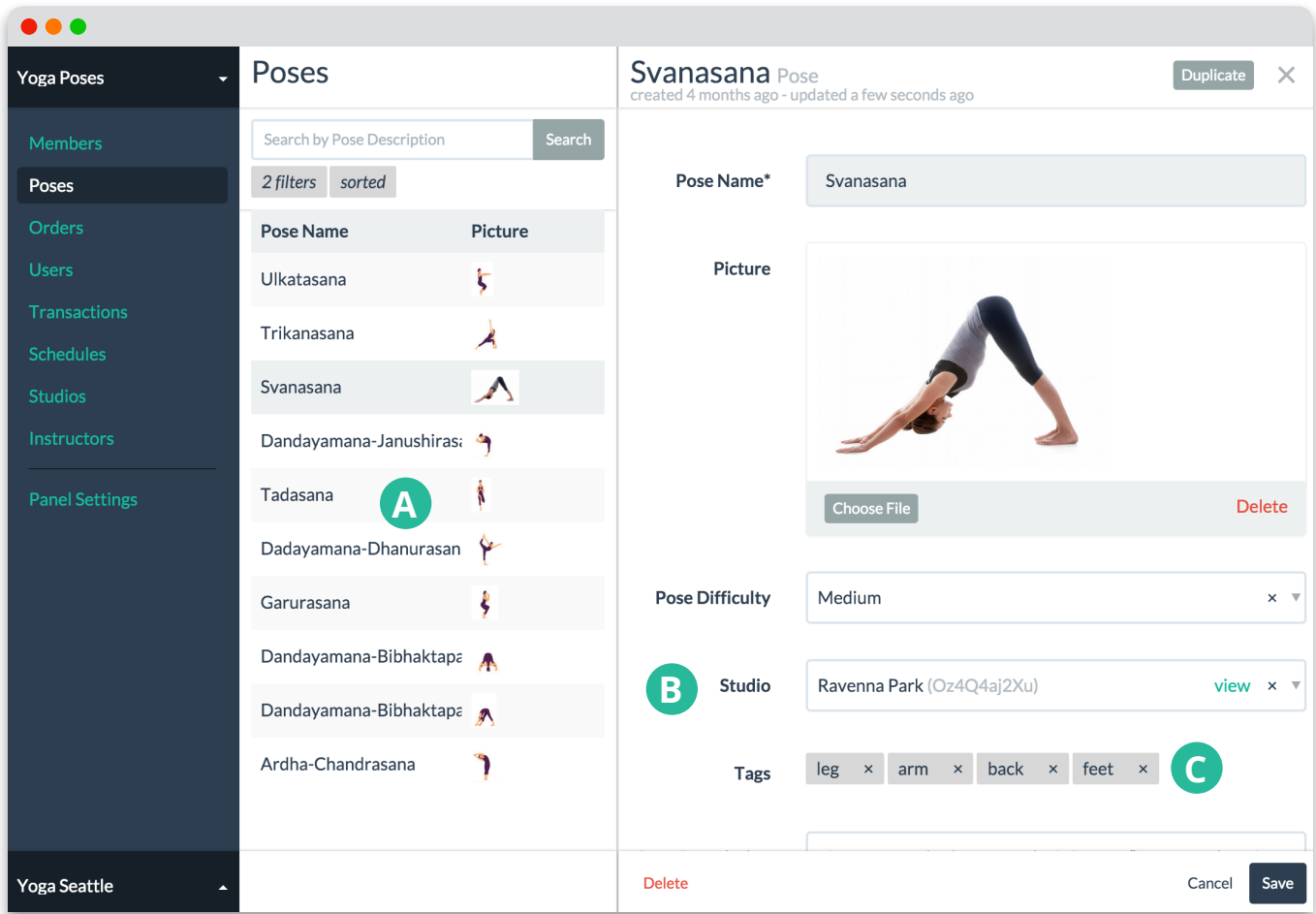
— Former catalog specialist at an e-commerce company

- A** In the first version the table only allowed vertical scrolling and was limited to one column. In the second version users can scroll vertically and horizontally and can view multiple columns at once.
- B** We changed the field labels from vertical labels (on top of the field) to right-justified horizontal labels. This reduces the vertical space needed while still having the label and the field close together.
- C** We added more field type formatting options. Strings, for examples, can either be formatted as an email address, a one line text or a long text.



▲ First Version

Second Version ▼





# DESIGN EVOLUTION

## Settings View

After creating the panel, the main task for an admin is to customizing the panel. Through our interviews, we learned about the problems that developers currently face.

### Limitations of current solutions

**All settings are defined through code.** No other person (e.g. a product manager) can make changes to the views without involving the developer.

**No easy way to customize views.** Once the admin panel is in production, it won't be modified because even small changes require a lot of work.

**One size fits all.** Admin panel templates and plugins only allow users to change a few settings that don't account for all scenarios.

Test Project ▼

Shops

Products Edit

Customers

Products

Edit Class

Class Settings

Edit Fields/Columns

Product Name

Name Field

Product Picture

Product Price

CreatedAt

Belongstoshop

Product Price

Tags

Product Review

CustomizeShop

Save Cancel

Product Review Settings

GENERAL

Visability ☒ Show Field/Column ☐ Hide Fields/Class

Label 

Product Review

Field Name *product\_review*

Description

PROPERTIES

Allow Users to: ☒ Edit this Field ☐ Only View this Field

☒ Search by this Field

Required ☐ Make this Field a required input

## Our approach

Adminca lets users modify the settings through an interface rather than code.

All settings for a class and its fields can be accessed and edited in one place. Changes are reflected instantaneous.

Adminca allows users to change granular permissions and settings for all classes and fields.

# DESIGN EVOLUTION

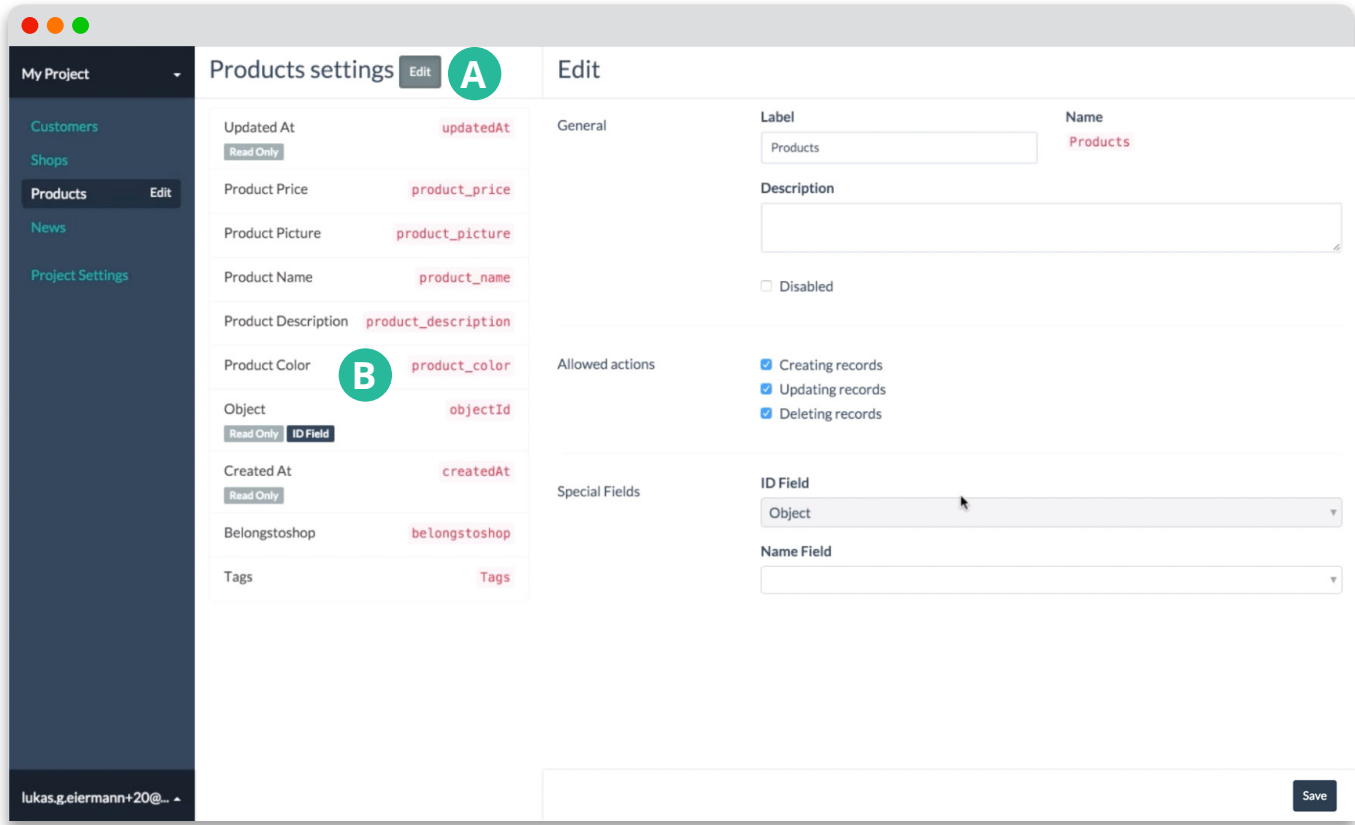
## Settings View (continued)

Based on the user feedback from our early prototypes we launched the settings view with the first version of Adminca. Responding to the usability test results we further iterated the design for version two

“Where are the class settings? I only see all the fields here.”

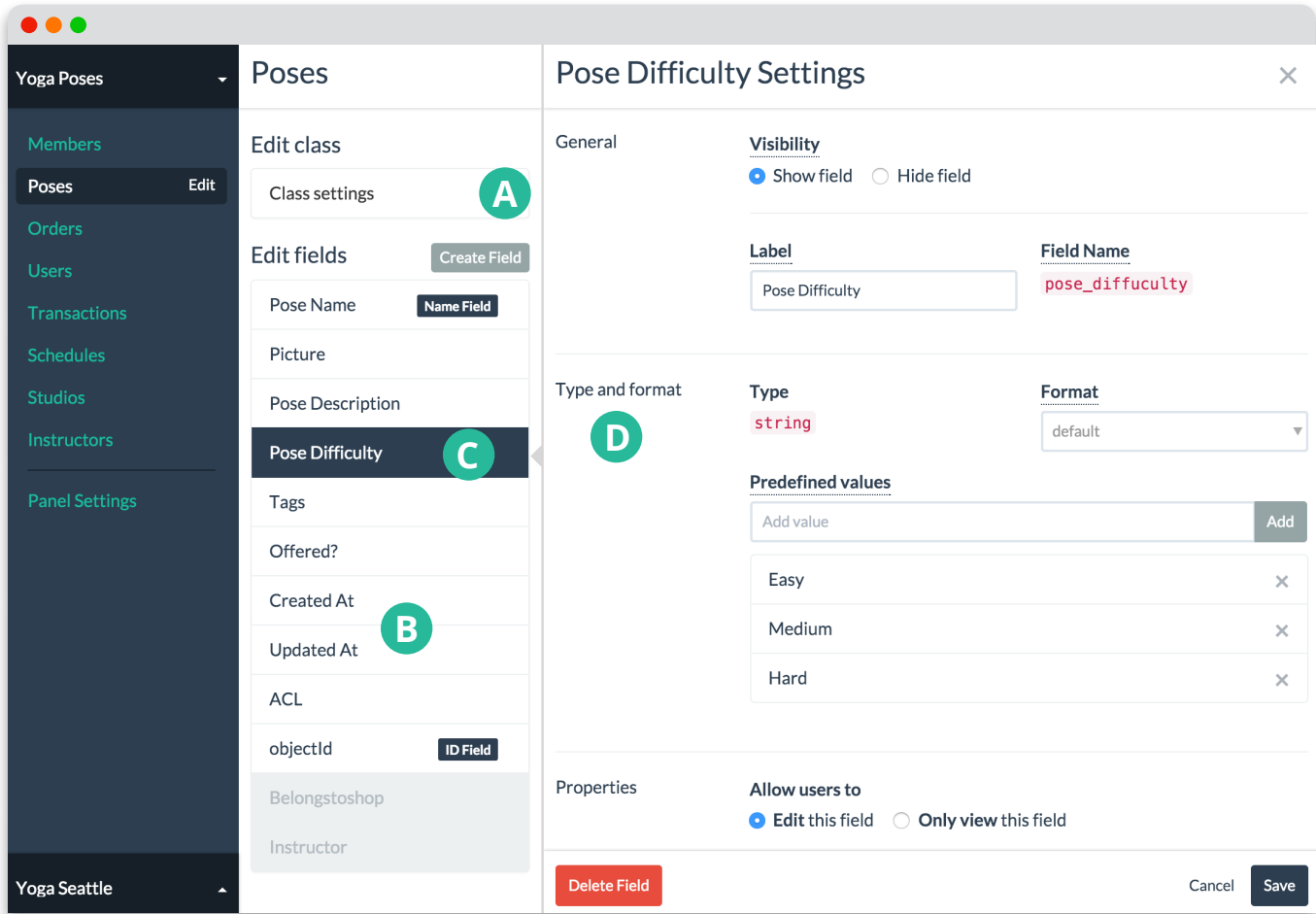
— Parse developer building apps for clients

- A** The class settings in the first version were difficult to discover. For the second iteration we adjusted the IA and moved it to the side navigation into it's own section to highlight it.
- B** We simplified the side navigation by removing additional information that was distracting for users.
- C** Users can now rearrange the fields in table by simply dragging the fields. The name field stays at the top and hidden fields are shown at the bottom.
- D** Simplified the structure in the main view. Created new and easier-to-understand categories and descriptions. Added tooltips for each field that explain details.



▲ First Version

Second Version ▼

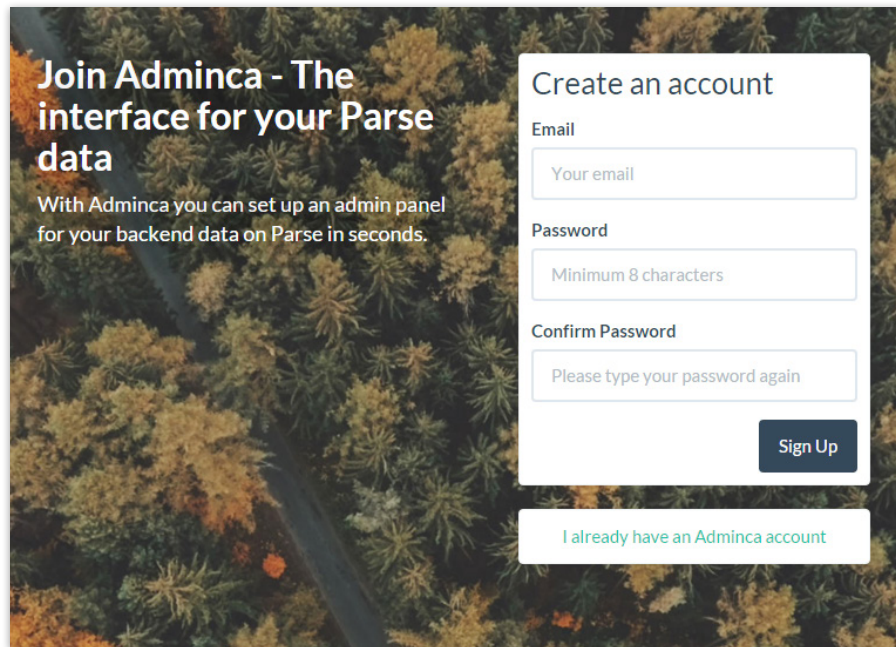




# DESIGN EVOLUTION

## Onboarding

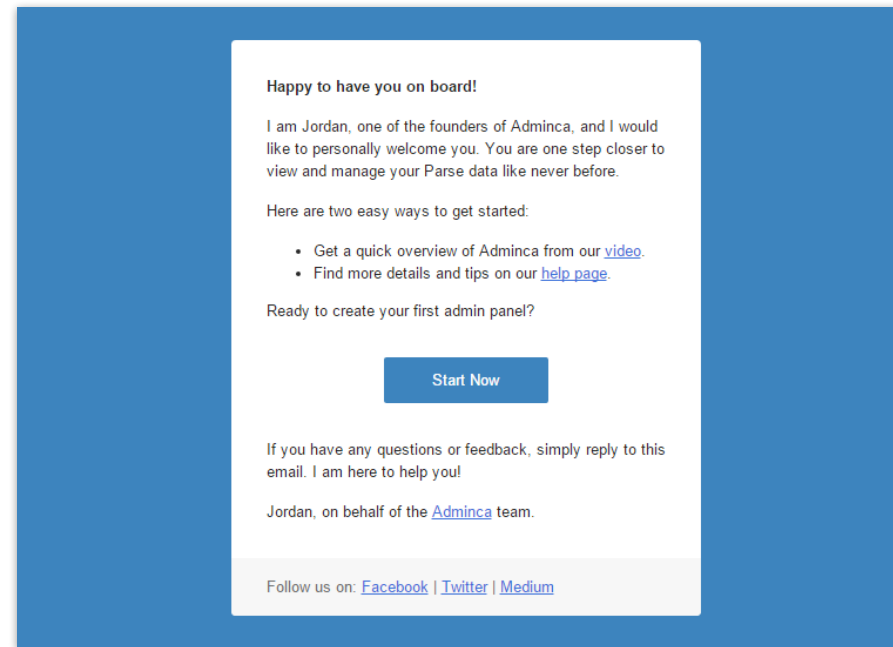
We introduced a new onboarding flow in the second iteration to simplify and streamline the signup process, and help users get started.



### 1. Account Creation

By selecting "Create admin panel" on the marketing website the user is brought to signup page.

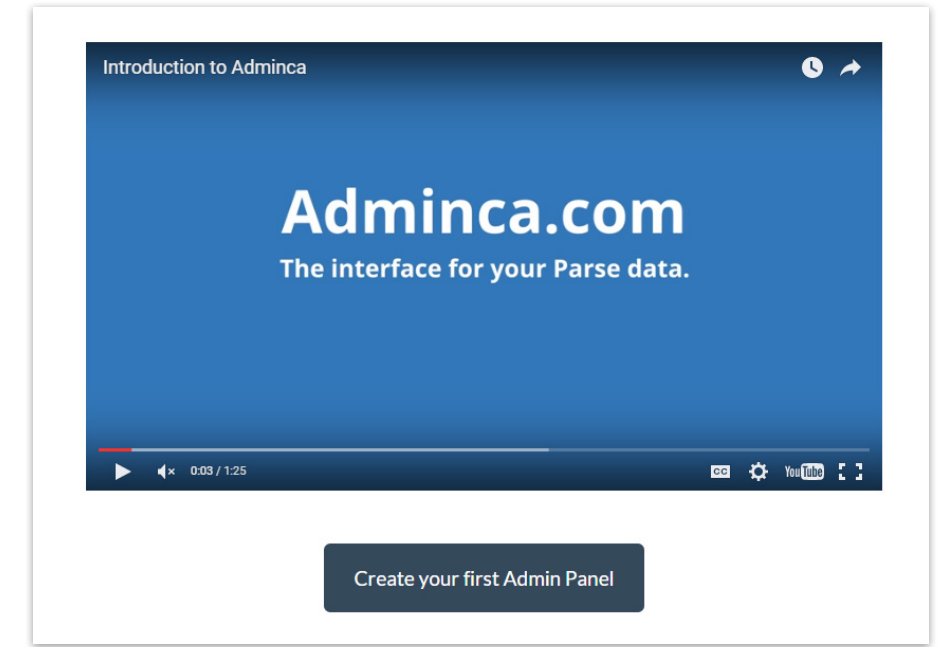
We removed unnecessary fields and reduced the required information to the essential (email and password).



### 2. Welcome email

Once the account is created the user receives a welcome email with additional resources.

Initially it was required to verify the email address with the welcome email. We removed this step so users can continue their process without interruptions.



### 3. Intro Video

After creating an account the user can watch quick a 90 second introduction video.

In the first iteration we saw that users understood the basic concept from our marketing website but weren't exactly sure how Adminca works. We added the video to provide a quick overview of Adminca and show how a panel can be created and customized.

# DESIGN EVOLUTION

## Onboarding (Continued)

Create an Admin Panel

Admin Panel Name

Admin Panel Name

Enter your Parse App Credentials. ?

Application ID

Application ID

Master Key ?

Master Key

Cancel

Create

### 4. Create Panel

After the video the user is brought directly to the panel creation page.

According to our usage data this was the step where most users dropped off. We learned that most users weren't comfortable with providing the Master Key and added a new tooltip that explained our security measures and why Adminca required the master key.

Table View (3/8)

Click on a class in the navigation to see its data in the table view. Sort data by clicking on column headers. With the search bar you can search data by any field.

Previous

Next

exPSPnqpXd	Public Read Write
O127ELwSXh	Public Read Write
RuC5HdPqia	Public Read Write
fXKnhl8ea0	Public Read Write
DZ8ELgFghv	Public Read Write
FM0m8Llkjk	Public Read Write

jordan@adminca.com

64001 records

### 5. Onboarding Tips

The admin panel is automatically created and the user is brought directly to the main view.

This is the 'wow' moment and we made sure to get the user to this point as fast as possible. We observed that users didn't know what exactly they were supposed to do next. We added a guided tour (8 steps for admins and 5 steps for contributors) to easily get started with Adminca.

Help

Creating a Panel

Using a Panel

Customizing a Panel

Inviting Collaborators & Admins

Glossary

Creating a Panel

Getting Started:

Creating an Admin Panel for your Parse data couldn't be easier. After signing up assigning a name to your first admin panel. Then you can put in your Parse App application key and the master key under settings/keys in your Parse app. When be automatically created for you.

Security:

- Why does Adminca require the Master Key? The Master Key is the only key app. Adminca requires the schema to understand which classes are included appropriately.
- Who can see/access the Master Key? Nobody. The Master Key can't be accessed by collaborators nor the Adminca team are able to view your Master key. The account in case someone else logs into your account.
- How is the Master Key stored? Adminca uses SSL to establish a safe connection. On our server the Master Key is encrypted with the AES-256 algorithm.

Next

### 6. Help Section

More detailed explanation about all of Adminca's features are available in the help section.

In our tests we saw that after the onboarding tips, most users didn't need a help section. However we still added it in the new version, because it was frequently requested by our users as something that they would like to have "just in case".

# CHALLENGES

## Cover edge cases

When we created our prototypes, we used our dummy app database to verify some of our decisions, for example, width of table column, area of class names, etc. But as we conducted usability testings with our real users on their real apps, we were shocked that sometimes there are super long class or field names, more than ten class, or more than 100 fields in a single class. These somewhat extreme cases gave us some hard to make our design scalable.

## Diverse Needs

Mobile apps can be very different one from another, and users who manage different app data have their unique needs. Thus, we often hear users asking for specific feature. It is hard to incorporate all the feature request into the design.

## Developer & non-tech needs don't align

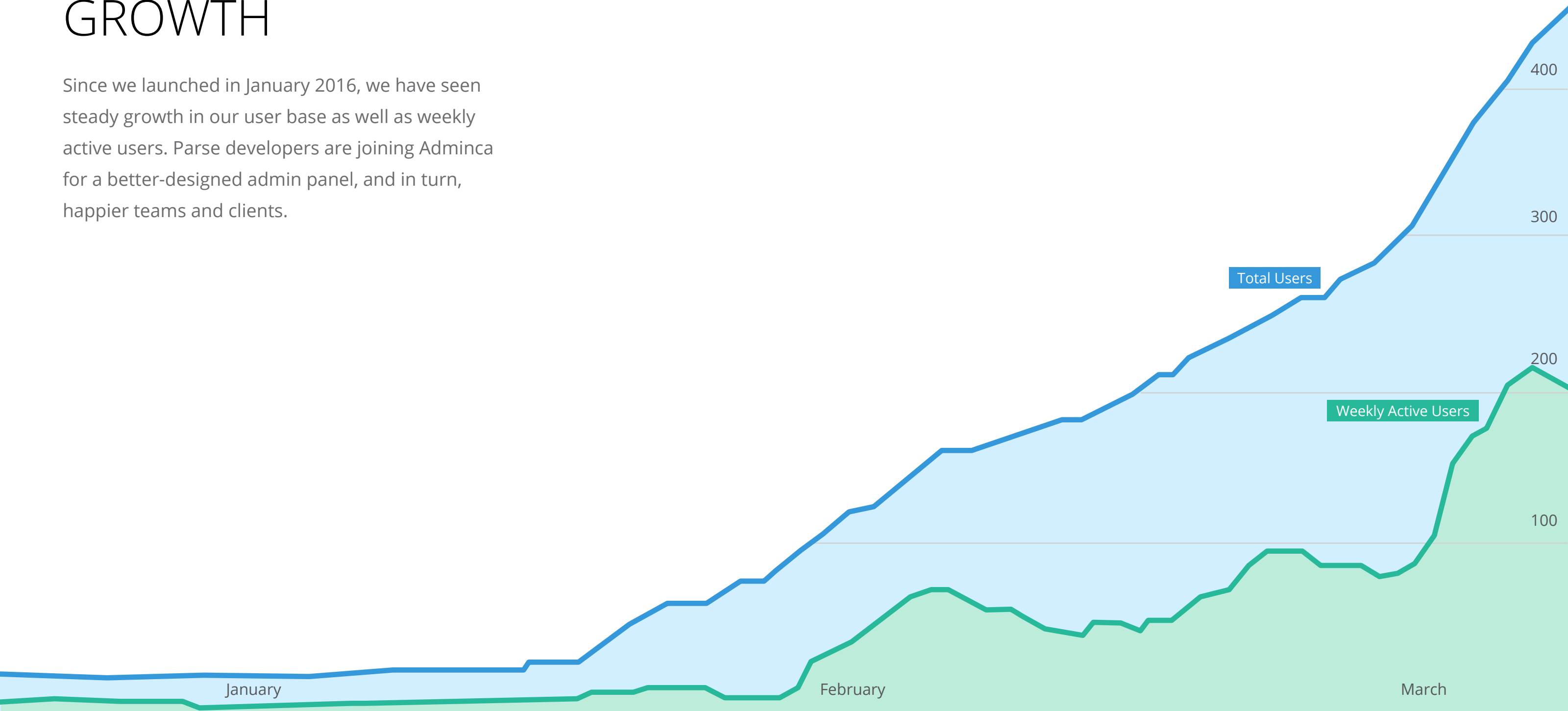
Adminca serves both developers and non-technical users, but sometimes it's challenging to meet both sides' need simultaneously. For example, some developer would like the downloaded CSV file to have the original Parse headers for the column name, but non-technical users prefer to see the labels they defined in Adminca as the headers.

# IMPACT & OUTLOOK

---

# GROWTH

Since we launched in January 2016, we have seen steady growth in our user base as well as weekly active users. Parse developers are joining Adminca for a better-designed admin panel, and in turn, happier teams and clients.



# USER FEEDBACK

Philip Kotler—the marketing guru—once said “The best advertisement is done by the satisfied customer”. Check out what our customers tweeted and emailed us about Adminca!

Good job on the product! Awesome to see you guys helping our Parse developers.

✉ James Yu,  
Co-Founder of Parse

I blown away when I had up and running in less than 30 seconds. [...] I can't believe how user friendly it is!

✉ Gabel W

@A\_dminca, I really like your product, it's the missing piece.

🐦 Chris B, Co-Founder of Catalyst Mobile

You guys are awesome! Nice work. Clean, simple, useful.

✉ Alex S, CEO at Kenmore Design

Probando @A\_dminca ES INCREIBLE!!

🐦 Arturo J, Co-Founder of Brounie

I really love what you guys have created. It's so much better than Parse's own Dashboard!

✉ Johan N, Co-Founder of Meetio

The dashboard seems extremely powerful and useful, I'm very happy with it!

✉ Antonio G, Director at Mobixee

@A\_dminca nice! Adminca is a godsend. :D

🐦 Jayson

@A\_dminca is a breath of fresh air to my Parse based projects. Finally a client friendly UI that just works!

🐦 Sasha R, CTO at Hyper Apps



# FINAL REFLECTION

Thinking back, there are a few things that we would do better if we had the opportunity to start again.

## Recruit more non-technical users

Even though we were able to learn about our target users and identify the main workflow for developer and non-technical users, we still want to validate our findings with more users in each type. Due to the limitation of our recruitment methods (mainly searching for Parse developers on GitHub, and mobile app development agencies), we were not able to talk directly to as many non-technical users who manage their data on Parse. We have tried to reach out to local startups and apps that are listed on the customer page of Parse or other mobile backend as a service website, but wasn't able to recruit non-technical users.

It is not until our developer users invited their non-technical partners or clients when we finally directly hear about the non-technical user feedback. It is highly possible that we could have

discovered more about backend data management needs from a variety of non-technical users. More use cases and scenarios were likely to be found, which could in turn lead to new features we need to design.

## More market research

Related to the non-technical user recruitment issue, we should have done more market research at the beginning of the project. Since we decided to turn this capstone project into a startup, we should have examined closely the feasibility of the idea as well as the market size and profitability. As we found during our interviews, there was only a small portions of app agencies that build client app with Parse. Among the 400,000 Parse developers, most of them are working on solo project, in which case an admin panel is not a necessity. Therefore, to validate our project idea as potential business initiative, we should have done more market research before committing to the idea. It is very possible that market research would revealed us

either a large potential market for other backends, or a totally new approach in the backend data management field.

## Parse Shutdown

As we started last September, Parse was still one of the leading mobile backend as a service, backed by Facebook and used by more than 400,000 developers. In November 2015, Parse hosted promotional events in Seattle and encouraged developers in the community to jump on board to leverage the advanced features of Parse. Even in January 2016, Parse launched a visually redesigned website as well as data browser. Everything looked promising that Parse was going to expand and provide better support to developers.

It was so shocking to us and the community that on Jan. 28, Parse announced that it is going to shut down in a year, and everything is going to be open-sourced. To developers, it means they need to find a new service or host their app on their own server.

Either way there will be a lot of extra work, or even rewriting their app. For us, it means Parse is even less likely to be the top choice for app agency to build client apps with, which was then our primary customer base.

But it also created an opportunity for us. Because Parse didn't say anything about open-sourcing their data browser, the interface for Parse data management. It left a void to fill, because now even a solo developer would need an admin panel. And that's what Adminca can provide.

So we shifted our focus from designing and developing new features to supporting the open-sourced Parse Server. This time we attracted a lot more attention from the Parse developer community. We thought filling the admin panel void could be the turning point of our business outlook. However, 4 hours after we announced "we now support Parse Server", Parse announced that the data browser is also open-sourced. Another lightning over head.

# FUTURE DIRECTIONS

## Launch Version 3

We already conducted the next round of usability tests on the current product, and are ready to iterate for another time. On our roadmap are exciting new features like bulk actions and predefined views, based on our findings and user feedback.

## Create white label solution for Partners

Companies from New York, Florida and the UK reached out to us because they were interested in using Adminca as a data interface for their platform. We see this as an opportunity to sell a white label version Adminca.

## Explore other backends

From the beginning it was our goal to bring better admin panels to all types of backends. We already started working on making Adminca available for Kinvey (service similar to Parse) as well as Ruby on Rails.

[www.adminca.com](http://www.adminca.com) ►

Adminca

[Blog](#)

[About](#)

[Parse Shutdown FAQ](#)

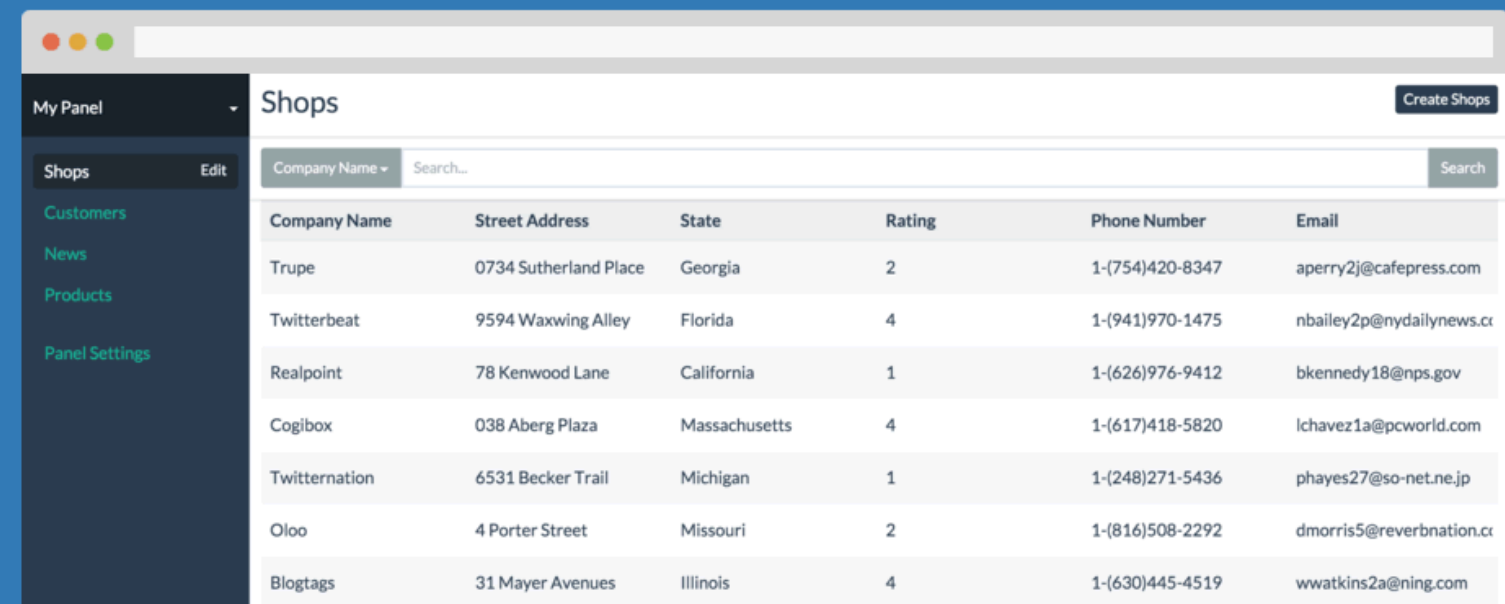
[Log in](#)

[Sign up!](#)

# The interface for your Parse (Server) data

Adminca empowers your team to view and manage data like never before.  
Get an admin panel in seconds without writing any additional code.

Create your Admin Panel



Company Name	Street Address	State	Rating	Phone Number	Email
Trupe	0734 Sutherland Place	Georgia	2	1-(754)420-8347	aperry2j@cafepress.com
Twitterbeat	9594 Waxwing Alley	Florida	4	1-(941)970-1475	nbailey2p@nydailynews.c
Realpoint	78 Kenwood Lane	California	1	1-(626)976-9412	bkennedy18@nps.gov
Cogibox	038 Aberg Plaza	Massachusetts	4	1-(617)418-5820	lchavez1a@pcworld.com
Twitternation	6531 Becker Trail	Michigan	1	1-(248)271-5436	phayes27@so-net.ne.jp
Oloo	4 Porter Street	Missouri	2	1-(816)508-2292	dmorris5@reverbnation.c
Blogtags	31 Mayer Avenues	Illinois	4	1-(630)445-4519	wwatkins2a@ning.com

## Connect your team or clients with Parse data

Your team or clients need access to their data in order to manage it. But inviting them to your Parse Account can put your whole backend data and code at risk. With Adminca you can create a Parse admin panel and have full control of which data can be viewed or edited.



# THANK YOU

Thanks to Liz, Ruth, and Jared, as well as the HCDE department at the University of Washington for all their support and guidance throughout the project!

We also want to thank everyone who signed up for Adminca. We appreciated your feedback and support during the last three months.

