



Argi Kartika

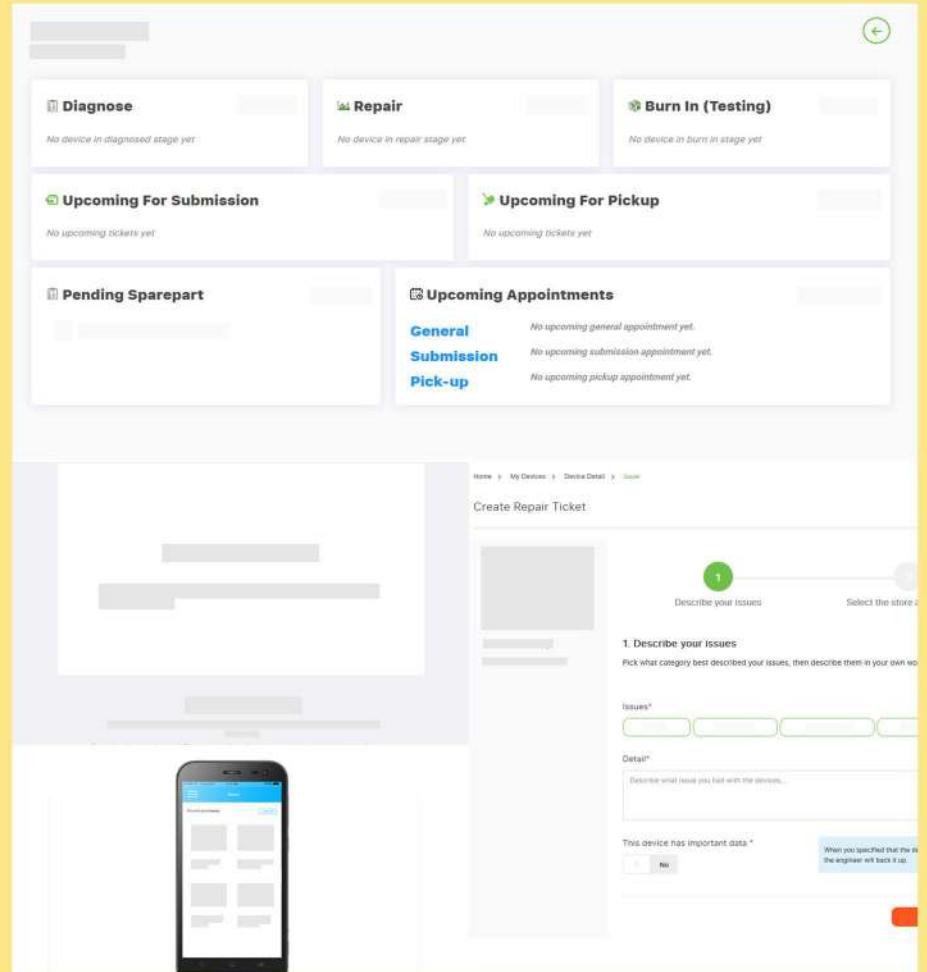
Web Designer &
Developer

Hello! I am a web designer/developer with a background in illustration, mainly works with Vue.js. I love exploring and experimenting in search of delightful and maintainable website. In my spare time, I like to read, write, and draw things related to ancient Japanese culture.

+62 858 2891 5534
argikartika@gmail.com
codepen.io/artikandri

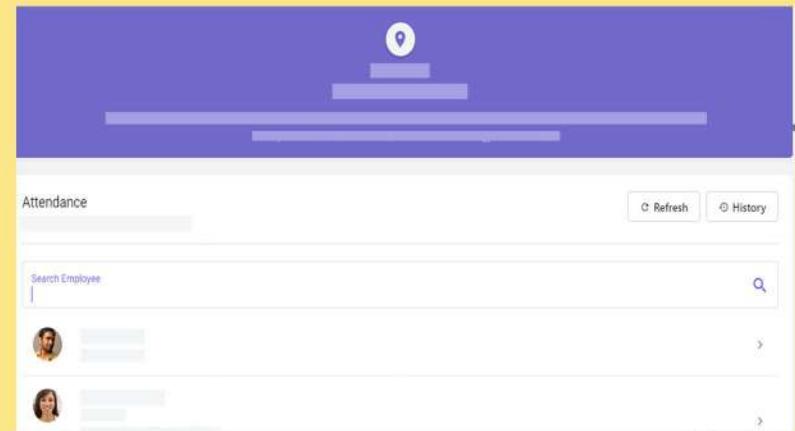
Repair Service

A fully featured repair services web & mobile app; designed to facilitate smooth repair and ensure direct communication between the customers and people behind the service, such as the engineers and the administrator. Made with Phalcon, socket.io, JQuery, Bootstrap, and Vue.js.



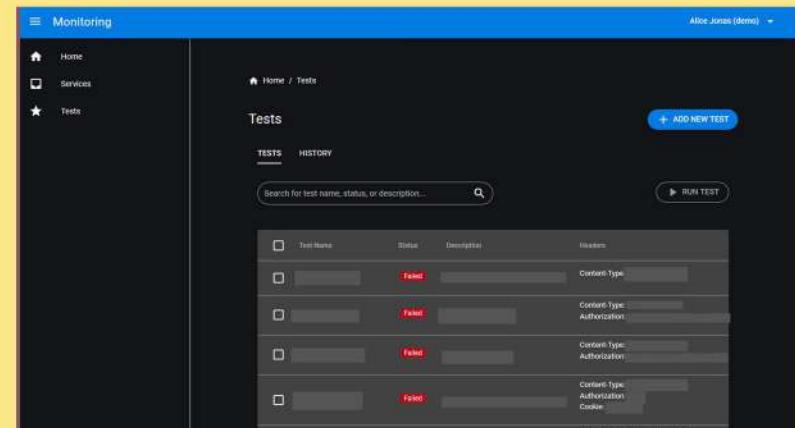
Attendances Taking

Desktop app to facilitate employee's attendances taking process; made with Electron, Vuetify, and Vue.js



API Monitoring

Web app to monitor API request trend and catch errors, made with Quasar and Vue.js



HRM: Access Level

Control employee ability to access certain menu, information and/or features with simple configuration; made with Bootstrap and Vue.js

The screenshot shows a configuration interface for access levels. It includes three tabs: 'General' (selected), 'User', and 'Others'. The 'General' tab allows setting menus and settings for all users. The 'User' tab allows setting what user with this access level may do to their own profile. The 'Others' tab allows setting what user with this access level may do to the other employee's profile. Below the tabs, there are sections for 'Main menu*' and 'Setting menu (optional)', each with a tree-view checkbox interface for selecting menu items.

HRM: Dashboard

Keep track of what's happening in the company with fully customizable widgets; made with Bootstrap and Vue.js

The screenshot shows a dashboard interface with a grid of empty widgets. A message at the top says 'The list is empty' and 'No applicable filter'. At the top right, there are buttons for '+ Add Widgets' and 'Save Widget Changes'. The bottom right corner shows the date '02-Feb-20'.

HRM: Others

Notification: Receive e-mail and push notifications on demand

User Management: Create/generate/manage/delete users

Documents: Upload and share documents between users

Apps: Create and share installable add-ons for the company

Report: View and download summary as PDF/Excel report

Payslip Setting: Configure payslip content and size

The screenshot shows a notification for a time off request. It features a clock icon and the text 'Your Time Off Request is waiting for review'. Below this, it says 'You have requested a time off with the following details.' and lists fields for 'Applicant Name', 'Time Off Name', 'Duration', and 'Date'.

Adept: E-learning

Mobile app for viewing courses, practicing, and taking test; built with Framework7, Vue.js, and Cordova.

Preview Answers Close

1. What are you doing when fire coming?
Call fireman

2. What will you do if there was a suspicious person?
There is no more true than this

SUBMIT

< Practice

Question (1 of 5)

1. What are you doing when fire coming?

Select an Answer

Call fireman

Screaming

Spring the water

NEXT

PREVIEW & SUBMIT

1 2 3 4

Misc: Illustration

Various traditional & digital illustrations, drawn with Photoshop and Paint Tool SAI (digital) or watercolor (traditional).

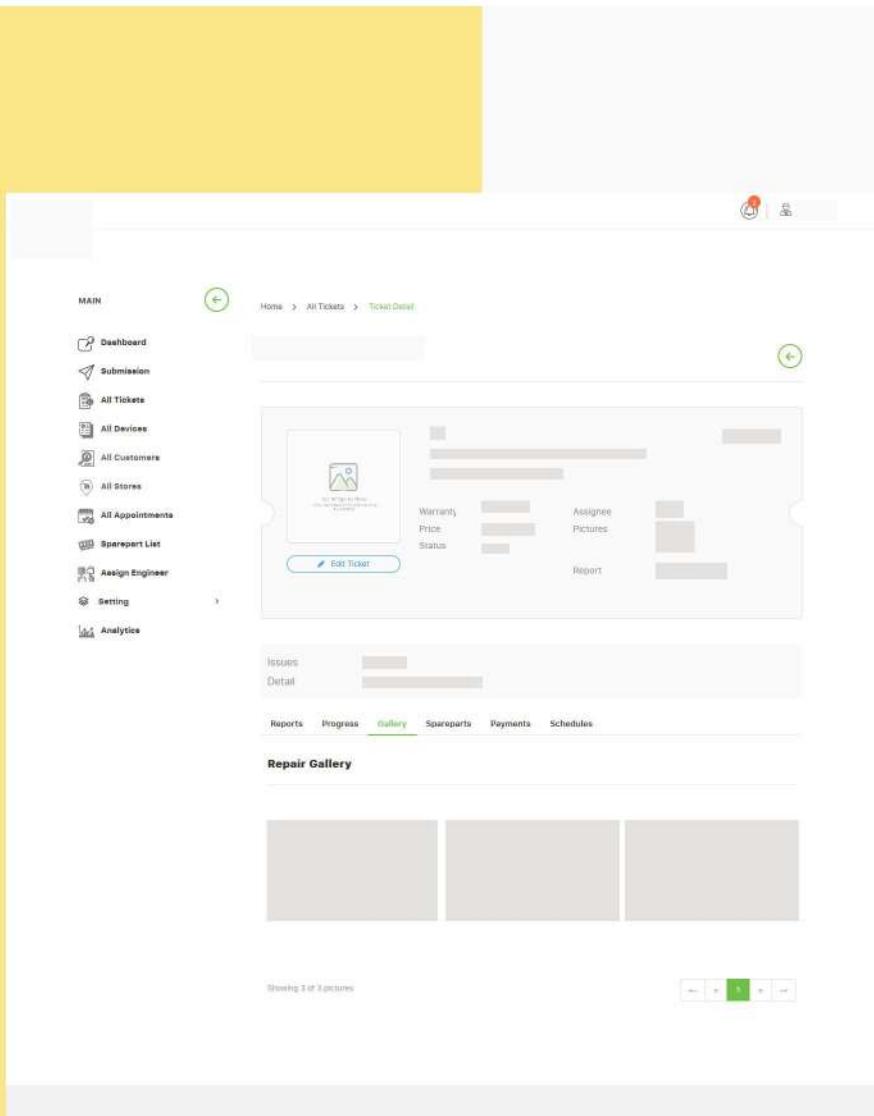




Repair Service



Repair Service



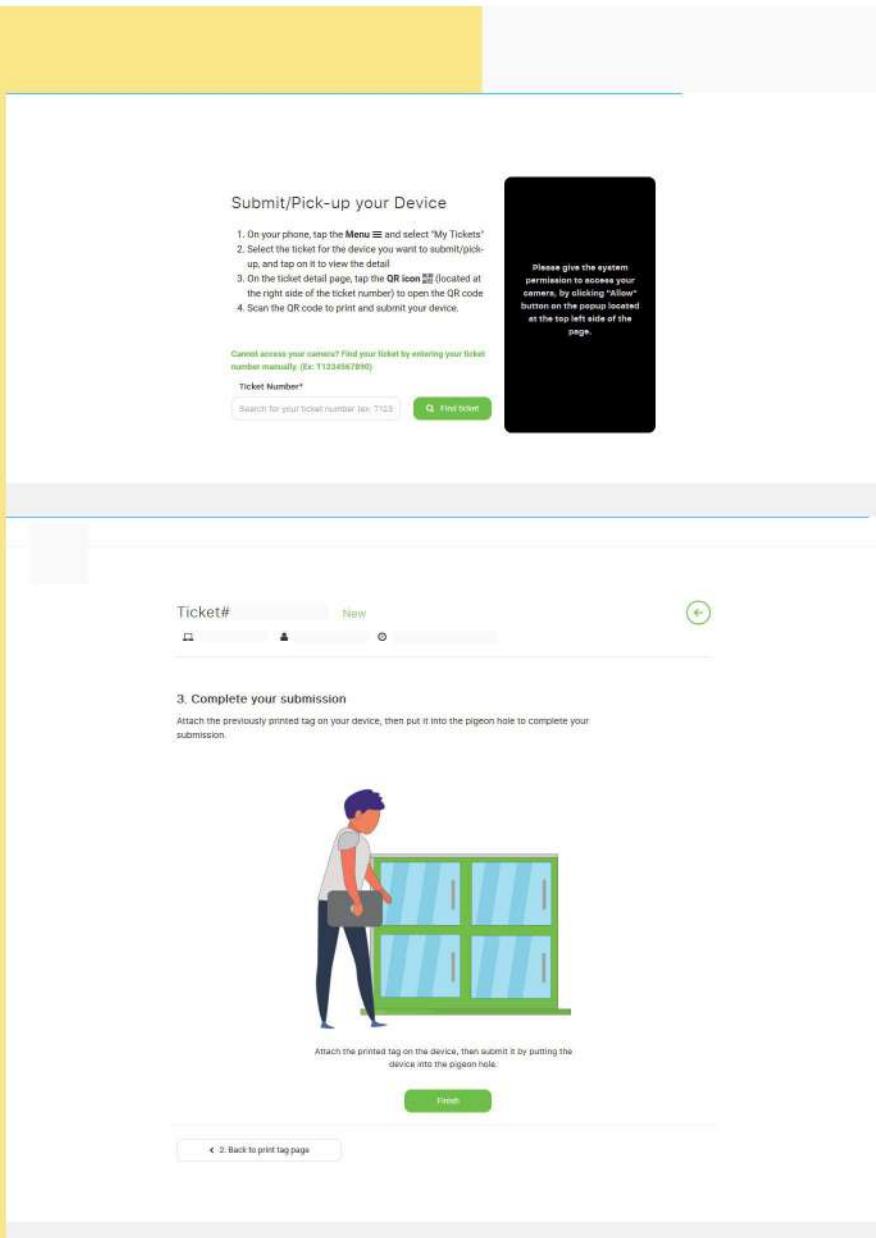
Project brief

A one-year project to design and develop a web app and mobile app to monitor repair service progress. Depending on the user type, the system may allow the user to manage tickets, schedule appointment, send chats, receive real time notifications (push notification and email notification), add report, suggest sparepart, and manage the payments; from start to finish.

Platform

Web, App

Repair Service



Roles played

Designer (Web, App prototype, E-mail)
Front-end Developer (Web, E-mail)

Tools

Prototype & Design
Balsamiq, Photoshop

Development
Phalcon, Vue.js, JQuery, Bootstrap, socket.io

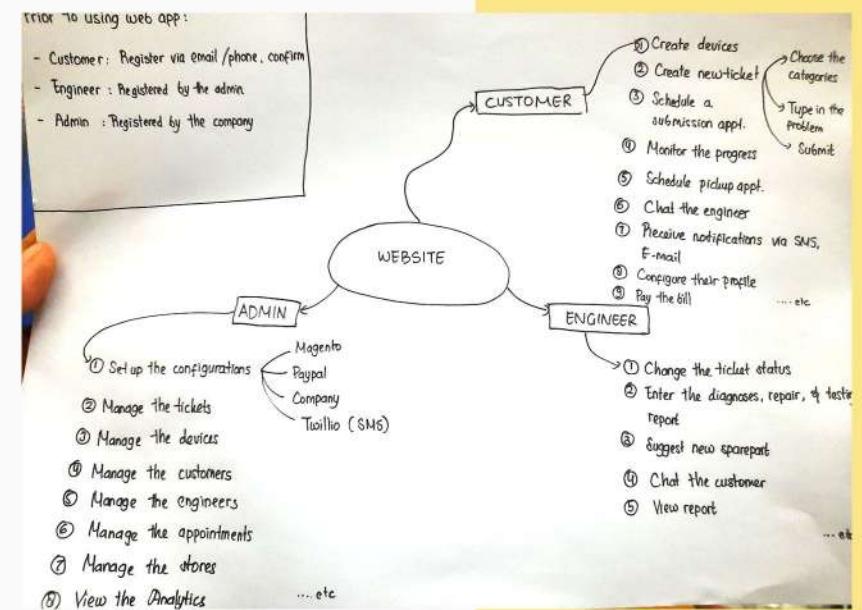
Language
HTML5, SCSS, JS (ES5), PHP

— Repair Service

Gathering the insights

Firstly, separated interviews with representatives from the company (done by another team mate) revealed that existing system used by the company focus on admin process only. Because of that, the client wants the new system to be approached from the engineer and customer sides.

In addition to that, the client planned to create a new, separated ecommerce web (w/ Magento), so the new system should be able to integrate data from that web as well. Later on in the future, the client also planned to integrate the system with automated hardware in the designated stores, so the new system should provide the configurations to enable doing so.



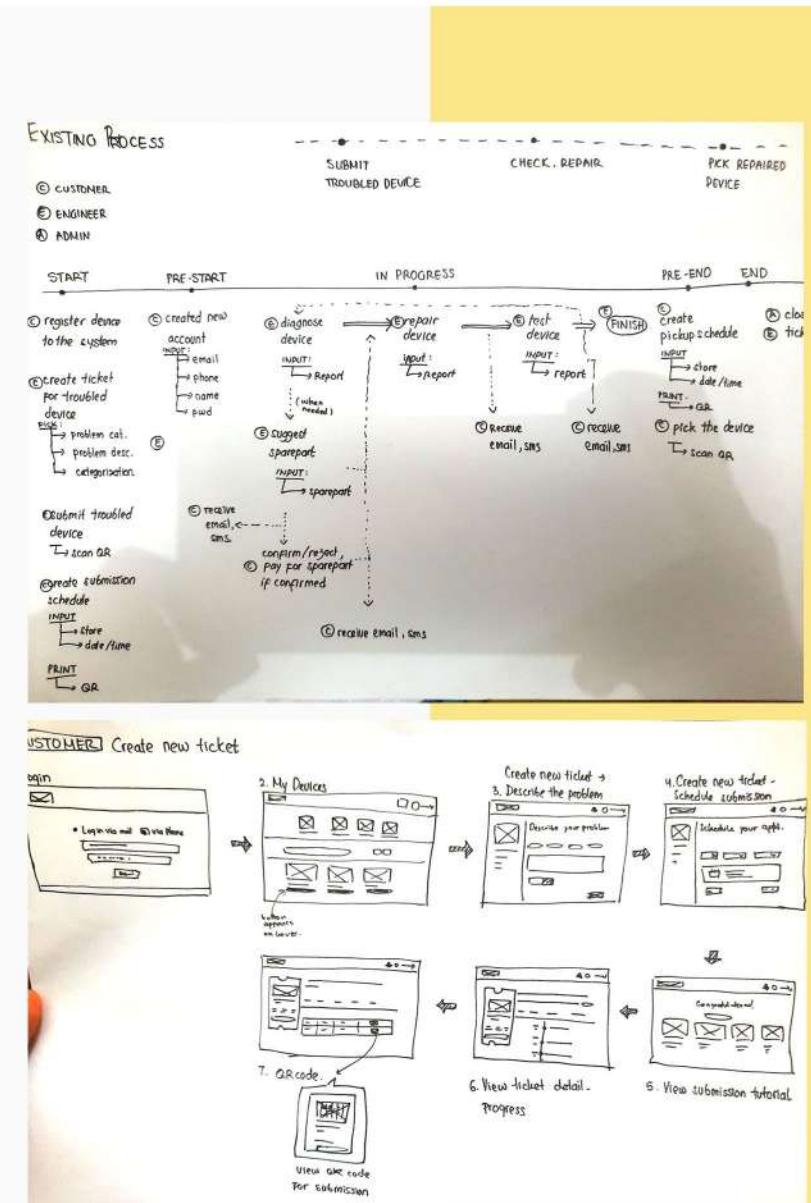
Repair Service

Analyzing the insights

After gathering the insights, I and another teammate started the development by breaking down the repair service process from start (creating the ticket) to finish (close the ticket). This enabled us to identify the key features for each user. In addition to that, we were also able to identify the abilities and restrictions for each user as stated below:

- . Which feature should be made available to the user?
- . Which feature should not be made available?
- . Why so?

After that, I tried designing the low-fidelity prototype of each feature. The prototypes were not pretty, but they were enough to help clearing our thoughts before starting development.

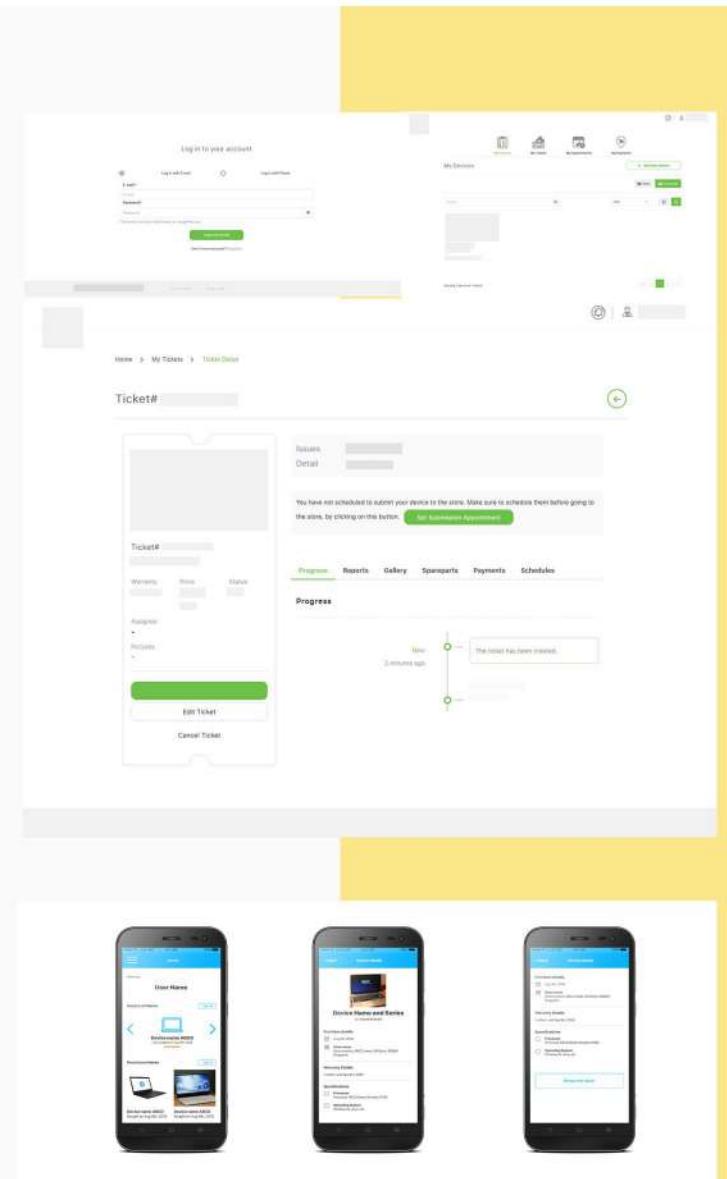


— Repair Service

Designing and Prototyping

Next, I started developing the high fidelity prototype for the website and mobile app. For the website, the best sort of prototype for getting realistic results was a full-fidelity HTML and CSS prototype, because it saves time and aligned with our agile process. The look and feel of the website was already designed by another teammate, so my responsibility was to translate the design into reusable code blocks. The web app was built with the Phalcon PHP framework, Bootstrap CSS framework, JQuery library, Vue.js JS framework, and socket.io.

On the other hand, the mobile app was given longer due date, so I was able to design the high fidelity prototype for the mobile app. While most of the app initial designs are no longer used, it still serves as the basis for later development. The mobile app was designed with Balsamiq (lo-fi) and Photoshop (hi-fi).



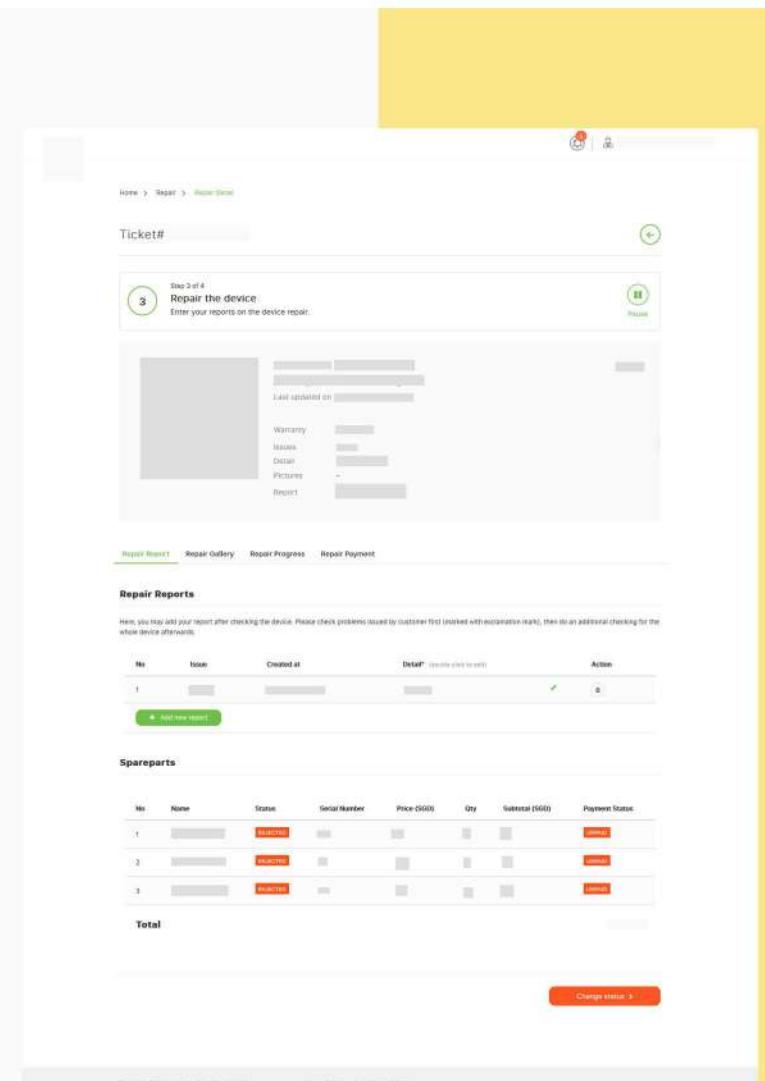
— Repair Service

Integration and Finishing

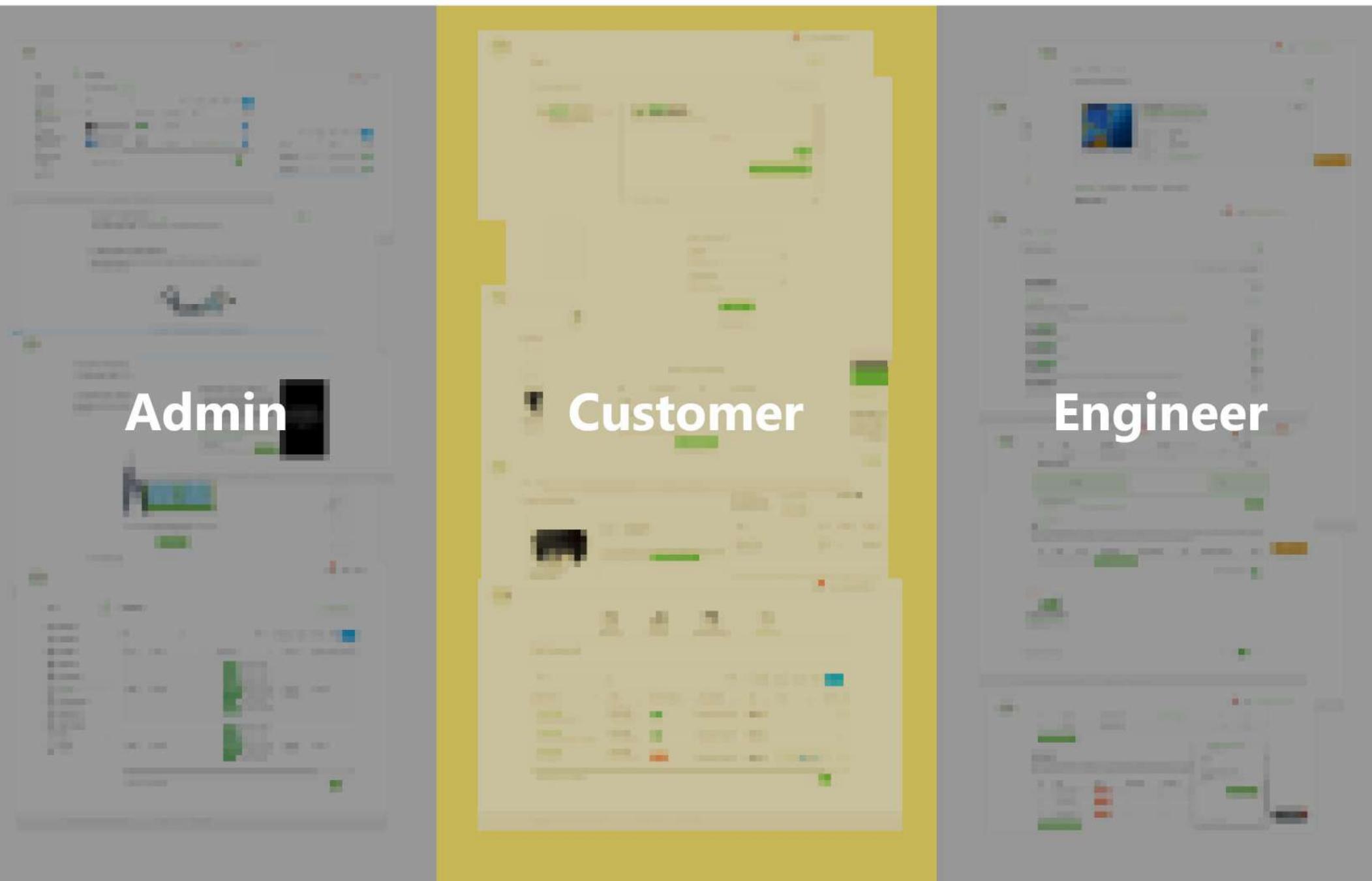
After prototyping, the next step was to implemented the REST APIs made by the back end, and make sure the feedback and error catching system for each API request works well. I also integrated the whole system so that it may communicate with each other in real time using socket.io. Finally, the task was finalized after writing the test cases and documentation for each feature.

Challenges

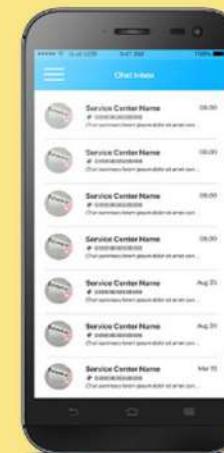
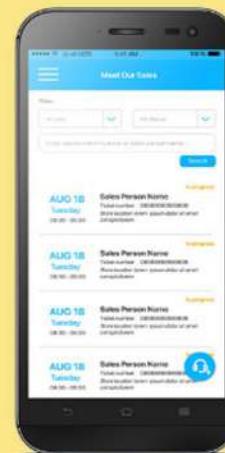
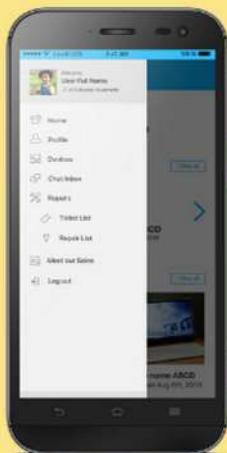
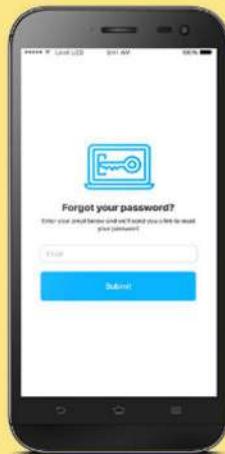
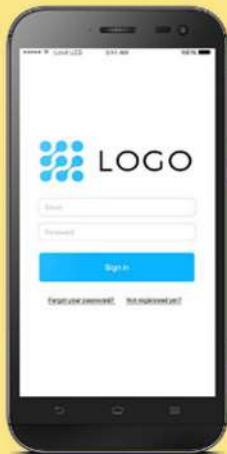
The very agile nature of the team and everchanging requirements were real challenges, especially considering the tight deadline. I overcame this by designing reusable and adaptable components, so I dont have to work on same feature twice.



Repair Service



Repair Service: Initial design of the mobile app

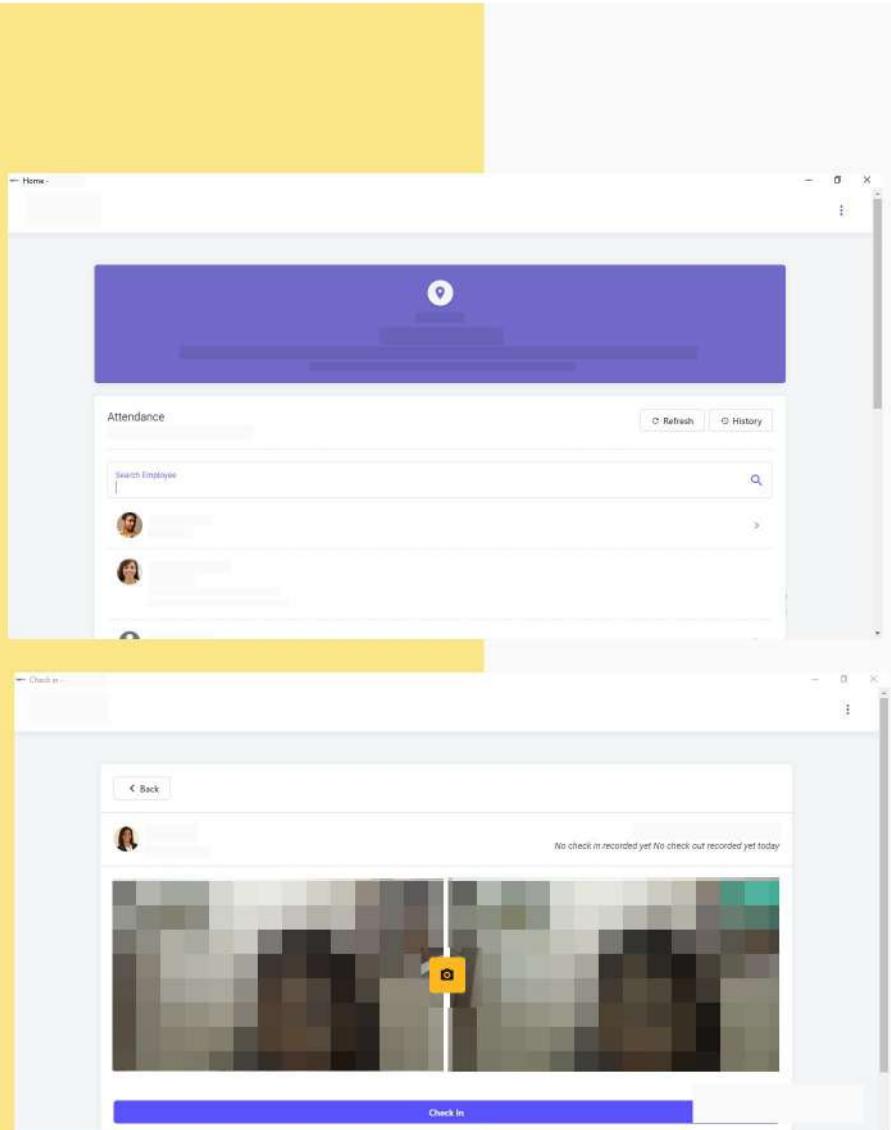


二

Attendances Taking



2 Attendances Taking



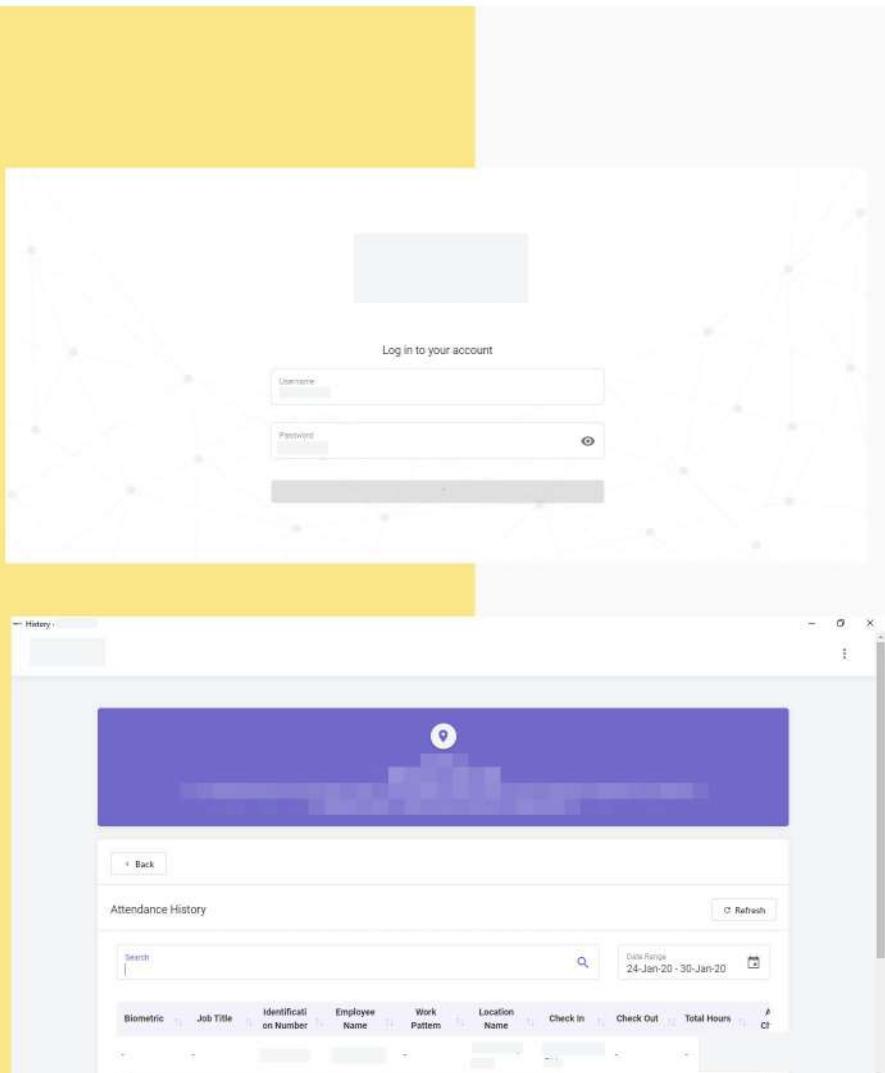
Project brief

A three months project to design and develop a desktop app for taking employee's attendances and viewing the attendances history under a specific location. The desktop app was built with Electron, and available for Windows and Mac.

Platform

Desktop App

2 Attendances Taking



Roles played

Designer, Front-end Developer

Tools

Development
Vue.js, Vuetify, Electron

Language
HTML5, SCSS, JS (ES6)

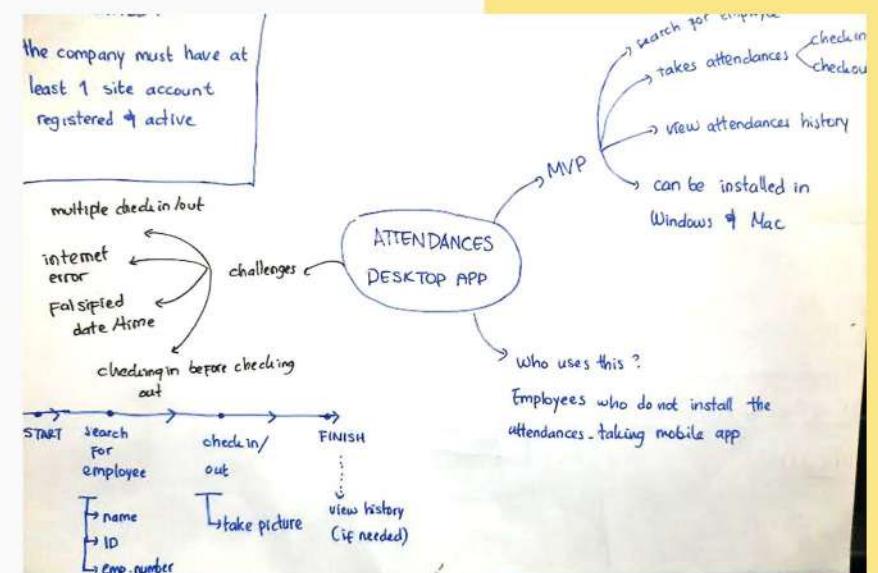
2 Attendances Taking

Gathering insights and analysis

Currently, the employees registered with the system are taking their attendances by taking selfie with a designated mobile app. Yet not everyone has the privilege to own device that supports the mobile app, so this desktop app was designed to help taking attendances with a computer instead.

Design and development

The app was built with Vuetify and Vue.js, wrapped with Electron. The styling was done with the help of SCSS preprocessor. The final product was a hybrid desktop-app, built with web technology but having the look and feel of a desktop app.

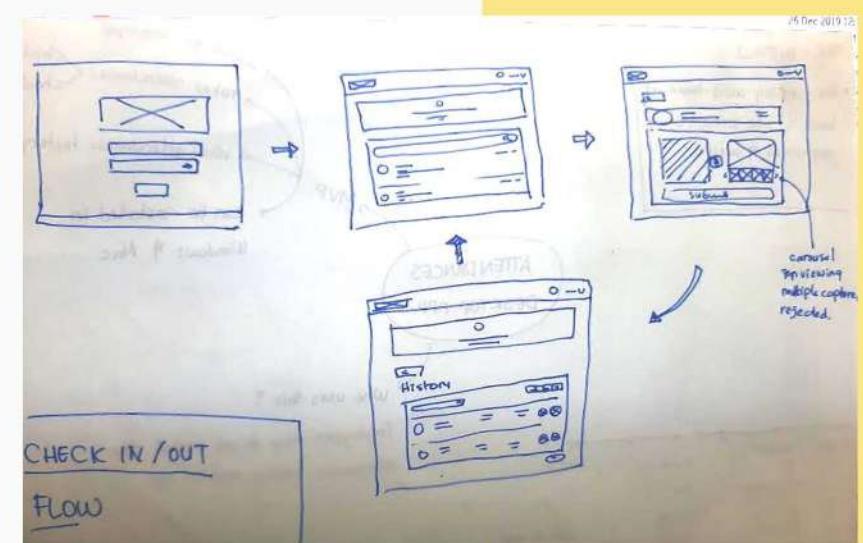


ニ Attendances Taking

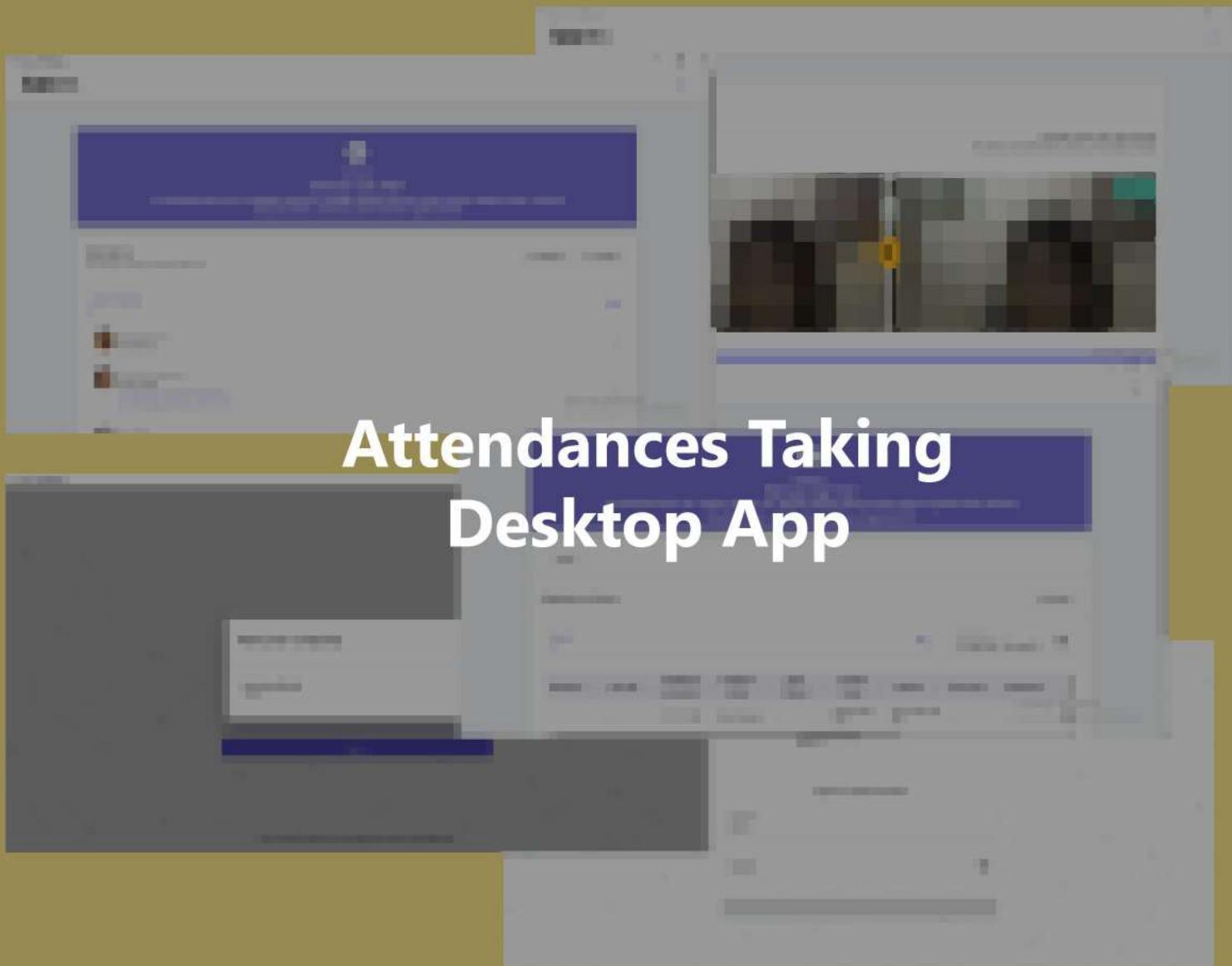
Challenges

While the scope of the project was small, it was not without trouble, since it was our first time building a desktop app with Electron. I was able to overcome that by reading the documentation.

Another trouble was when the client requested all of the font sizes to be multiplied by twice or thrice the initial, but I were able to finish the task within the deadline since I heavily employed SCSS variables and mixins in the design system in my search of scalable and maintainable CSS.



2 Attendances Taking

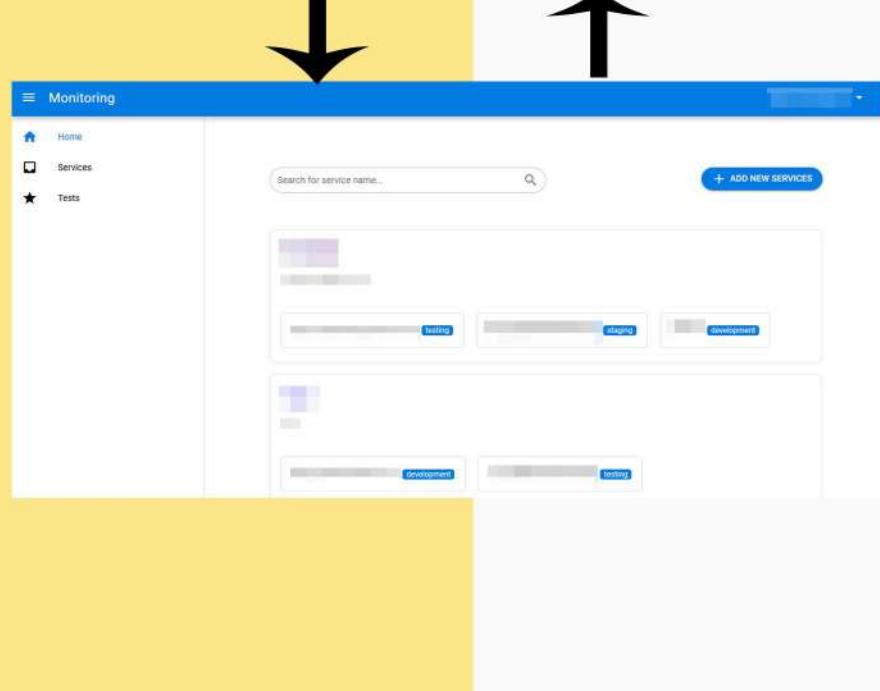
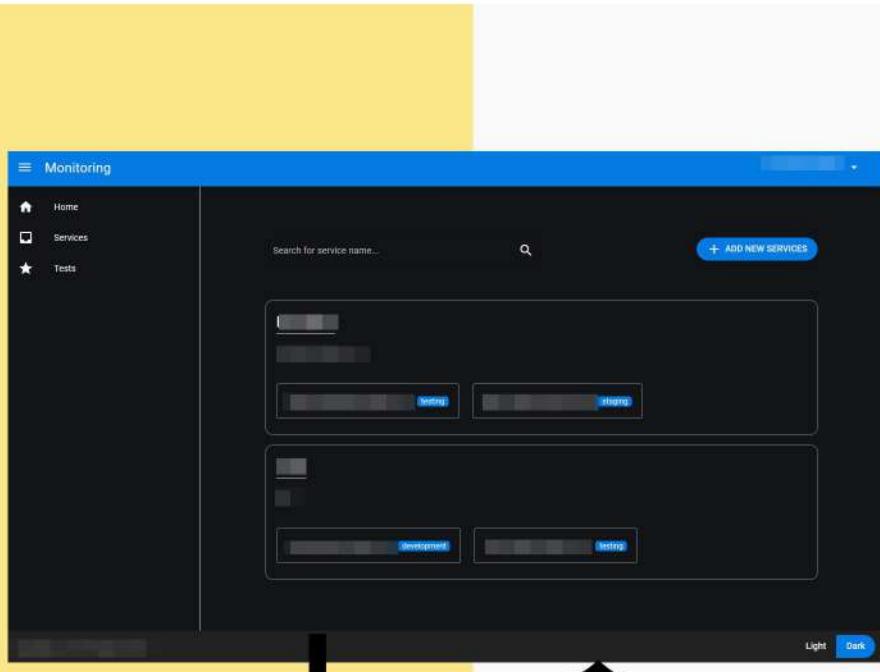




API Monitoring



三 API Monitoring



Project brief

A two months project to design and develop a simple API monitoring web-app for running recurring tests on specific API url, then catch errors. The errors are then recorded and notified to the developers via Slack, Whatsapp and Telegram. For some fun, I also added the functionality to switch between Light and Dark theme.

Platform

Web

三 API Monitoring

The screenshot displays the API Monitoring application's user interface. It features a dark-themed header bar with the title 'Monitoring'. Below the header, there are two main sections: 'ENVIRONMENTS' and 'TESTS'.
ENVIRONMENTS: This section shows a 'Setting' card with fields for 'Name' (redacted), 'Description' (redacted), and 'Environments'. Under 'Environments', there are two entries: 'Media' (Content-Type: multipart/form-data) and 'Signed' (Content-Type: application/x-www-form-urlencoded). A blue 'EDIT' button is located at the top right of the card.
TESTS: This section shows a table of test results. The columns are 'Status', 'Description', 'Headers', 'Importance', and 'Action'. There are five rows, all of which are 'Failed'. The 'Headers' column for each row includes 'Content-Type: application/json' and 'Authorization:'. The 'Importance' column for all rows is 'Critical'. The 'Action' column contains a right-pointing arrow icon. A search bar at the top of the table allows filtering by test name, status, or description. A 'RUN 1 TEST' button is located above the table. At the bottom of the table, it says '1 record selected of 5' and 'Records per page: 5 - 1 of 5'.

Roles played

Designer, Front-end developer

Tools

Development
Vue.js, Quasar

Language
HTML5, Stylus, JS (ES6)

三 API Monitoring

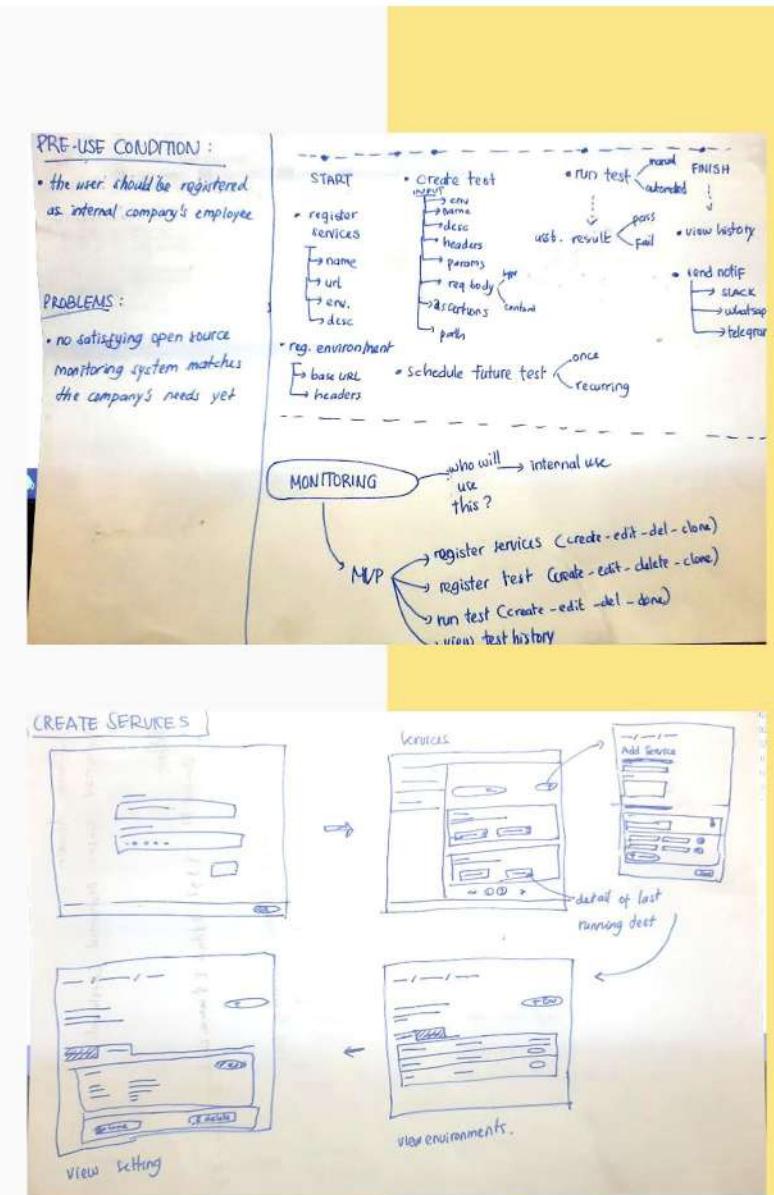
Gathering insights and analysis

After research on similar system, we concluded that:

- the user should be able to create/add/delete new services/environment for the service/test;
- each service may have > 1 environment;
- each service may have > 1 API and it may be shared between environments under the same service
- each API may be tested > 1 time
- the user should be able to view test history of each API; as well as viewing the detail of each test

Design and development

The web app was built with Quasar and Vue.js, while its styling was done with the help of Stylus preprocessor. The final product was a serverless web-app, deployed with CircleCI and hosted with S3.

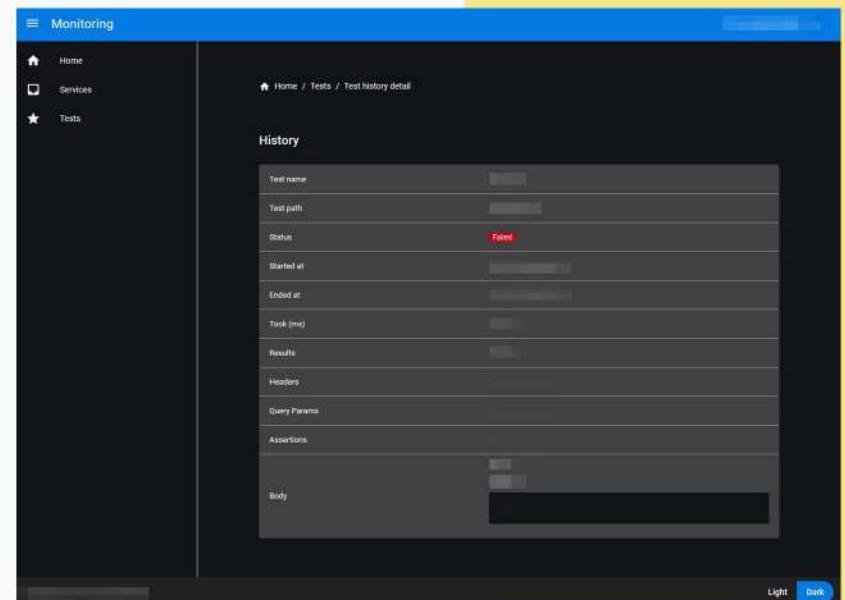


三 API Monitoring

Challenges

Every project has their own challenges, and this project was no exception. In addition to the tight deadline, it was my first time deploying the web-app directly to S3 with CircleCI by myself. This project was also my first experience with Quasar framework and Stylus preprocessor. Thankfully it was a smooth experience since both of their documentations were kept up to date.

The last challenge was playing with the dark mode toggle. I was able to update multiple nested components with the toggle change by implementing global mixins and toggling the variables.



三 API Monitoring

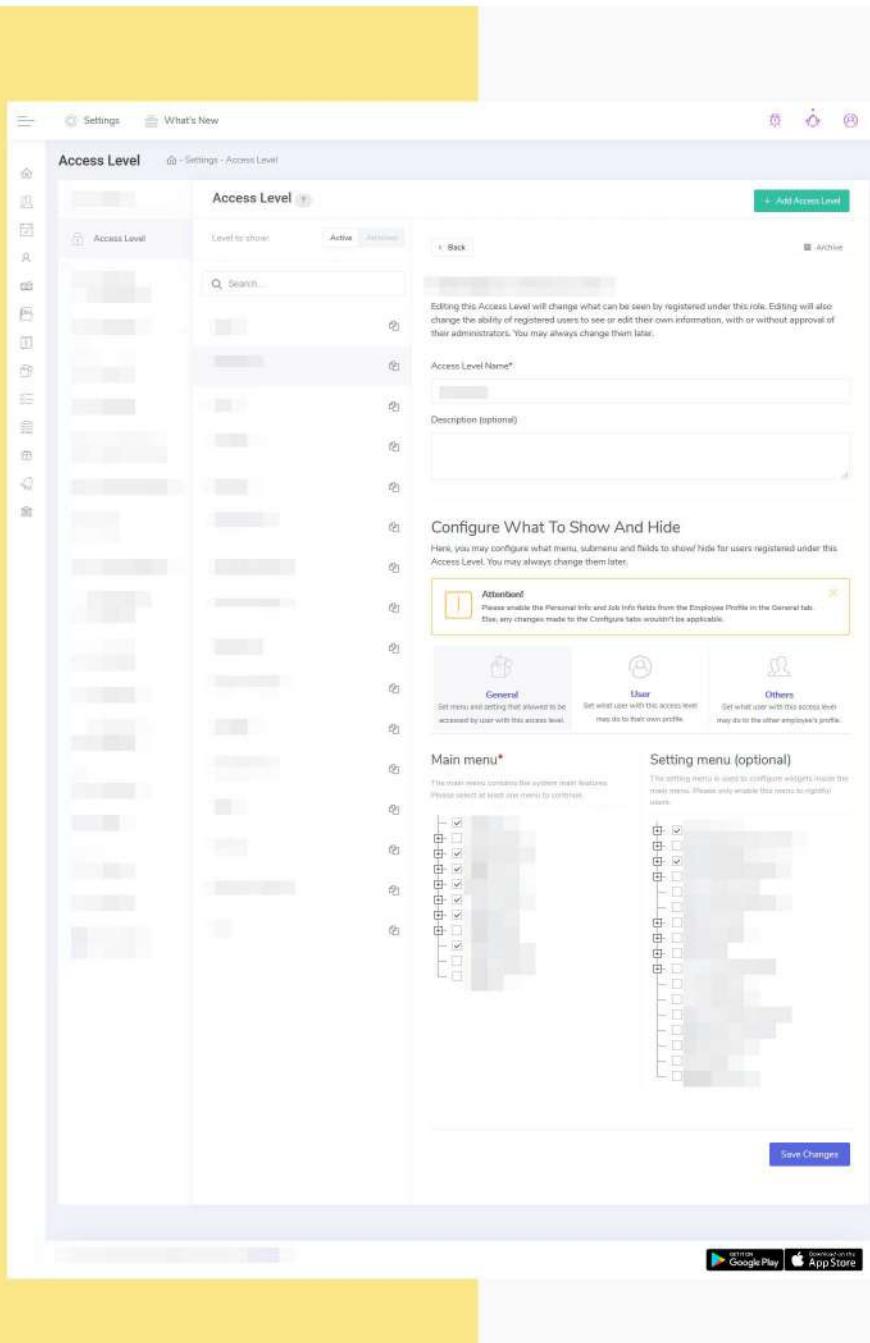




HRM Features



HRM Features: Access Level



Project brief

This project was part of a bigger project, a human resource management web app. As the name suggests, the Access Level feature is used to configure the permission of every users (employees) who are registered to the site, and what happens to them when their permission changes.

In this project, I did a major redesign on the previous menu, while also researching on how to adapt the newly designed configuration in Vue-based environment.

Platform

Web

HRM Features: Access Level

A screenshot of a web-based application interface titled "Access Level". The main title bar includes "Settings" and "What's New". Below the title, there is a breadcrumb navigation: "Access Level > Settings - Access Level". A green button labeled "Add Access Level" is visible. On the left, there is a sidebar with various icons and a list of access levels. The main content area shows a table with columns for "Access Level Name*", "Description (optional)", and "Configure What To Show And Hide". The "Configure What To Show And Hide" section contains several tabs: "General", "User", and "Others". Under "General", there is a note about enabling "Personal Info" and "Jobs Info" fields. The "User" tab shows settings for "Set what user with this access level may do to their own profile". The "Others" tab shows settings for "Set what user with this access level may do to the other employee's profile". There are also sections for "Set all fields to:" and "Select an option" with radio buttons for "Hide", "View Only", and "View and Edit".

Roles played

Designer, Front-end developer

Tools

Development
Vue.js, Bootstrap, JQuery

Language
HTML5, SCSS, JS (ES6)

HRM Features: Access Level

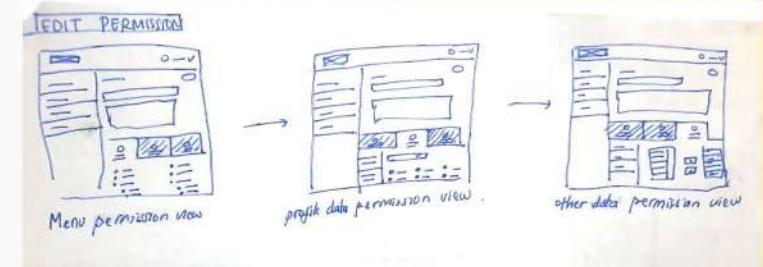
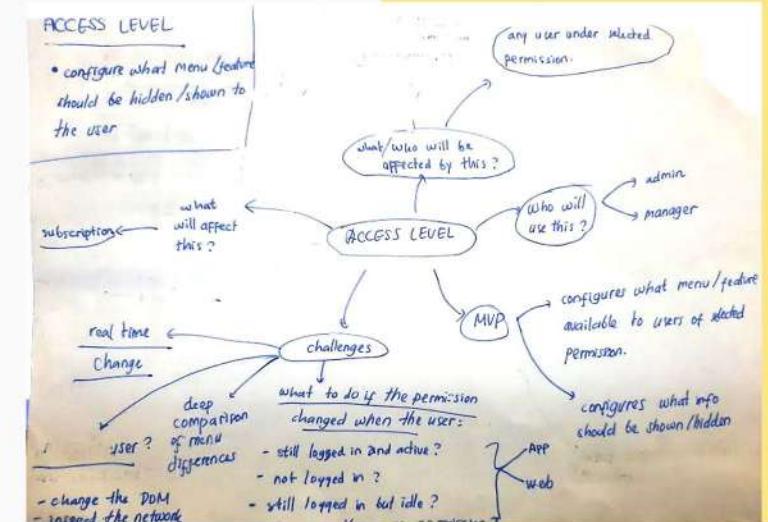
Gathering insights and analysis

Before starting on this feature, I did some research on previous permission menu. In addition to that, I also did some mini interviews with my teammates (who had never used the previous system) and lurking around the old system to fully understand its business process. I also researched on suitable plugins for the feature.

Based on these insights, I concluded that the system could be improved by separating and categorizing the setting into 3 tabs, according to its use.

Design and development

The web app was built with Bootstrap and Vue.js, while its styling was done with the help of SCSS preprocessor.

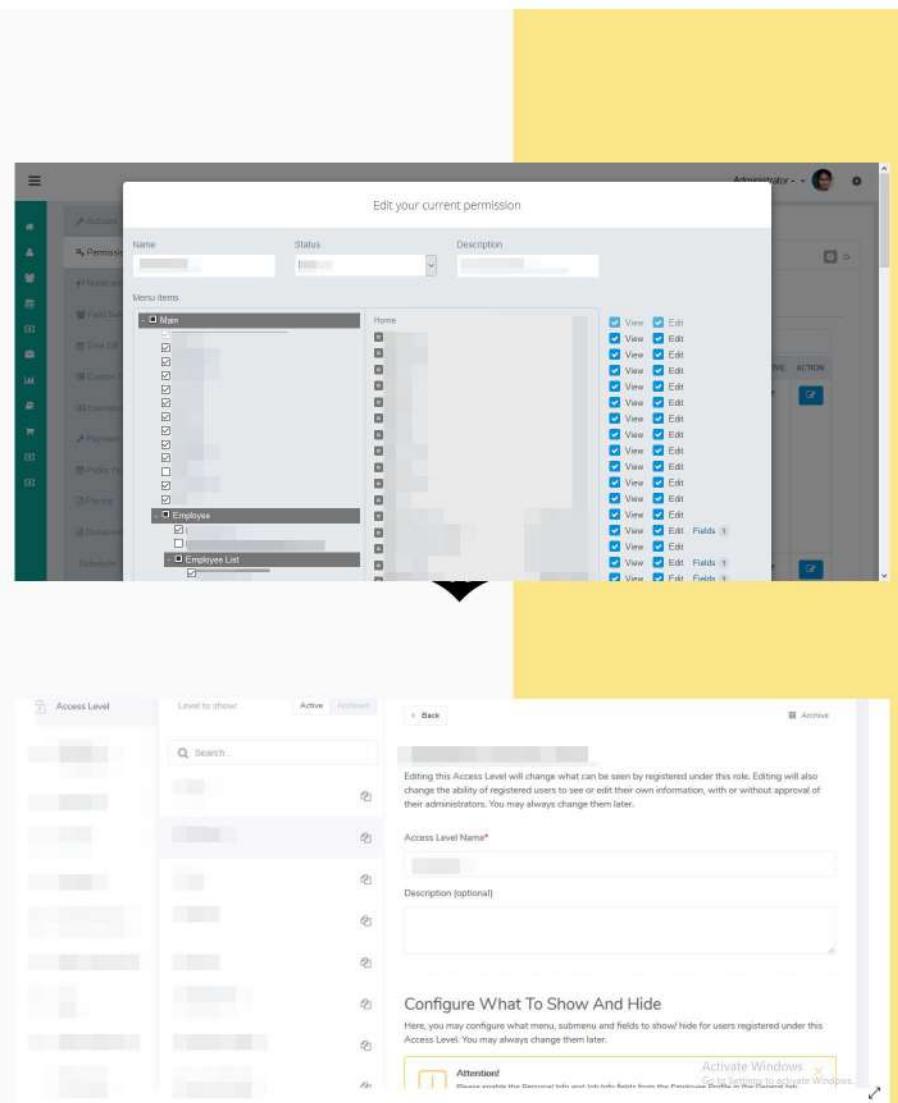


④ HRM Features: Access Level

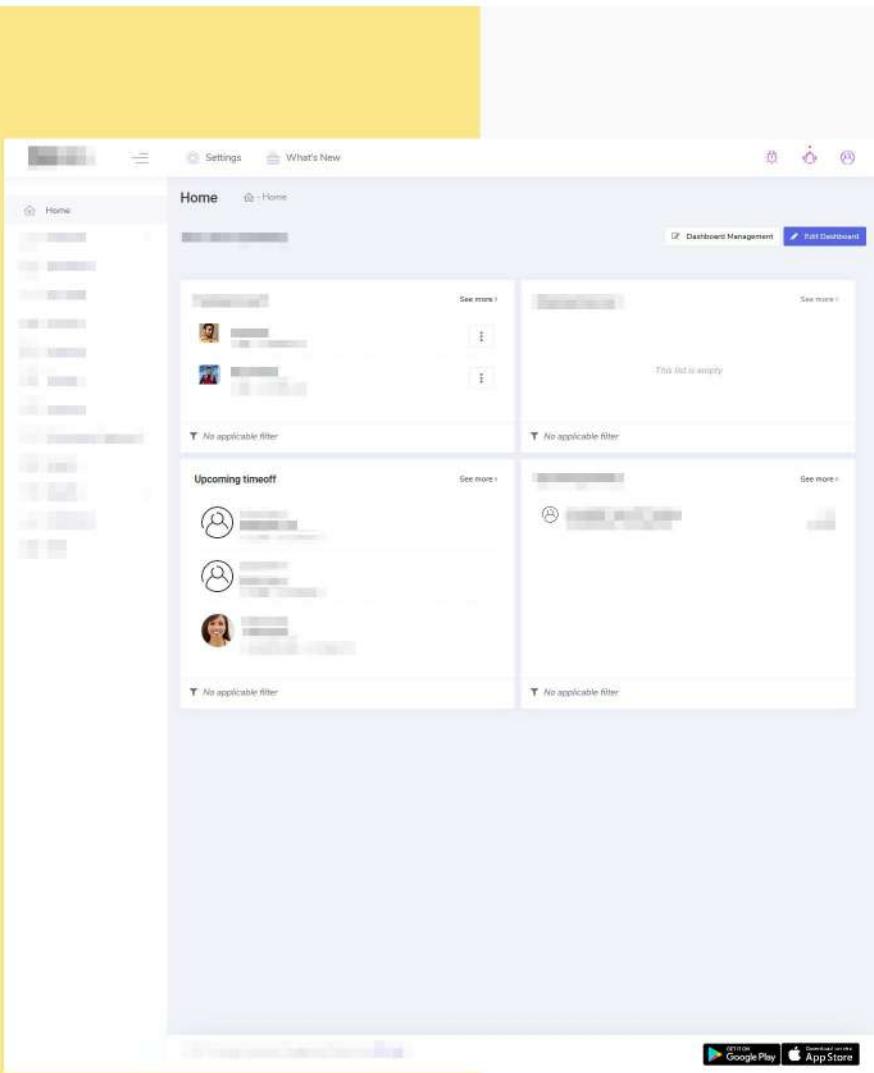
Challenges

Looking back, the greatest challenge of working on this feature was thinking of the many possibilities the user may encounter, and the many workaround an advanced user may do to bypass the configuration. I overcame this by reading the documentations to fully understand on how vue-router and Vue.js works; either in the development or the production environments. I also had to think creatively for the workarounds.

The other challenge I encountered while working on this project was manipulating the deeply nested, dynamic arrays and the dynamic configuration data. I managed to deal with it by using Lodash and native Javascript array manipulation techniques.



HRM Features: Dashboard



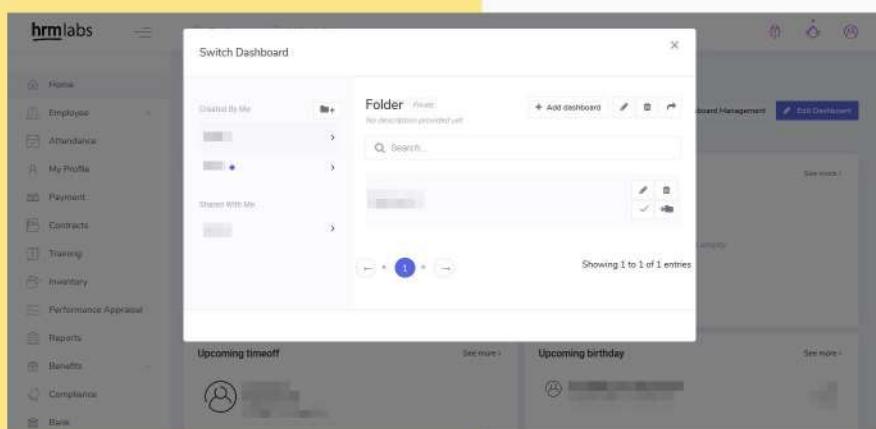
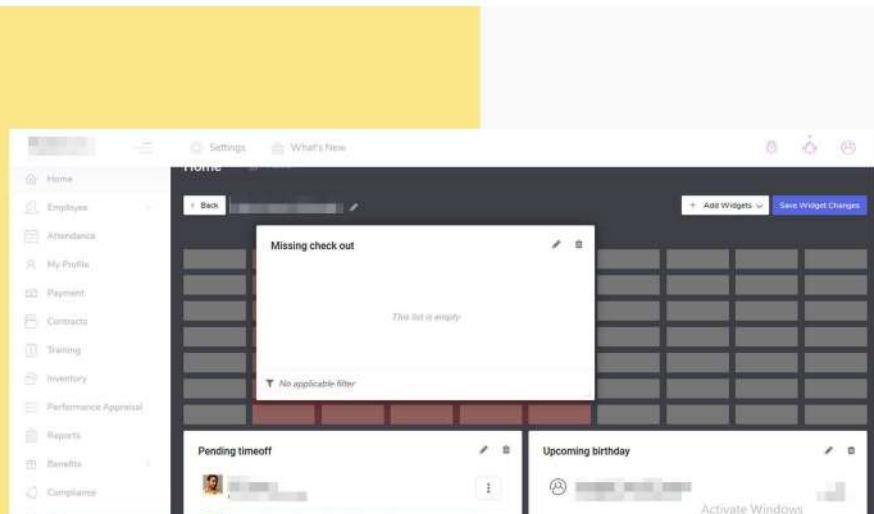
Project brief

This three-months project was also part of a bigger project, a human resource management web app. The Dashboard feature enables the user to do a quick review and simple actions on certain site features with configurable, dynamic widgets such as chart or list. The dashboard may then be saved or shared to other users under the same company.

Platform

Web

HRM Features: Dashboard



Roles played

Designer, Front-end developer

Tools

Development
Vue.js, Bootstrap, JQuery

Language
HTML5, SCSS, JS (ES6)

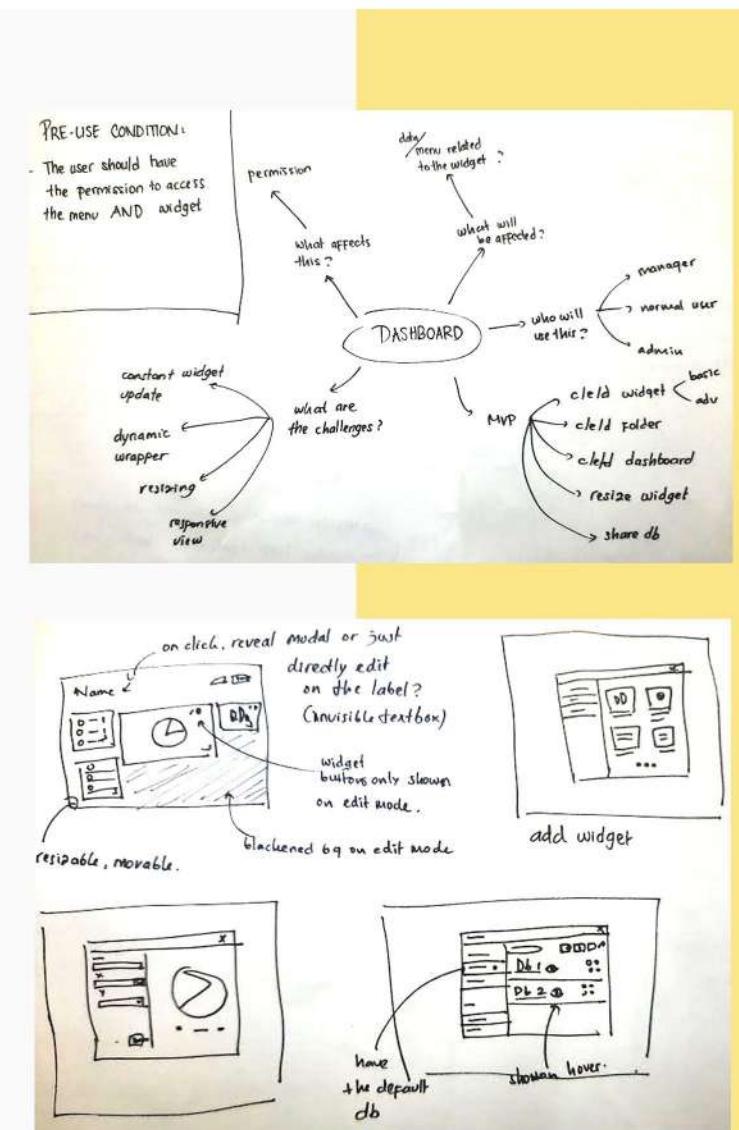
HRM Features: Dashboard

Gathering insights and analysis

Before working on the project, I did some research on similar feature from similar services, and noted down each of their advantages and disadvantages. I also did some research on suitable plugins for the feature. After that, I drew simple mind map and mockup of the Dashboard for later reference. From these researches, I was able to conclude that the user needs customizable widgets with configurable title, data source, shape (for chart) and filters. In addition to that, the widgets should also be resizable and nice to look at in the mobile site.

Design and development

The web app was built with Bootstrap and Vue.js, while its styling was done with the help of SCSS preprocessor.

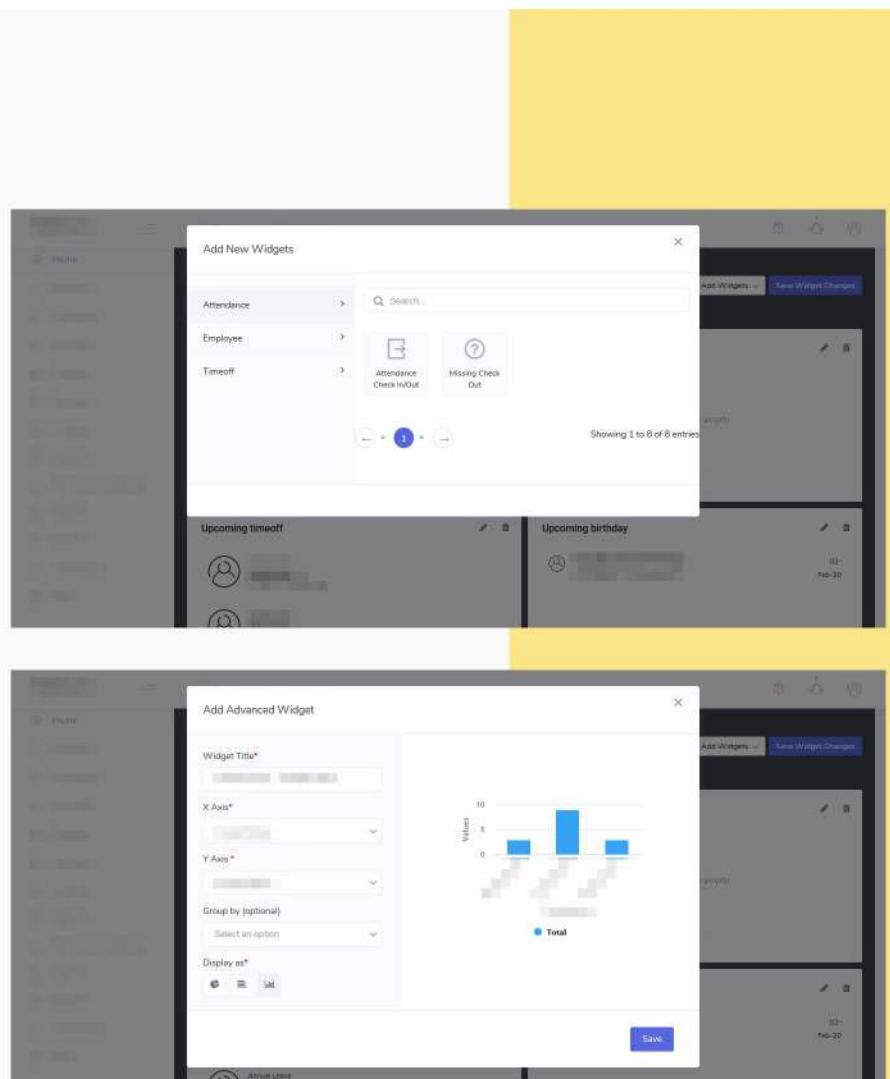


HRM Features: Dashboard

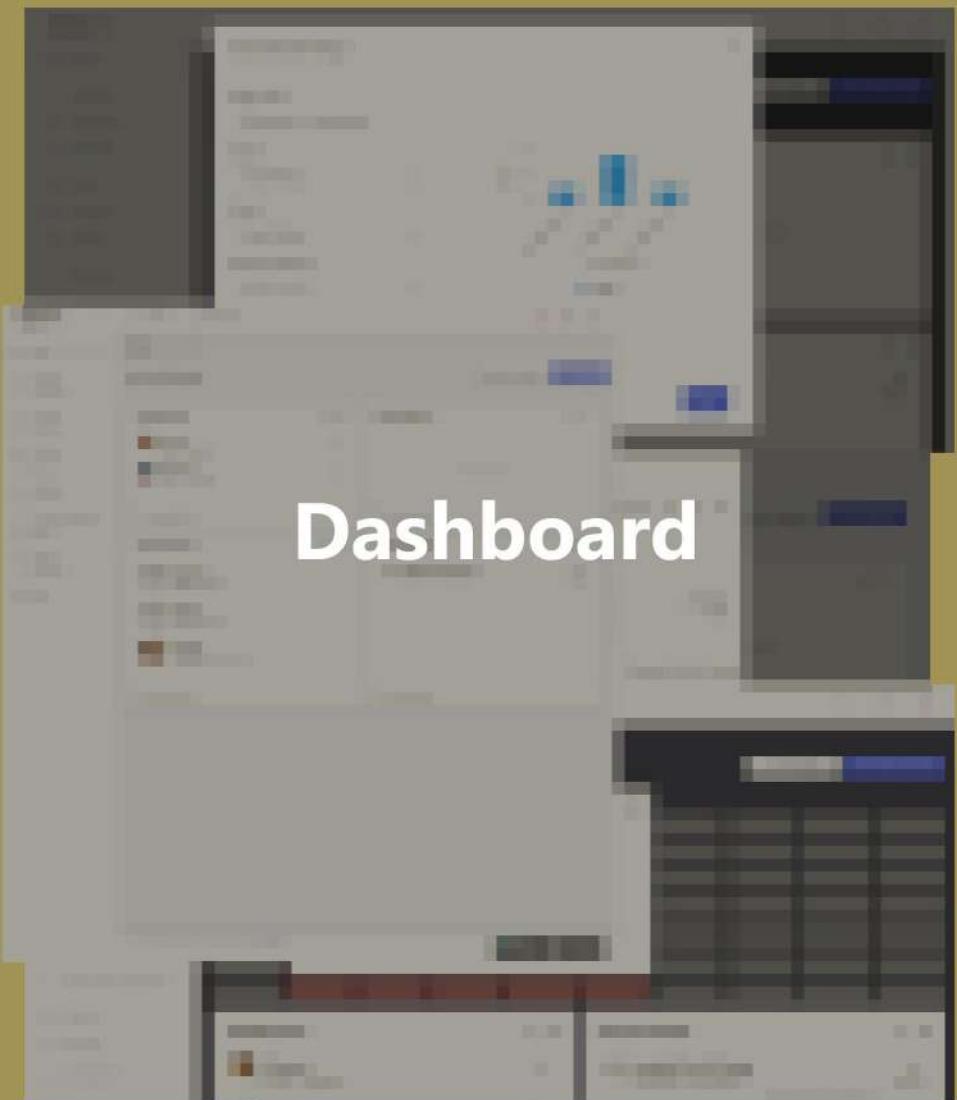
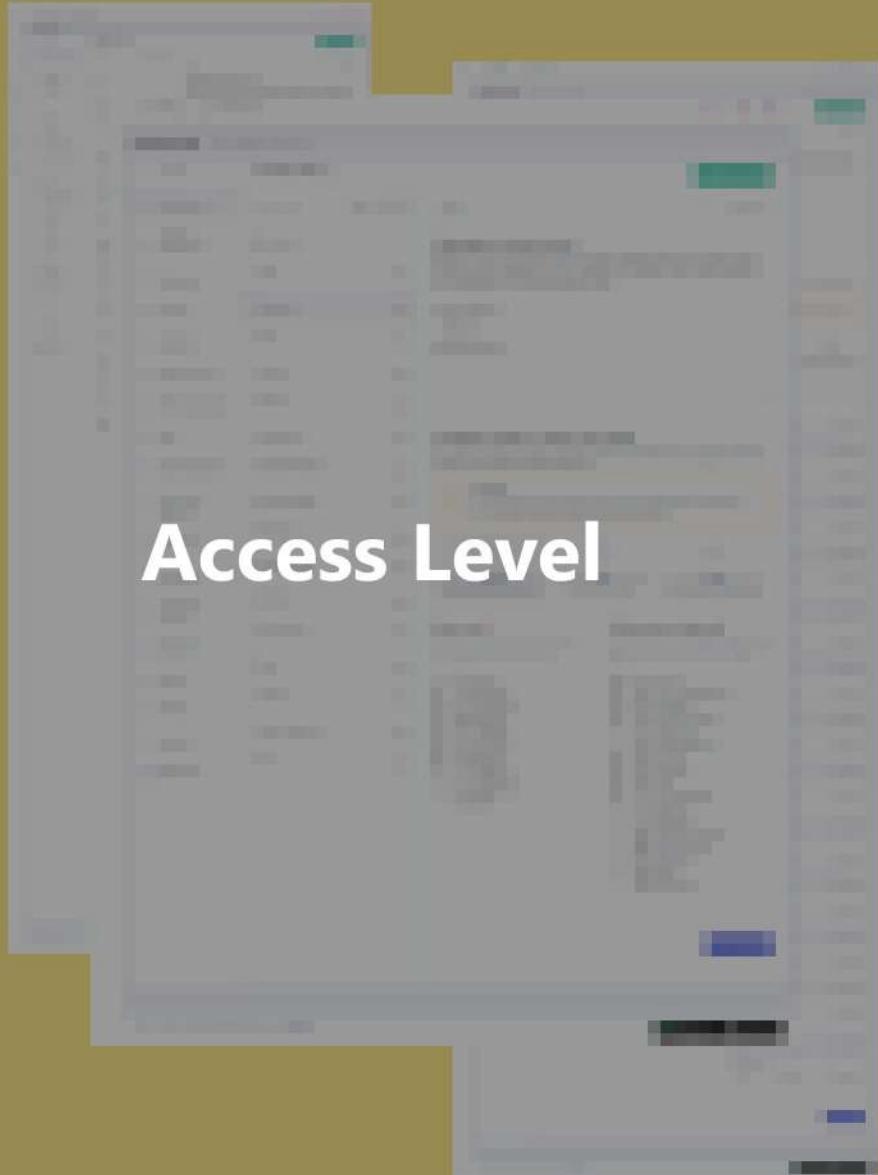
Challenges

For me, the hardest challenges of working on the Dashboard was the code structure. Since this feature was planned to include many widget types, I have to write the code as clean and dynamic as possible. I solved this by implementing the clean code principle.

The other challenge was resizing and moving the widget. It was hard because I have to update the widget coordinates and the content size/count frequently, without disrupting the flow. I overcame this by storing the coordinates and sending the updates regularly on each resize/move.



HRM Features

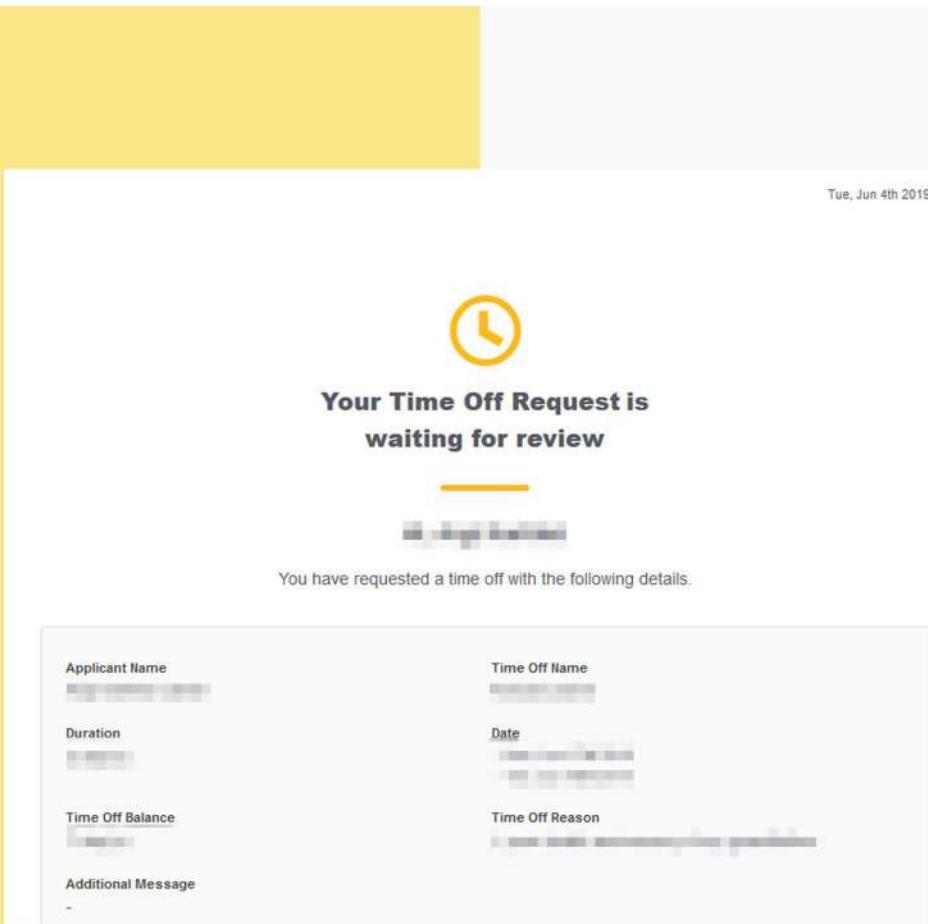


五

HRM: Others



五 HRM Features: E-mail Notification



Project brief

This one-month project was part of the Notification feature of a human resource management web app. In this project, I was challenged to design and code a lightweight e-mail template from scratch with Pug and SCSS. The final product was a custom template with cross-browser compatibility, which looks good in both mobile and desktop view.

Tools

Development
Pug, Express.js, socket.io

Language
HTML5, SCSS, JS (ES6)

五 HRM Features: User Management

The image contains two side-by-side screenshots of a web-based HRM system. Both screenshots have a yellow header bar at the top.

Screenshot 1 (Top): User Management - All Users

This screenshot shows a list of all active users. The interface includes a search bar, a table with columns for Name, Status, Username, and Action, and a "Bulk Edit" button. A modal window is partially visible in the background.

Name	Status	Username	Action
[Redacted]	[Redacted]	[Redacted]	[Redacted]
[Redacted]	[Redacted]	[Redacted]	[Redacted]
[Redacted]	[Redacted]	[Redacted]	[Redacted]
[Redacted]	[Redacted]	[Redacted]	[Redacted]
[Redacted]	[Redacted]	[Redacted]	[Redacted]

Screenshot 2 (Bottom): User Management - Generate New Users

This screenshot shows a feature for generating new users from a list of employees. It includes a search bar, a list of selected employees (2 data selected), and a "Generate 2 users" button. A modal window is displayed, showing a table with columns for Username and Action.

Username	Action
[Redacted]	[Redacted]

Project brief

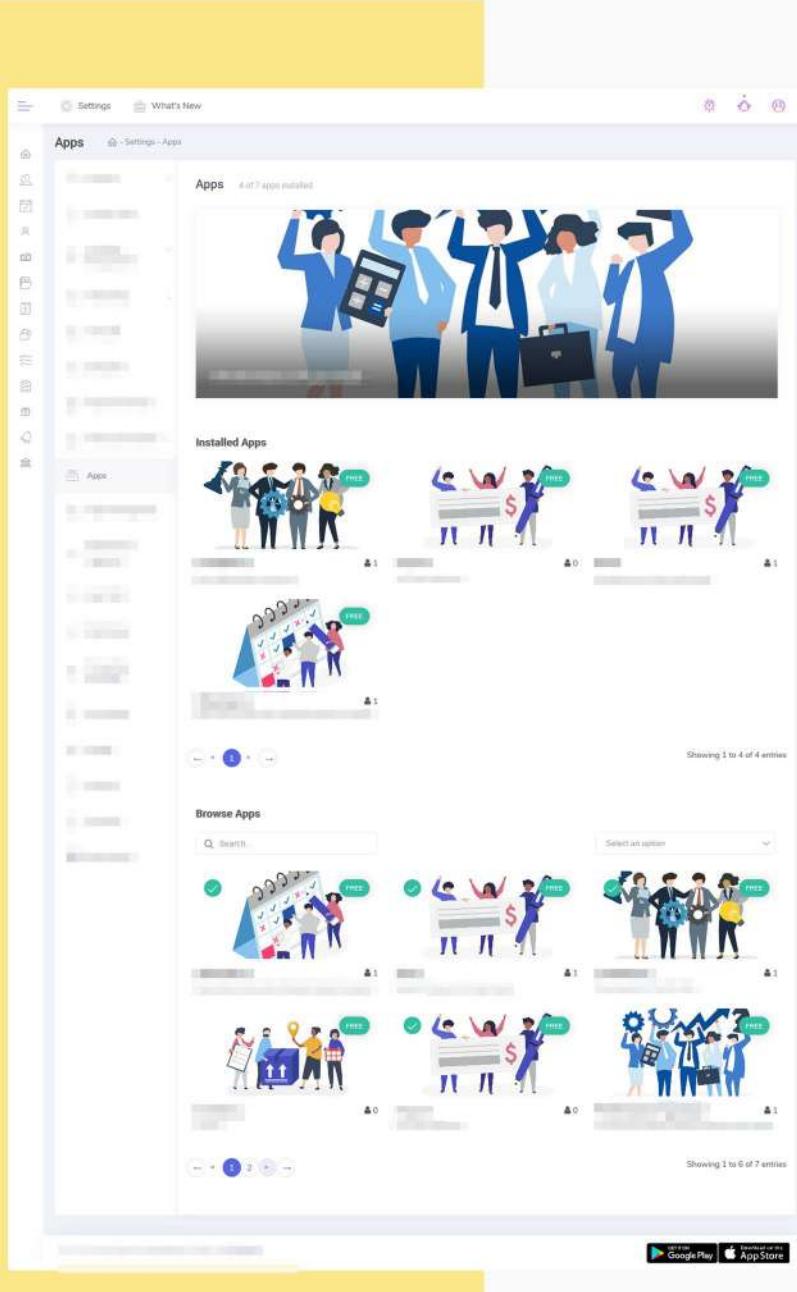
This three-months project was part of a human resource management web app. In this project, I was tasked to design and develop a user-management feature, which includes the ability to generate, edit/manage, and delete users. The feature also includes the ability to change role/permission for each user.

Tools

Development
Bootstrap, Vue.js, JQuery

Language
HTML5, SCSS, JS (ES6)

五 HRM Features: Apps



Project brief

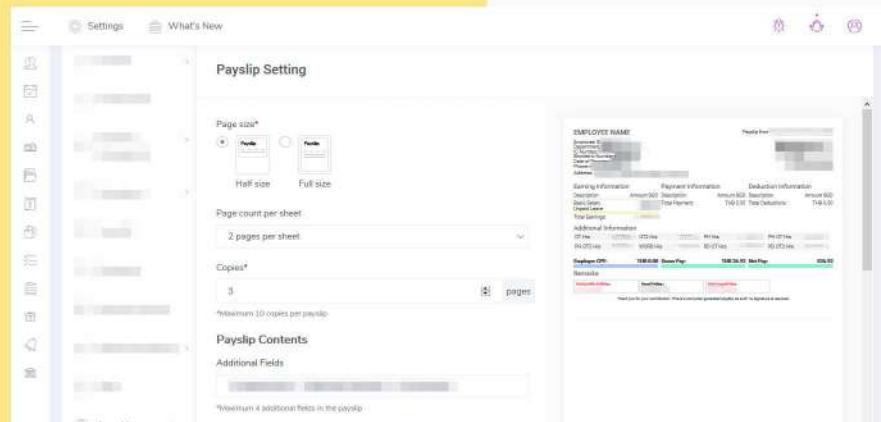
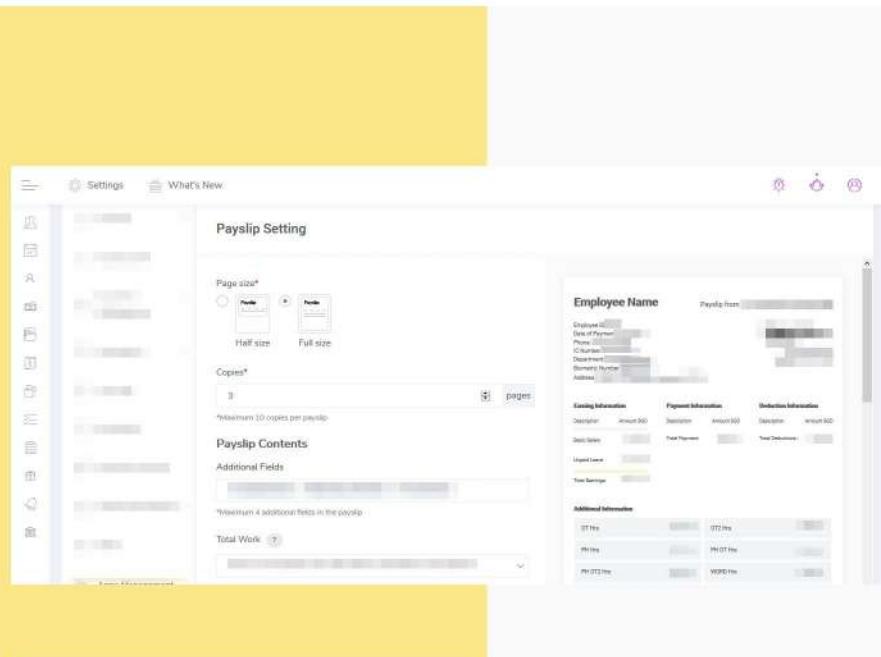
This two-months project was part of a human resource management web app. In this project, I was responsible to design and develop a mini app store for the website. Buying the apps means you will subscribe to use that app (feature) for a certain time (the time range is chosen before payment).

Tools

Development
Bootstrap, Vue.js, JQuery

Language
HTML5, SCSS, JS (ES6)

五 HRM Features: Payslip Setting



Project brief

This one-month project was part of a human resource management web app. In this project, I was responsible to design and develop a setting menu to configure the data shown in the payslip, as well as the payslip size and design. The final product was a configured and downloadable payslip for every employee.

Tools

Development
Bootstrap, Vue.js

Language
HTML5, SCSS, JS (ES6)

五 HRM Features: Documents

A screenshot of the HRM web application's document management feature. The interface includes a sidebar with navigation icons and a main content area titled 'Documents'. The main area shows a table with the following data:

File Name	File Size	Shared to	Created at	Action
[REDACTED]	20.42 KB	[REDACTED]	31-May-18	[REDACTED]
[REDACTED]	1.01 MB	All Users	15-Aug-18	[REDACTED]
[REDACTED]	1.01 MB	All Users	15-Aug-18	[REDACTED]
[REDACTED]	6.64 KB	[REDACTED]	27-Mar-19	[REDACTED]
[REDACTED]	6.64 KB	[REDACTED]	27-Mar-19	[REDACTED]

A screenshot of the 'Upload Document' modal. The modal has fields for 'Files*' (with a 'Choose file to upload' input and a 'Browse' button) and 'Share to (Optional)' (with a dropdown menu and a search bar). The background shows a preview of the 'Documents' list.

Project brief

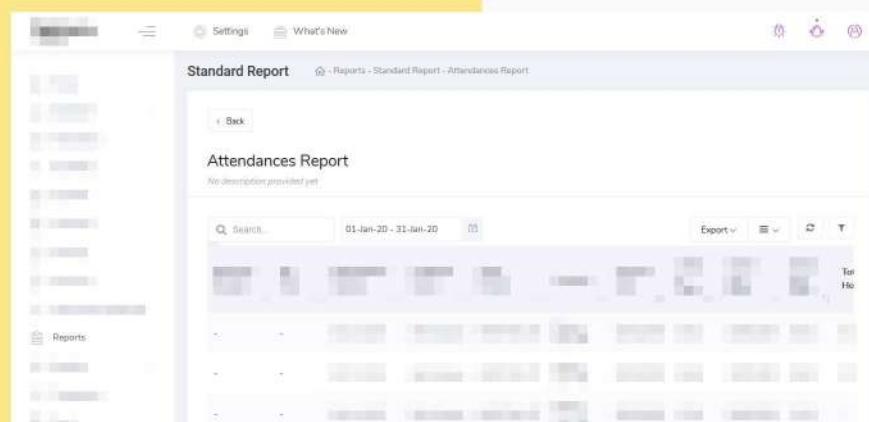
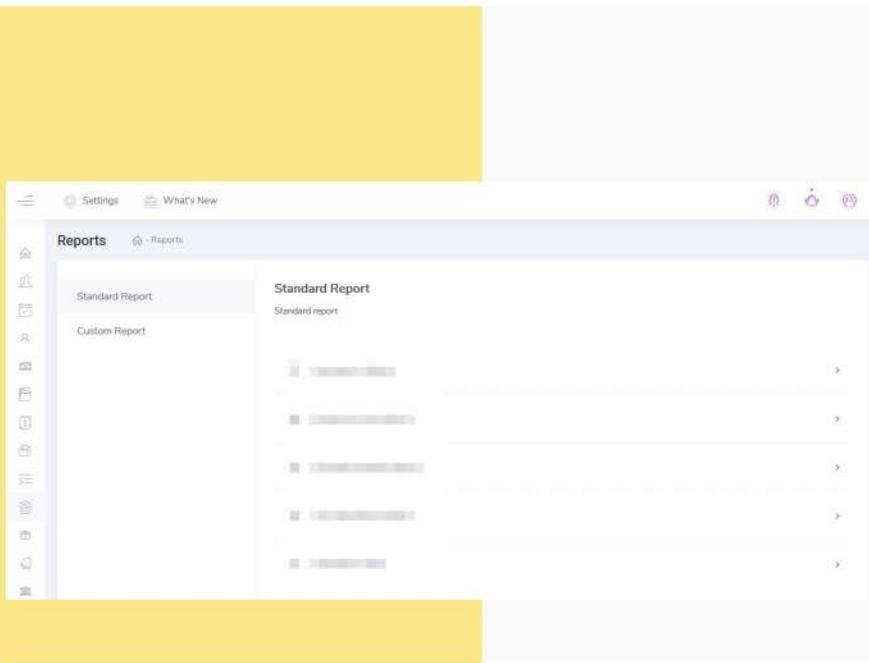
This one-month project was part of a human resource management web app. In this project, I was responsible to design and develop a feature to upload documents to Amazon S3, then share them towards employees of the same company.

Tools

Development
Bootstrap, Vue.js , JQuery

Language
HTML5, SCSS, JS (ES6)

五 HRM Features: Reports



Project brief

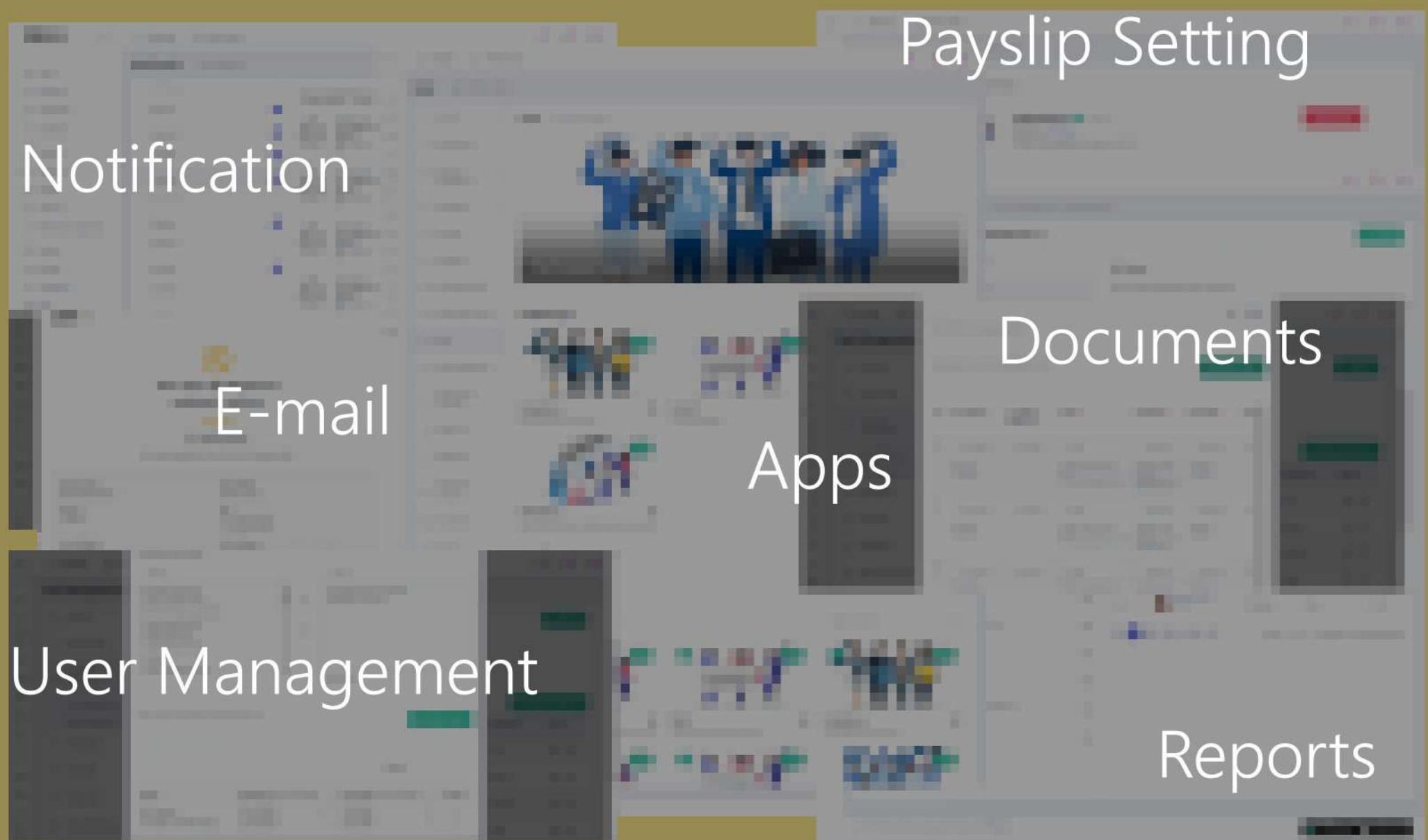
This two-months project was part of a human resource management web app. In this project, I was responsible to design and develop a feature to view dynamic reports and filters based on the report type, then download them.

Tools

Development
Bootstrap, Vue.js , JQuery

Language
HTML5, SCSS, JS (ES6)

五 HRM Features

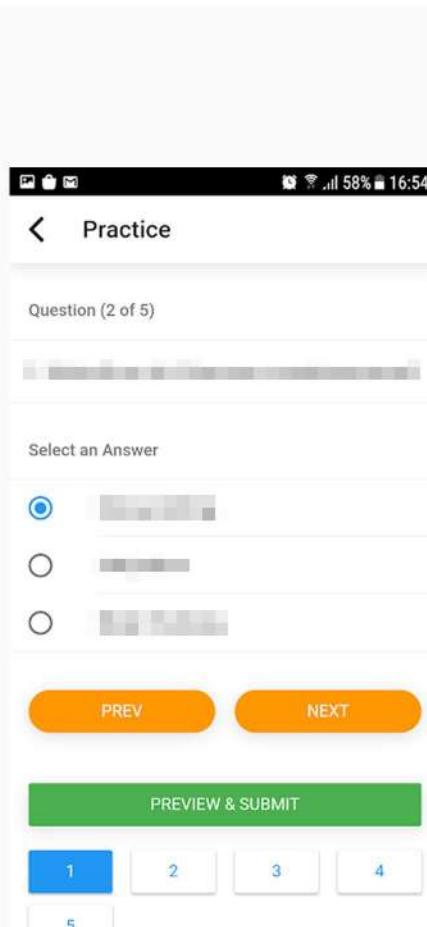
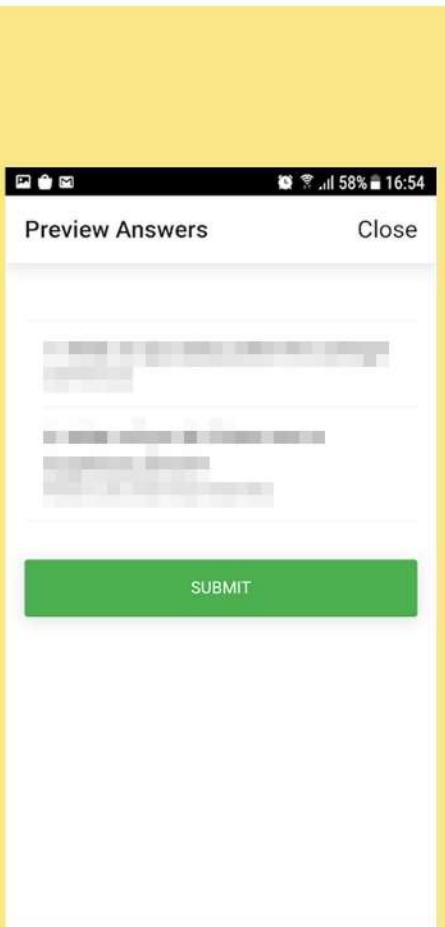




Adept: E-learning



Adept: E-learning



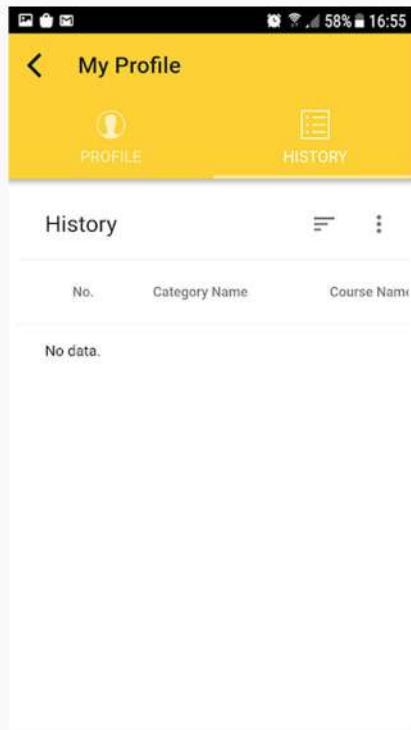
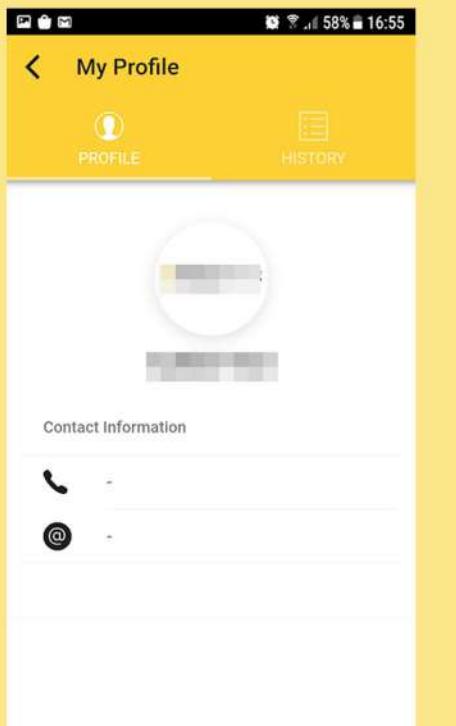
Project brief

This three-months project was part of a bigger project, a dynamic form management web app. In this project, I was responsible in developing and maintaining a mobile app that serves various courses, practices, and tests for the users. The app was developed with the help of Framework7, Cordova, xcode (for ios), and Vue.js

Platform

App (Android & IOS)

Adept: E-learning



Roles played

Developer

Tools

Development
Vue.js, Cordova, Framework7, xcode

Language
HTML5, SCSS, JS (ES6)

Adept: E-learning

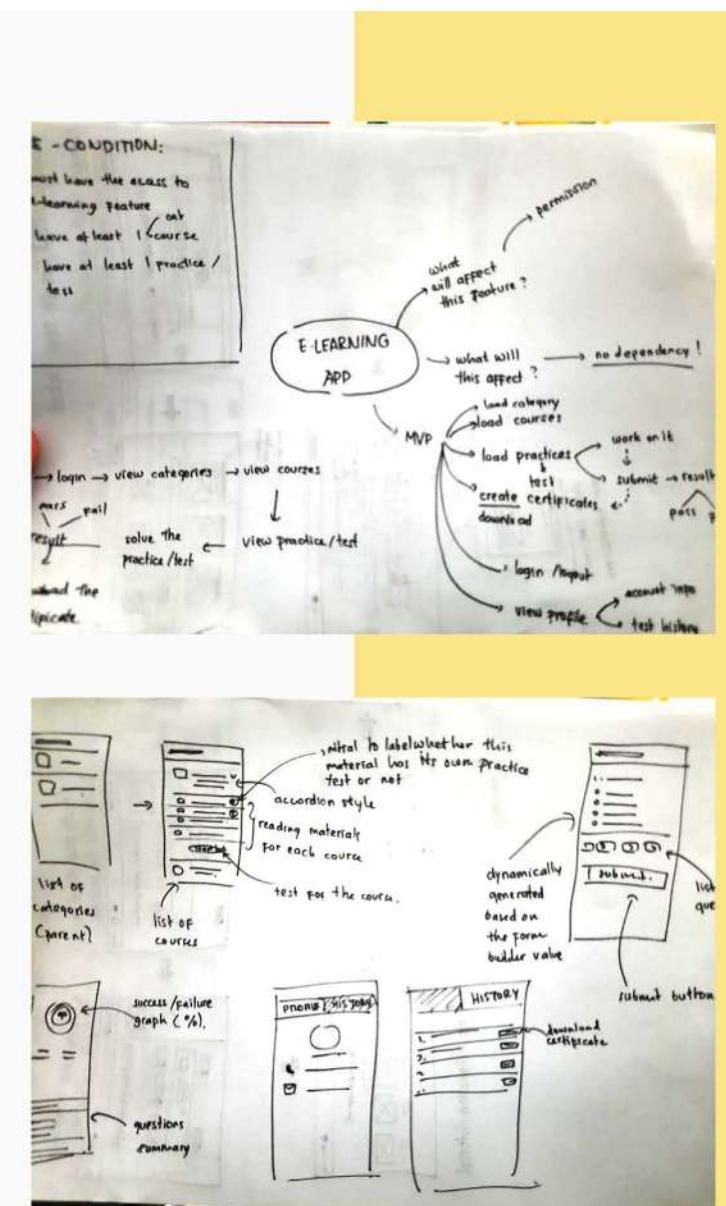
Gathering insights and analysis

Before starting on this feature, I did some researches on similar services; then write down their advantages and disadvantages and matched them with the requirement. From these insights, I was able to conclude several questions that will serve as the basis to develop the app as below:

- Who will use this app?
- What is the final purpose of using this app?
- What can the user do with this app?
- What should the user do before working on the practice test served in the app? ...etc

Design and development

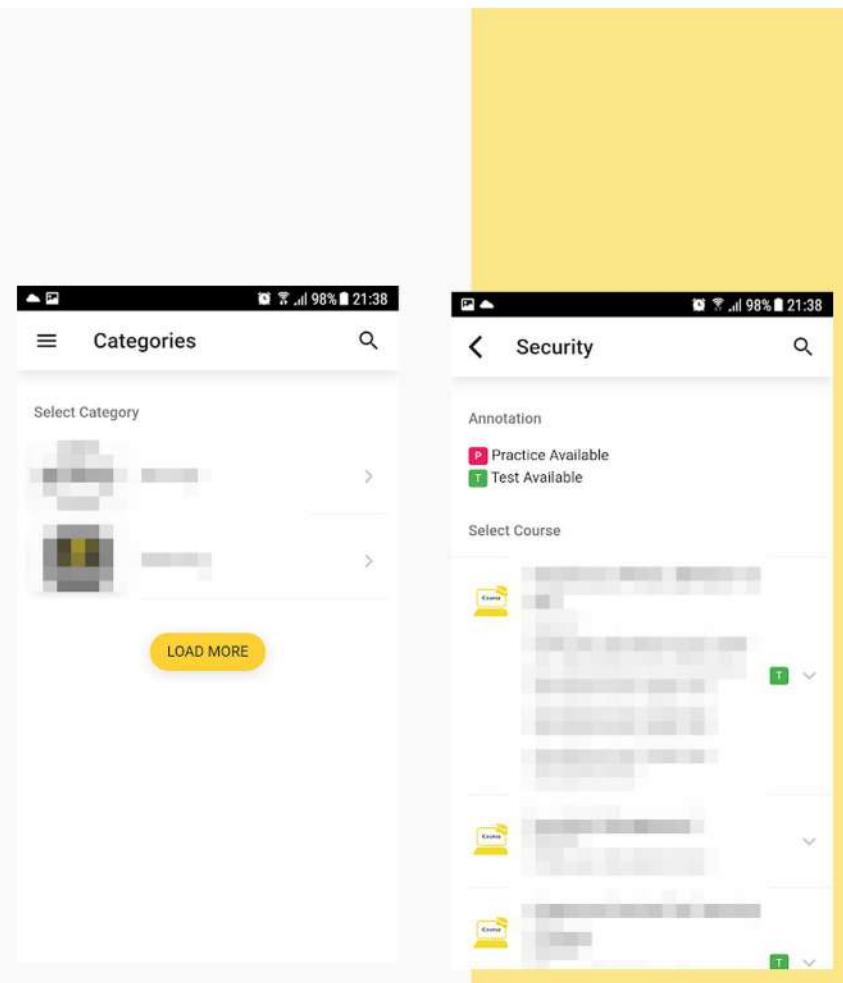
The app was built with the help of Framework7, Cordova, and Vue.js; while the styling was done with the help of SCSS preprocessor.



Adept: E-learning

Challenges

The greatest challenge of working on this app was not the development process , but the environment and the building process. Since this was my first experience working on a mobile app with Cordova, setting up the dev environment took quite some time; especially the configuration for the Mac. Yet it was exciting, because this project taught me to develop, build, release, deploy, and sign the installation package for both Android and IOS.



Adept: E-learning

PROFILE HISTORY



Admin Test

Contact Information

PROFILE HISTORY

History

No.	Category Name	Course Name
No data.		

PREV NEXT

Body checking

There is no more true than this

SUBMIT

Understanding and remembering

1. Do you know how many extinguishers you have?



Select Category

SECURITY



LOAD MORE

Annotation

P Practice Available

T Test Available

Select Course



Score: 20%



Thank you for completing your practice.

七

Misc: Illustration



7 Misc: Illustration



Project brief

When I'm not coding or reading articles about development, I love to draw, read, and write about ancient Japanese culture. Most of them are drawn digitally with Photoshop and Paint Tool SAI, but sometimes I also do traditional work with watercolor and colored pencil.

Tools

Digital
Photoshop, Paint Tool SAI

Traditional
Watercolor, Colored pencil

終了
Thank you!

You may view my previous
porto at bit.ly/portoArgi2018

