

LUMA with OOUX

The combination of Object-Oriented User Experience (OOUX), LUMA forms a comprehensive system for addressing, delivering, and maintaining complex and competing requirements in the design and development process.

Team Lead: Michael Long (UX Delivery Associate Director, Design Director)

Devs: Diego Armando, Lucas de Melo, Arthur Annibal

Designers: Julieta Poy, Jesica Nuñez, Michelle Borgström

PM: Marcelo Clarens

Considering Code: While the focus of this paper is on OOUX systems, it also extensively discusses code-related matters. The reason being, this manuscript specifically addresses the development and application of a design library that is intricately linked to themed Material UI components in a codebase.

The Issue: Previous software solutions have displayed a lack of professional rigor, falling short of meeting intricate universal requisites, and demonstrating suboptimal performance.

- The development process is unable to faithfully implement or construct the original designs
- The resulting products lack accessibility, hindering their usability
- The progress of development is impeded by the burdensome need to create individualized versions for both mobile and desktop platforms
- The software solutions are not compatible with older browsers, limiting their accessibility across diverse user groups
- The performance of the software products is impaired due to an accumulation of unplanned patches and workarounds during the development phase
- Despite being developed on robust, professional frameworks like React Material UI and React Bootstrap, which are designed to accommodate complex requirements such as accessibility and responsiveness, these essential features are conspicuously absent in the final product
- Product language localization is not considered until the final stages of development where it creates considerable issues

If the final software looks significantly different from the initial designs and you're spending time at the end of development fixing issues and addressing complex requirements, using the methods talked about here will help solve your problems.

Challenge Statement: Our objective is to merge UI/UX and Front-End Development with a holistic framework that caters to intricate UX demands, such as usability. The focus is on facilitating the creation of top-notch, efficient products, while also curbing engineering costs to optimize time and financial resources.

Problem Statement: When looking back at past work, we noticed that our initial, simple design drafts did not align with Material UI's standard design patterns. These drafts were supposed to be transformed into standardized forms and graphics that would work well on both desktop and mobile devices. Unfortunately, we overlooked this mismatch when moving from simple drafts to more detailed designs. As a result, our team unintentionally made decisions that strayed too far from the framework, proving challenging to correct at the end of the process.

Our decision to implement an OOUX strategy paired with Material UI stems from the need to construct a web based application using fundamental web components such as buttons, forms, and graphs. It's essential to note, however, that for less conventional interfaces — like XR, video games, art tools, or any interface requiring more than the standard web components — the OOUX approach is not recommended.

Strategy Statement: Our suggested plan for a smooth design and development process involves providing designers with a toolkit in Figma. This toolkit mirrors our company's Material UI design style (colors, typography, etc), components, and charts, complete with component states and requirements. This strategy meets Material UI's high standards and makes sure every component is accessible and responsive.

By giving designers a set of Figma components that strictly follow these guidelines, we can shape our product more effectively. This method ensures we stick to the Material UI framework and meet all user experience needs. In the past, designers have created "design systems" without considering the technology underneath. But when we tried to get these "design systems" up and running, we encountered bugs, failed to meet accessibility needs, and ended up with a less professional product.

Setup OOUX Design & Development:

The designer and developer will build a Material UI theme within an existing library, making it easier to apply the company's brand to all components. This can be done simply, even in complex systems. There are plenty of excellent Material UI toolkits available online. These toolkits have clean, well-documented codebases and can be licensed for use. It's crucial to match the design and the codebase. We looked at several commercial libraries in detail to pick the best resources. To improve these systems, we could consider adding the DevExpress Table plugin and the ChartJS Graphing library or other similar libraries.

Resources:

<p>Material UI https://mui.com/store/</p> <p>React Bootstrap https://react-bootstrap.netlify.app/</p> <p>Semantic UI https://react.semantic-ui.com/</p>	<p>Material UI</p> <p>Great resources to start with Material UI: https://mui.com/store/</p> <p>Combining resources from different dashboards is often the best approach.</p> <p>Material UI Theming https://mui.com/material-ui/customization/theming/</p>
---	--

UI

Synchronizing Design and Development with a Theme

Design Task: We are starting with the corporate look and feel. We need to create a theme for our React Material UI library. Our goal is to work with development so that we can have a completed Material UI theme IN CODE based on the following design theme.



Many Figma files already include all these Material UI variables and components. These match the coded components represented in design kits. So, the designer's first job is to set up basics like fonts, colors, and corners.

Note for Developers: Developers, you are presented with two choices here. First, you can encapsulate all components within a library, which will then act as a dependency. Alternatively, you can opt to utilize a dashboard filled with a variety of coded components. Creating and maintaining a library that works as a dependency might appear more sophisticated, but it can become problematic if the dependencies span across multiple applications. This is because developers frequently resort to makeshift solutions in the library application, inadvertently generating issues where automatic updates are superseded by bugs appearing in other applications reliant on the same library.

Personally, I favor a basic dashboard wherein the components are presented but not incorporated into a library. Although these dashboards usually call for multiple dependencies, they are simpler to dissect and demand fewer corrections. We can encapsulate and break apart the library later into smaller dependencies. If you have ever attempted to update an application to a newer version of an underlying framework (like Material UI), it still requires some time. The resources for a small company to accomplish the same task smoothly while maintaining complex requirements can quickly get out of control. Even well tested frameworks built with considerable resources and the best developers find it difficult to create smooth upgrade cycles.

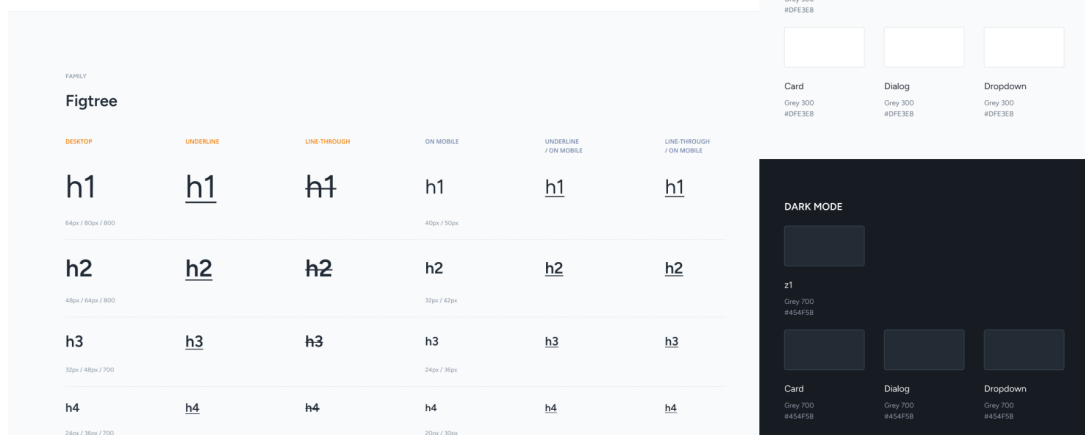
So, our first step is to transfer the basic colors and typography from our design into the six basic Material UI variables (primary, secondary, info, error, warning, success)

Figma After Color and Typography Added

COLOR

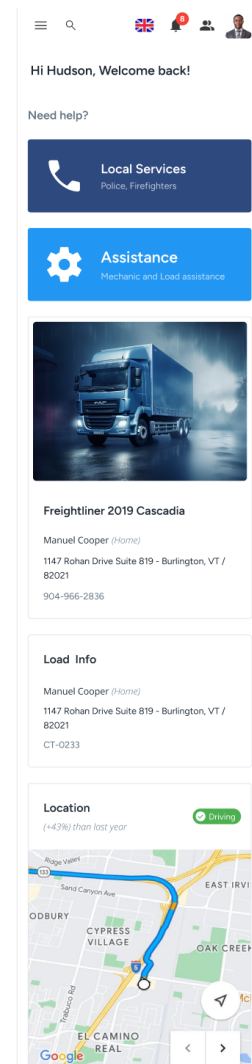
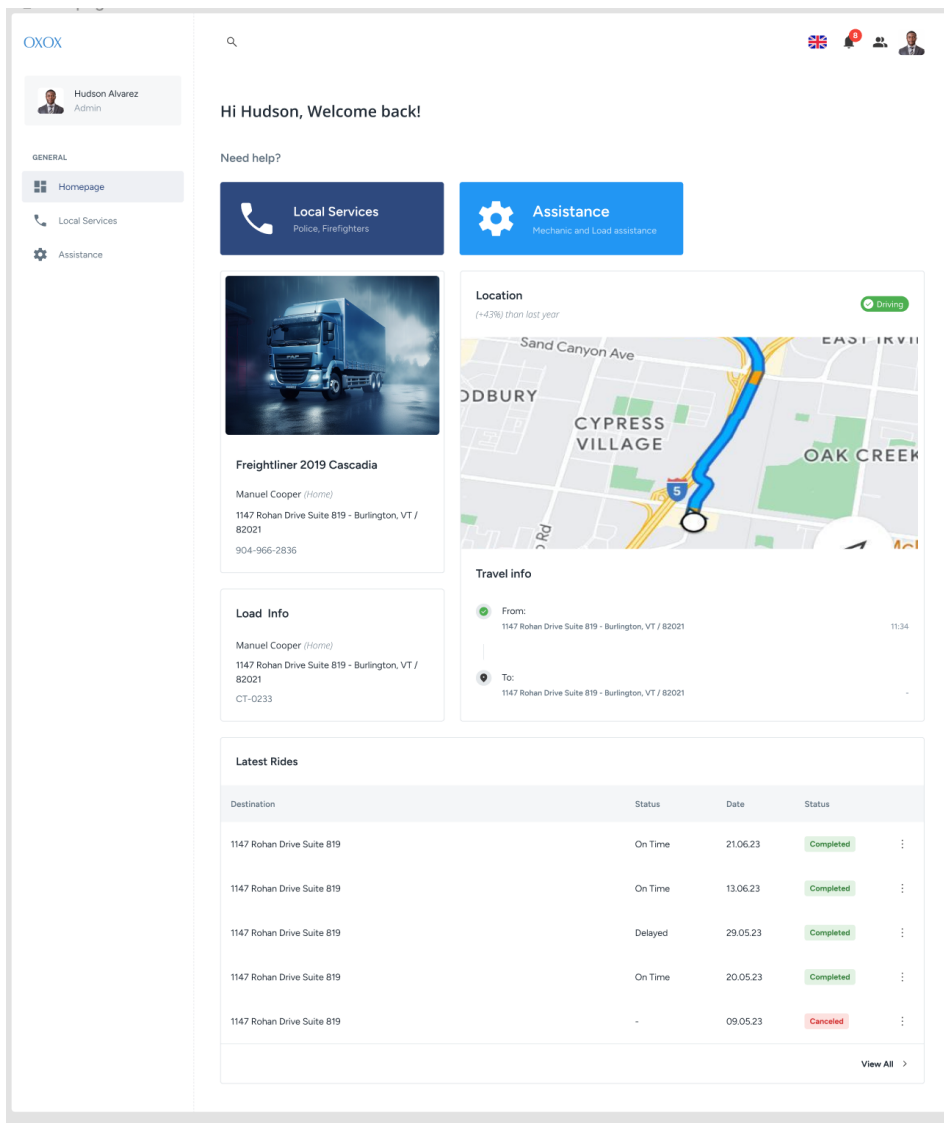


TYPOGRAPHY



The practical implications of this approach are readily apparent in the end product. Alterations made to these variables in Figma are visible in individual components and layouts, thus providing real-time visualization of the changes. Many of these systems come equipped with

an array of layouts, which can serve as excellent references for your design, offering a coherent and uniform design across all user interfaces.



The subsequent step involves translating these variables into code. This process is relatively straightforward due to the consistent use of the same variables across both Figma and our coding environment. Consequently, we just need to adjust the Material UI theme file in the codebase to reflect these changes. This ensures that any alterations made in the design will be accurately represented in the final product, promoting a cohesive user experience across all platforms.

Create React Material UI Theme

Altering a React Material UI Theme to reflect your corporate aesthetic is straightforward and can be accomplished in a few steps. All modifications are made in the theme configuration file, typically referred to as "theme.js" in most React Material UI projects.

The image is a composite of three parts illustrating the theme development process:

- Code Snippets:** Three blocks of JavaScript code defining color palettes for 'orange', 'red', and 'dark' themes. Each block lists variables like 'name', 'lighter', 'light', 'main', 'dark', 'darker', and 'contrastText' with corresponding hex color values.
- Dashboard Screenshots:** A collection of UI components including a donut chart showing 'Total 188,245' with a legend for Mac, Window, iOS, and Android; a 'Sale By Gender' donut chart showing 'Total 10,989'; and four data cards displaying user counts: 73.90k, 113.90k, 86k, and 8.60k.
- User Table:** A table with columns for Avatar, Name, Email, Last login, Rating, Status, isAdmin, and Performance. It lists 12 users with their respective details and performance percentages.

The Cyclical Theme Development Process: Bridging Design and Development

The most effective way to craft a theme is through phased development. Separating the theme development process into phases and goals can expedite your design process. Each

goal should be jointly accomplished by the designer and developer before transitioning to the subsequent phase. In many instances, the "Theme File" phase might suffice for implementing a brand. The following outlines our working methodology within the Material UI theming system:

Goal 1: **Theme File**

The Material UI theme file encompasses all fundamental variables such as color, spacing, shadows, and more. It's commonplace for the app to toggle between theme files to produce varying aesthetics, such as dark and light versions, each encapsulated in distinct theme files.

Goal 2: **Global Overrides**

Global overrides can be implemented to impact all components simultaneously. For example, if there's a requirement to alter all textboxes concurrently, this modification should be executed through a global override.

Goal 3: **Component-Level Overrides**

There could be instances where a specific component like a textbox in the search menu needs to have a unique look. In such scenarios, it's necessary to introduce an override at the individual component level to achieve this variation.

By adopting this phased approach to theme development, we can maintain a balance between overall design consistency and the need for specific customizations, leading to a more harmonious and user-friendly interface.

Tips for Efficient Theme Development

Throughout our journey, we've built intricate themes for leading Fortune 500 companies using this process and successfully fitted diverse brands into the Material UI framework. However, we did encounter some missteps which led us to refine our approach. Here are key insights on how the process should be managed:

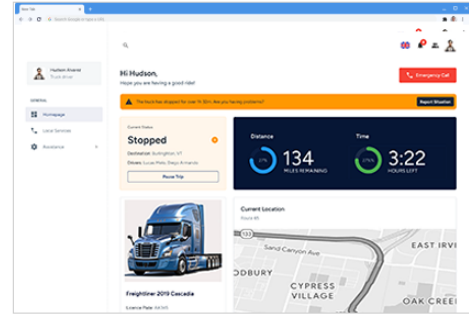
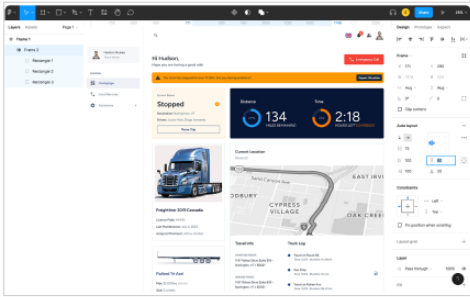
1. Material UI utilizes an array of lighter and darker hues within its color variables. While this isn't a strict requirement, it's crucial to understand that not all brands align neatly with this lighter and darker system. Instead, directly assigning colors to variables often proves to be a more efficient approach. It is recommended to maintain a base theme that accommodates both systems for flexibility, but modifying every component to fit this structure can be time-consuming.

2. Material UI components typically expect six fundamental color variables (primary, secondary, error, info, warning, and success). From a designer's standpoint, it is important to consider your design within this framework. Creating new variables or repurposing colors such as using the red error color elsewhere may render your component illogical, potentially leading to increased engineering costs.
3. Testing your design immediately by integrating your variables directly into the code is key. This allows you to review all components (buttons, pages, settings, forms, etc.) to ensure any issues are promptly addressed. This is of utmost importance - if you can't immediately align your codebase with what you've designed in Figma, you might be working with poorly structured resources and may need to find alternative options.
4. Your color variables will impact your accessibility score. Attention must be given to the contrast between the background and foreground to ensure legibility, and the coded result should be tested promptly.
5. Your coded and themed Material UI Design system should serve as the "source of truth" for both design and development. Figma should be used to design from real components, rather than deriving real components from Figma. For complex components with infinite configurations, such as tables, designers should clearly indicate the state or configuration they are utilizing in the design. Deviations from the framework are acceptable as long as the consequent engineering costs are acknowledged and reviewed with the engineering team in advance.
6. There could be areas that fall outside the OOUX framework and need customized design. For instance, a unique feature like a facial recognition video component may require base components (atoms) that do not currently exist. While this is generally a small portion of most modern web applications, tracking these areas is vital as they could necessitate substantial engineering resources.

Goal for UI Design System

FIGMA DESIGN LIBRARY IS
100% ACCURATE TO CODE

IN ALL STATES AND CONFIGURATIONS



Requirements Complete from UI phase:

- Mobile
- Accessibility
- Responsive
- Old Browser Support (Polyfills)
- Language Localization
- IOs and Android (iOS) Support
- Brand Styles
- Component States already exist
- Component's already support all states shown in design (important for tables)

The theme UI phase is about creating the framework system for fast and very accurate (mobile, accessible, etc...) prototyping in later phases.

UX

Building from your theme

**Business Need - Design Thinking, LUMA
Solve for client, business area, create a new
component.**

Armed with a strong design library in both code and Figma, which matches the brand, covers a wide range of needs, and syncs with coded Material UI components, we're ready to go. Because our Figma designs are based on coded Material UI parts, we know our design system can handle complex needs.

Why OOUX Leads to Better UX

Think about all the unique and often conflicting requirements each design change involves. Starting with a coded Material UI theme mirrored in Figma and then designing an application using these pre-coded Figma parts, makes the process much quicker and simpler.

How OOUX Simplifies the Designer's Job

OOUX makes the designer's job easier by letting them concentrate more on the end-user and unique features that will make the product stand out. Basic components and proven design patterns, like login processes or simple user settings, are already well-tested and outlined.

In our trucker app, special features like our AI chatbot mechanic, clear display of the truck's status, and a direct link to the mechanic are what will make our app special. These unique features, along with the efficient design processes OOUX provides, will make our app not only practical but also easy to use and competitive in the market.



"Are the user interface components intuitive and user-friendly across all devices?"

"How will the UI handle data fetching and display loading states?"

"Are the UI components modular and reusable?"

"What is the error handling strategy and how will the design incorporate error messages and alerts?"

"What's the strategy for handling hover states in a touch screen environment?"

"How should the design reflect system statuses, like loading, progress, or completed actions?"

"Are all potential states of each component considered in the design?"

"Does the design support multiple languages or potential localization needs?"



Focusing Designers on User Needs and Workflows:

This OOUX approach enables designers to concentrate on user needs and workflows rather than recreating standardized components like buttons, form fields, tables, and established design patterns.

Streamlining the Process for Developers:

Because the design is closely integrated with the development, developers can efficiently build, modify, and maintain our application. This synchronization frees up time to focus on critical features. In the context of our trucker application, most engineering effort will be dedicated to the AI chatbot and the real-time status of the truck and load.

Facilitating LUMA Design Exercises:

With a simplified design library at our disposal, we can initiate our Trucker Application LUMA design process. The "Looking" and "Understanding" exercises form the most beneficial parts of LUMA within the constraints of OOUX.

OOUX is engineered to create the fastest path to usable, testable prototypes in Figma or in code. Realistic prototypes yield the most accurate user feedback and provide a more tangible sense of the application, making it easier to visualize complex requirements coalescing.

UX Initial Phase:

Paving the Way to an Effective Application

At the start, our main goal is to set a clear direction and lay the foundation for the app. It's important to build momentum and move towards a working prototype quickly, instead of getting stuck in research or analysis (analysis paralysis). As the prototype develops, we'll identify where we might need to do more LUMA exercises. These exercises are often more useful if they're done throughout the prototype development, rather than just at the start. At the beginning of the development process, we put a lot of effort into the UI to speed up the creation of the prototype.



LOOKING - FLY ON THE WALL

We start by forming a workgroup of experts who know the subject matter inside and out. In our case, we need two truck drivers and a mechanic who can help us understand what happens when a truck breaks down. We also need relevant documents, like truck manuals and paperwork, to help our designers grasp the process in detail. From this, we can start to draft some initial requirements, create user flows, and come up with questions that we need to answer. This will lead to the creation of two important documents: one for the questions we

need to ask the experts, and one for the requirements. We came up with the following questions for the experts after our first interview.

Questions

Questions	Answer	Source
Do all Semi trucks have same emergency procedure?		Trucker - Charles R., James J
Do emergency procedures change based on the size of the truck?		Trucker Mechanic - Charles R., James J
Do all corp repair shops cover all sizes of trucks		DOC - Freightliner Cascadia 125 User Manu
What emergency procedures are required based on the truck load? Chemical, food, refrigeration, livestock		Trucker - Charles R., James J
		Trucker - Charles R., James J

Requirements Documentation:

Corralling feedback from LUMA and Subject Matter Experts (SME) into a comprehensive plan can be challenging. I prefer to channel all feedback into a requirements document. This document also functions as a reference for auditing all prototypes and can be useful when developing a testing plan.

Requirement	LUMA UX Source	Subject Matter Source
Login	Basic Function	
Logout	Basic Function	
Show User	Basic Function	Basic Setting
Alert to Emergency	Interview	Trucker - Charles R., James J
Alert to Disabled	Interview	Trucker Mechanic - Charles R., James J
Set to Emergency	Interview	DOC - Freightliner Cascadia 125 User Manual
Set to Disabled	Interview	Trucker - Charles R., James J
Diagnose the Truck Problem	Interview	Trucker - Charles R., James J
See the specs on the truck	Interview	Trucker - Charles R., James J
See the specs on the load	Interview	Trucker - Charles R., James J



UNDERSTANDING - INTERVIEWING

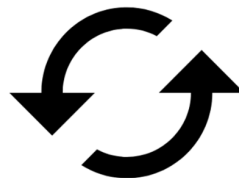


UNDERSTANDING -
IMPORTANCE/DIFFICULTY MATRIX

Answering questions quickly, adding in possible needs (that can be tweaked later), and deciding on what's most important with the experts helps the design process go smoothly. The experts should look at any changes to the design documents and give their thoughts. These ideas are then talked about in the next meetings. This back-and-forth chat keeps happening once it starts.



INTERVIEWING



UX EXERCISE
REQUIREMENTS



Material UI and similar well-established frameworks provide reliable and well-tested design pattern options. These can be modified to meet most needs. That's why the 'Lego' strategy of OOUX works well for these types of apps.

UX Secondary Phase

Refining UX artifacts with LUMA

As the UX designer moves forward with the interviews, requirements will be sorted based on user roles. There can be many personas, so it's easier to start with the main one (in this case, the trucker), then make a list of other personas to think about later. The most helpful personas align to user roles, each giving a separate view on the app. This is key since many personas will have similar needs. It's easier to address these needs in the main persona first. This way, designers don't end up creating different workflows and solutions for each persona.



UNDERSTANDING - PERSONA MAPPING



Persona 1 : Trucker

During interviews, it becomes easy to spot different personas when certain needs come up that aren't a fit for the main trucker persona. For instance, if a task involves adding trucks to the system, it's more likely to be for an administrator or a dispatcher who works for the trucking company. This task doesn't fit the trucker persona since you wouldn't want individual truckers adding more trucks to the system. These needs suggest there should be a new user role in the app.



Persona 2 : Admin Trucker Dispatch Persona

During another interview, a trucker expressed a desire to send pictures to the mechanic through chat. To facilitate this, the mechanic would need to have a designated contact point within the application and be responsible for updating this information. This implies that the mechanic shop owner would also need a unique 'view' into the application and a persona.



Persona 3 : Mechanic Shop Owner

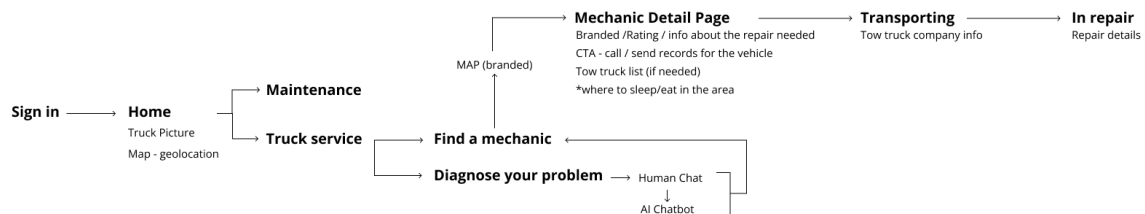
We are now at a point where we can integrate personas into our requirements document:

Requirement	LUMA / or Source	Persona	Subject Matter Source
Login	Basic Function	All	
Logout	Basic Function	All	
Show User	Basic Function	All	Basic Setting
Alert to Emergency	Interview	Trucker	Trucker - Charles R., James J
Alert to Disabled	Interview	Trucker	Trucker Mechanic - Charles R., James J
Set to Emergency	Interview	Trucker	Trucker - Charles R., James J
Set to Disabled	Interview	Trucker	Trucker - Charles R., James J
Diagnose the Truck Problem	Interview	Trucker	Trucker - Charles R., James J
See the specs on the truck	Interview	Trucker	Trucker - Charles R., James J
See the specs on the load	Fly on the Wall	Trucker	Trucker - Charles R., James J
Chat AI Diagnose Mechanical Problem	Interview	Trucker	Trucker Mechanic - Charles R., James J
Update Mobile Contact	Interview	Mechanic Shop	Trucker Mechanic - Charles R., James J
Add a new truck to the system	Interview	Dispatch / Ademin	Trucker Mechanic - Charles R., James J
Normal Person Diagnose Mechanical Problem	Fly on the Wall	Trucker	Trucker Mechanic - Charles R., James J

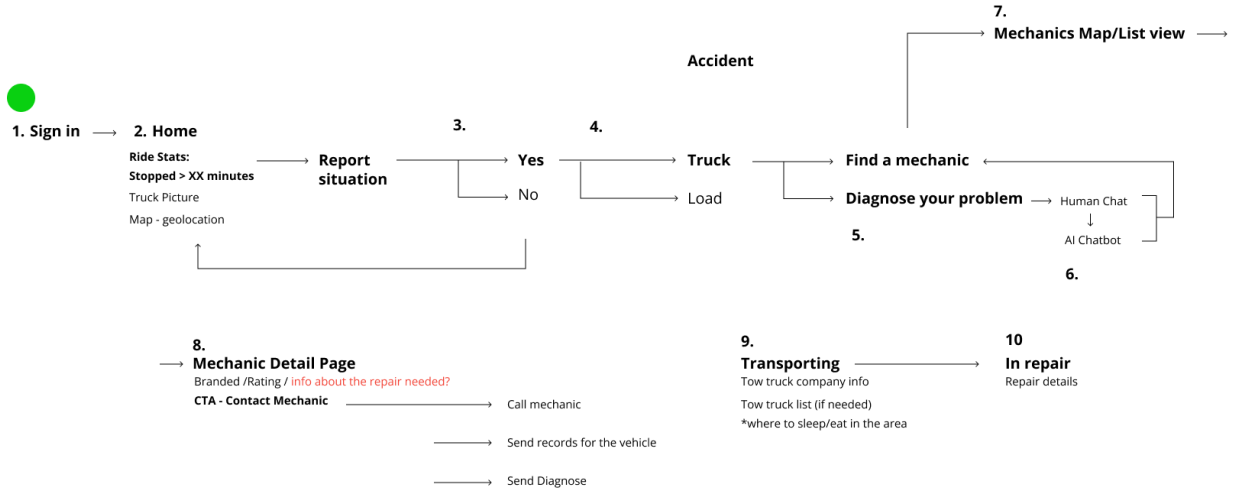


UNDERSTANDING - EXPERIENCE DIAGRAMMING

It is important to diagram the experience early in the application. The experience diagram is easy for our SME to understand in interviews and review. It is also a good roadmap for the initial prototype screens.



Over time, the user experience should expand and become more detailed as interviews reveal more required features.

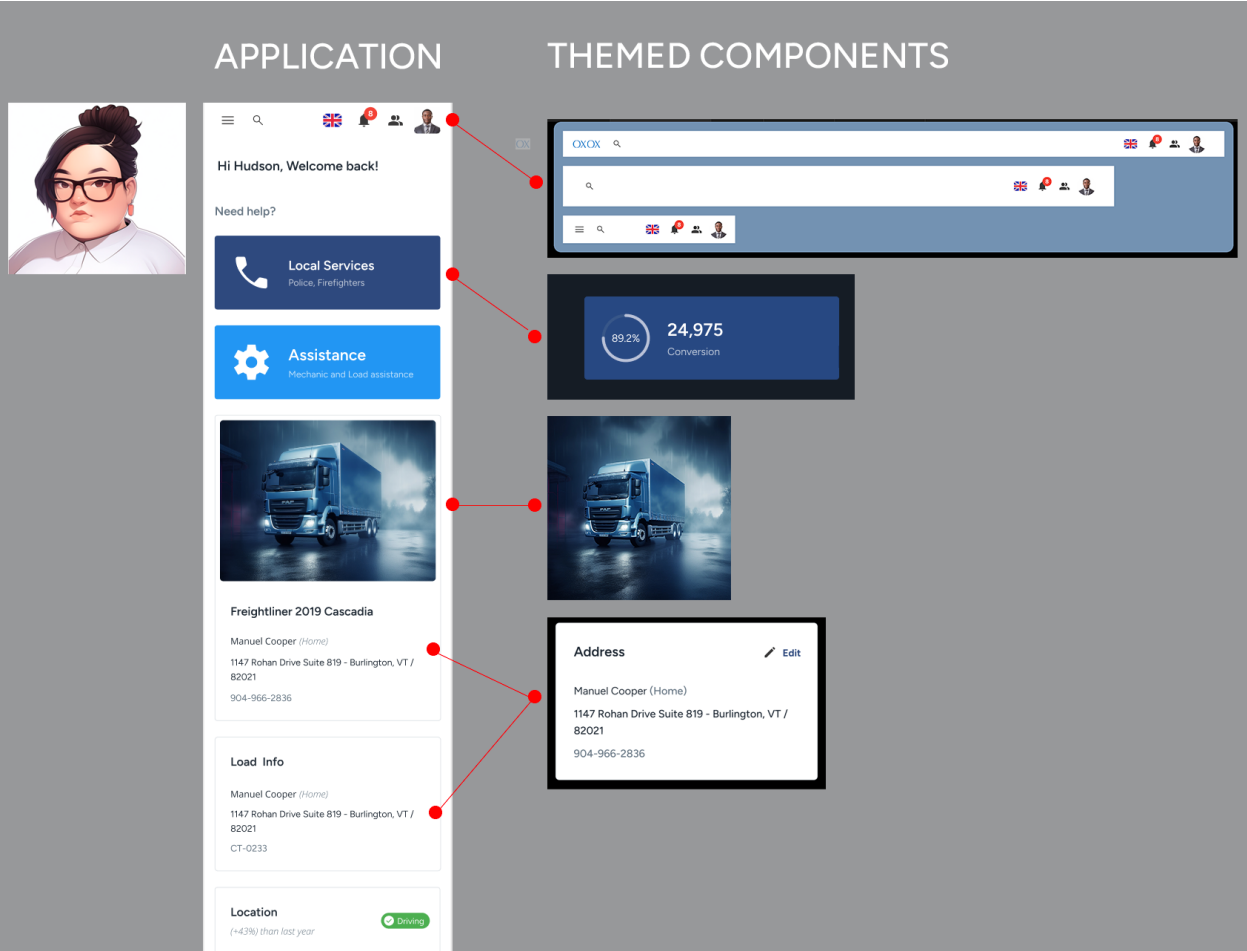


Create a bad prototype

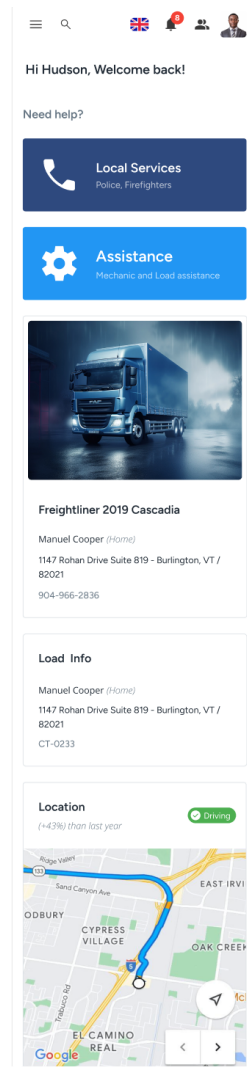
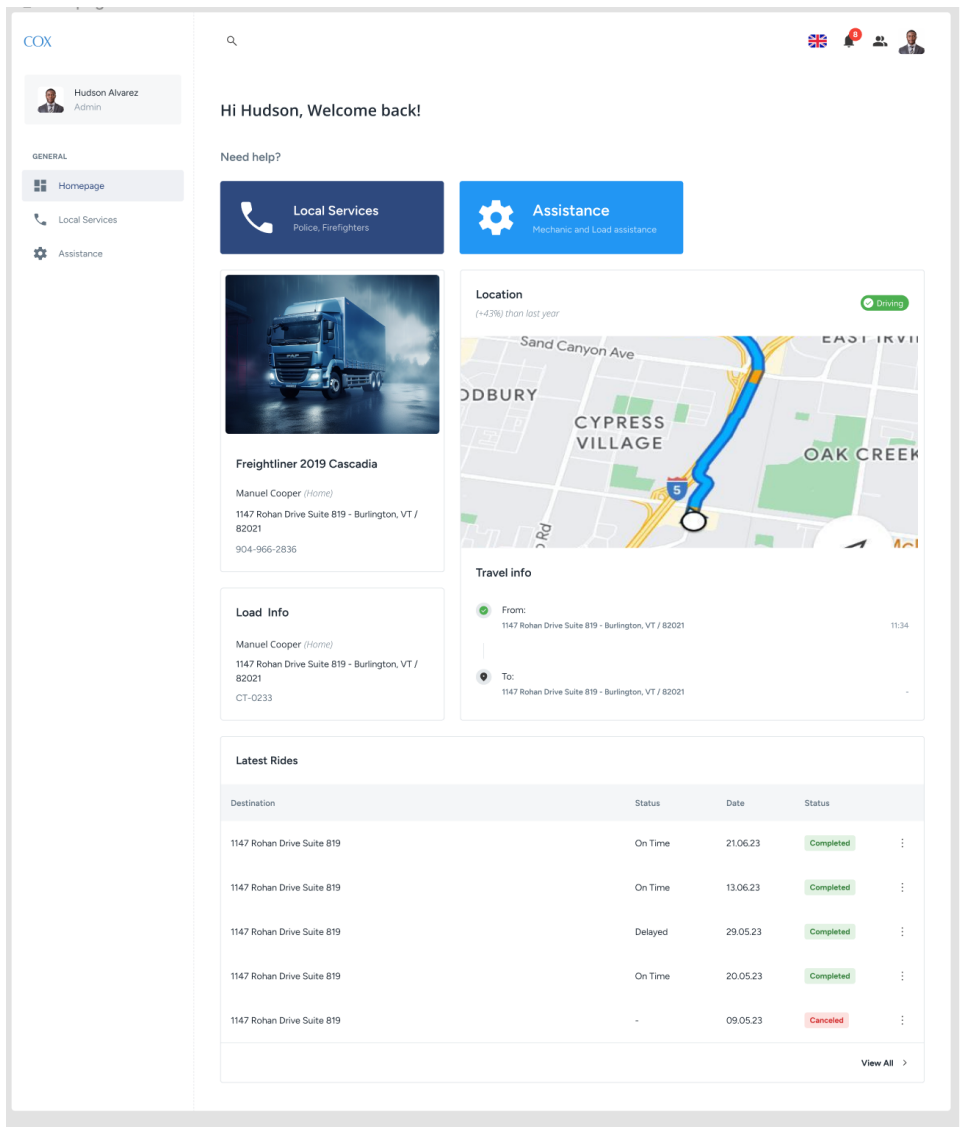
Set expectations that the initial prototypes may not look perfect or function flawlessly. The goal at this stage is to generate a tangible artifact that we can begin to assess, critique, and improve upon. The concept of 'failing fast' is highly valued in the product development process, as it encourages quick iterations and adjustments based on valuable feedback.

The beauty of OOUX (Object-Oriented UX) methodology and utilizing a unified design system (in this case, Material UI with Figma) is that it allows designers to quickly assemble high-fidelity prototypes with real-world constraints and complexity. These prototypes can be easily adjusted to reflect different screen sizes and device orientations, showcasing how components will behave across different user scenarios.

In our scenario, the designer would go through the library of themed components available in Figma, and assemble a mobile and desktop-friendly prototype within a short timeframe. For example, a menu component can be shown in different configurations from mobile to desktop, demonstrating the responsive nature of the design. This rapid prototyping ability reduces the time between idea generation and testing, thus accelerating the overall product development process.



Initial mobile and desktop Figma prototypes



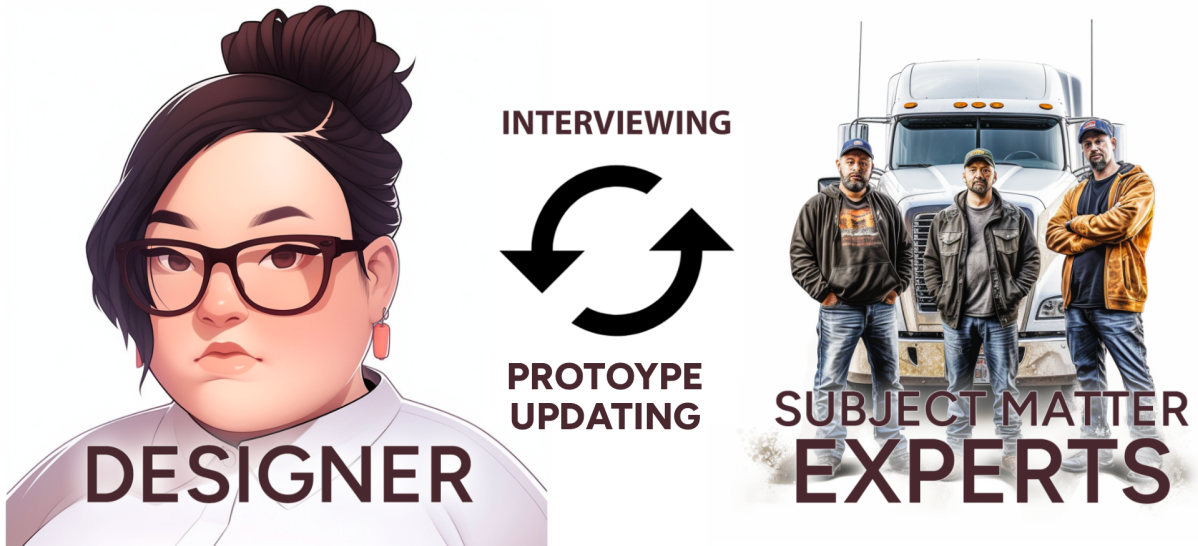
UX Third Phase

Turn a bad prototype into a good prototype.

Turning a preliminary prototype into a refined one is a crucial phase of the design process. This involves continuous iteration, testing, and obtaining feedback from SME, as well as end users. This ongoing refinement and validation process helps ensure the prototype evolves to meet users' needs effectively and intuitively.

We can help non-designers, like SME or stakeholders, to grasp the design decisions and participate effectively in the design process with the speed of OOUX prototype development. Their unique insights provide invaluable input to the prototype refinement process.

Designing an effective interface involves consideration of a multitude of factors and constraints, many of which aren't immediately apparent to the end user. These factors can include touch target sizes, text legibility, response times, animation quality, and more. However, when these factors are carefully considered and appropriately implemented, they contribute to an interface that "feels right" and is intuitive to use - even for users who aren't aware of all the design considerations that have gone into creating the interface. This reinforces the fact that great design often goes unnoticed by users - not because it's unappreciated, but because it allows them to achieve their goals effortlessly and intuitively



Resolving disagreements about UX can be challenging, but there are strategies to manage such situations effectively:

1. Facilitate open discussions: Encourage the stakeholders to openly discuss their conflicting requirements and possible solutions. An environment that fosters open communication can help people express their perspectives, leading to more comprehensive solutions.
2. Use auditing test questions: When conflicting requirements emerge, incorporating auditing test questions in the requirements document can help. These questions should aim to resolve the disagreement at hand. If stakeholders can agree on what question is being answered, it's a step towards finding common ground. Then ask each stakeholder to use the prototype to answer the audit question.

Requirement	LUMA / or Source	Person a	Subject Matter Source	UX User Task Audit Questions
Login	Basic Function	All		How do you login?
Logout	Basic Function	All		How do you login?
Show User	Basic Function	All	Basic Setting	
Alert to Emergency				
	Interview	Trucker	Trucker - Charles R., James J	What is the current state of your truck according to the application?
Alert to Disabled				
	Interview	Trucker	Trucker Mechanic - Charles R., James J	What is the current state of your truck according to the application?
Set to Emergency	Interview	Trucker	Trucker - Charles R., James J	Set the application to an emergency state.
Set to Disabled	Interview	Trucker	Trucker - Charles R., James J	Set the application to a disabled state.
Diagnose the Truck Problem	Interview	Trucker	Trucker - Charles R., James J	Diagnose the problem with the engine using the application

3. Leverage fast prototyping: The OOUX framework is designed for fast prototypes. If there's disagreement, create prototypes for both solutions and let stakeholders interact with them. This hands-on approach can help people understand the pros and cons of each solution better and can often clarify which solution is best.
4. Engage in collaborative decision-making: Having everyone in the workgroup present when discussing solutions can be extremely valuable. This collective decision-making process can ensure that all perspectives are considered and that any decisions made have broad support.

When new requirements or deficiencies are identified in the design (such as the trucker's need to define the state of the truck, trailer, and load in our application), the process of addressing them should involve similar strategies. Open discussion, prototype iterations, and collaborative

decision-making will help to incorporate these new requirements into the design in a way that satisfies all stakeholders.

Once the initial prototype has been developed, other issues become apparent. For example, the trucker workgroup has some disagreements regarding the most suitable metrics to be displayed on the front page. Given the application's mobile-first design approach, it is necessary to structure the application from simple to complex to accommodate the user's experience on smaller screen sizes. This means the initial dashboard should be limited in that amount of data it can display for mobile users. This is a good place to explore the prioritization of features with another importance/difficulty matrix. Any LUMA exercise might be required in this part of the process. It all depends on the problem that needs to be solved.



UNDERSTANDING - IMPORTANCE/DIFFICULTY MATRIX

Requirement	Imp	LUMA / or Source	Persona	Subject Matter Source	UX User Task Audit Questi
Login	1	Basic Function	All		How do you login?
Logout	1	Basic Function	All		How do you login?
Show User	1	Basic Function	All	Basic Setting	
See the specs on the load	2	Fly on the Wall	Trucker	Trucker - Charles R., James J	
Chat AI Diagnose Mechanical Problem	2	Interview	Trucker	Trucker Mechanic - Charles R., James J	
Update Mobile Contact	2	Interview	Mechanic Shop	Trucker Mechanic - Charles R., James J	

The initial dashboard view should present enough information to provide the trucker with an overview of their vehicle's status and their current journey. Each metric should be designed around a "click or don't click" principle. For instance, if the dashboard indicates that the truck is stationary and is behind schedule, that is sufficient information for the trucker to realize that they need to delve deeper into this metric for more information.

Thus, the design approach emphasizes clarity and conciseness, presenting only the most relevant information at first glance while offering deeper layers of information for those who need it. This approach helps to maintain an uncluttered, user-friendly interface while still providing comprehensive data for users who require it.

Our initial prototype:

COX

Hi Hudson, Welcome back!

Need help?

Local Services
Police, Firefighters

Assistance
Mechanic and Load assistance

Freightliner 2019 Cascadia
Manuel Cooper (Home)
1147 Rohan Drive Suite 819 - Burlington, VT / 82021
904-966-2836

Location
(+43%) than last year

Travel info

From: 1147 Rohan Drive Suite 819 - Burlington, VT / 82021 11:34

To: 1147 Rohan Drive Suite 819 - Burlington, VT / 82021

Latest Rides

Destination	Status	Date	Status
1147 Rohan Drive Suite 819	On Time	21.06.23	Completed
1147 Rohan Drive Suite 819	On Time	13.06.23	Completed
1147 Rohan Drive Suite 819	Delayed	29.05.23	Completed
1147 Rohan Drive Suite 819	On Time	20.05.23	Completed
1147 Rohan Drive Suite 819	-	09.05.23	Cancelled

View All >

Hi Hudson, Welcome back!

Need help?

Local Services
Police, Firefighters

Assistance
Mechanic and Load assistance

Freightliner 2019 Cascadia
Manuel Cooper (Home)
1147 Rohan Drive Suite 819 - Burlington, VT / 82021
904-966-2836

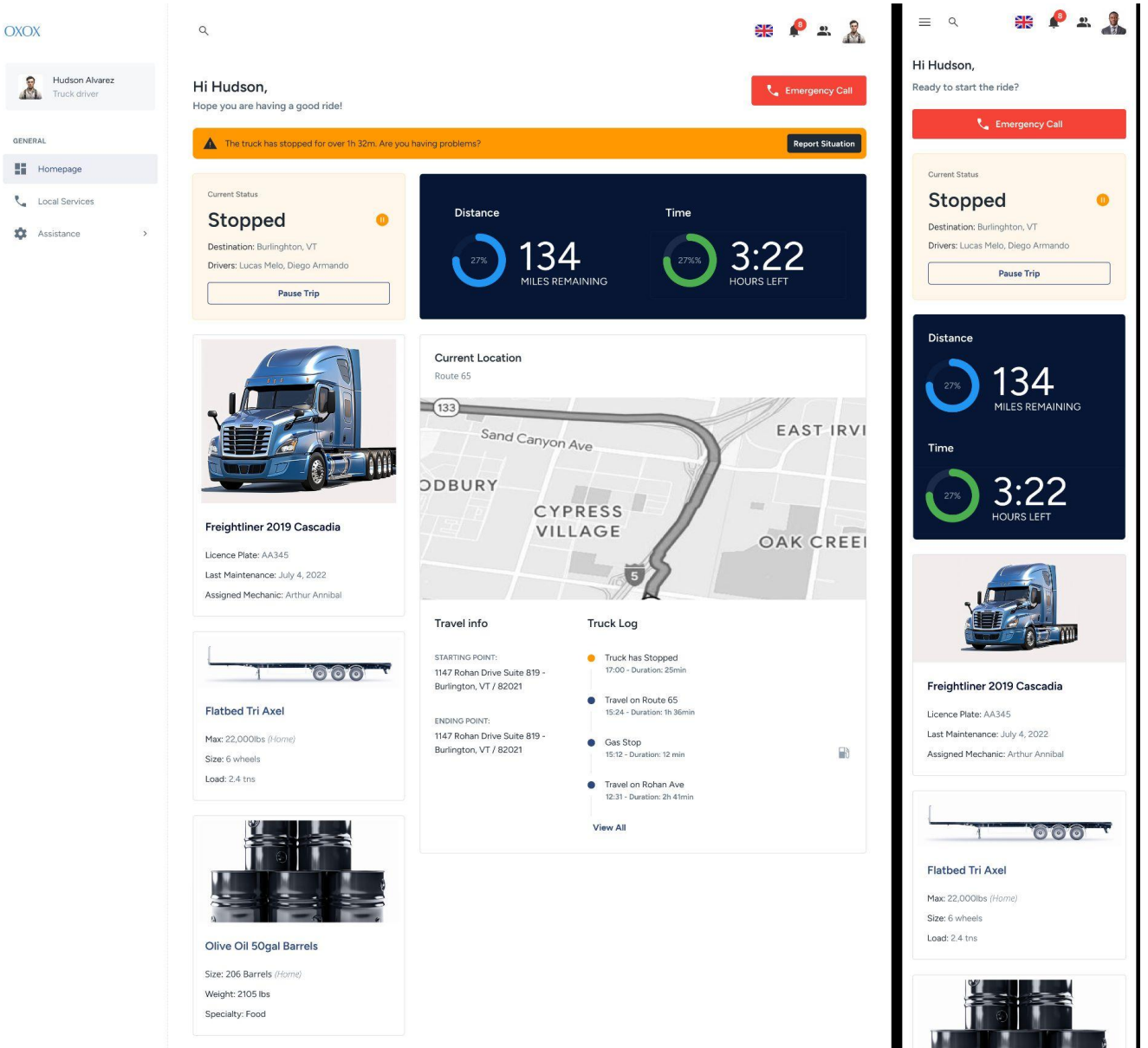
Location
(+43%) than last year

Latest Rides

Destination	Status	Date	Status
1147 Rohan Drive Suite 819	On Time	21.06.23	Completed
1147 Rohan Drive Suite 819	On Time	13.06.23	Completed
1147 Rohan Drive Suite 819	Delayed	29.05.23	Completed
1147 Rohan Drive Suite 819	On Time	20.05.23	Completed
1147 Rohan Drive Suite 819	-	09.05.23	Cancelled

View All >

Our final prototype:



Through iterative cycles of interviews with subject matter experts, the team arrived at the final prototype design. Several important changes are noticeable in this new version, showing an evident response to user feedback.

- The operator's responsibilities, namely the truck, trailer, and load, are prominently displayed on the front page of the dashboard. This positioning allows operators to access essential information quickly and conveniently report any issues.

- Moreover, based on user feedback, the estimated time and distance metrics have been given prime real estate on the front page. Operators and stakeholders indicated these metrics as the most critical for their workflow and decision-making.
- One of the key changes in this iteration is the status report of the truck. The application now promptly reports if the truck is stopped and needs repairs to the dispatch, enabling swift communication and efficient decision-making.
- Furthermore, an inclusion of a truck log that captures everything reported about the truck creates a centralized point for all historical data. This feature can serve as an invaluable reference for mechanics, drivers, and the management.

This final prototype exemplifies a thoughtful approach to integrating user feedback. By striking a balance between delivering essential information and maintaining a clean, navigable interface, the design team ensures that the application remains user-centric. However, due to the ever-evolving user needs and business contexts, the design team is dedicated to consistently gather feedback and iterate on the design to stay abreast of the dynamic nature of the trucking industry.

OOUX offers the invaluable gift of time, enabling the design and development team to concentrate on features that will set the product apart from the competition.

When contemplating this in detail, it's notable that this method mirrors the way many of the most innovative products have been designed. Take Tesla, for instance. The first Tesla model brought to the market two significant improvements: an efficient electric engine and an innovative operating system. Yet, elements like seat warmers, tire rubber, and cup holders were no different from those in a Honda Civic. The OOUX methodology covers the basic features and design patterns, freeing up time and resources for elements that will differentiate the product.

The final product meets all the requirements:

Requirement: Responsive from Desktop to Mobile

OXOX

Hi Hudson,
Hope you are having a good ride!

Emergency Call

The truck has stopped for over 1h 32m. Are you having problems?
Report Situation

Current Status
Stopped
Destination Burlington, VT
Drivers Lucas Melo, Diego Armando

Distance 100% **873** Miles Remaining
Time 100% **10:04** Hours Left

Current Location
Route 65

Freightliner 2019 Cascadia
License plate: AA345
Last Maintenance: July 4, 2022
Assigned Mechanic: Arthur Annibal

Travel Info
STARTING POINT
1147 Rohan Drive Suite 819 - Burlington, VT / 82021

Truck log
● Travel not started yet

Emergency Call

Hi Hudson,
Ready to start the ride?

Stopped
Destination Burlington, VT
Drivers Lucas Melo, Diego Armando
Pause Trip

Distance 27% **134** MILES REMAINING
Time 27% **3:22** HOURS LEFT

Freightliner 2019 Cascadia
License Plate AA345
Last Maintenance: July 4, 2022
Assigned Mechanic: Arthur Annibal

Flatbed Tri Axel
Max: 22,000lbs (gross)
Size: 6 wheels
Load: 2.4 tons

Olive Oil 50gal Barrels
Size: 206 Barrels (gross)
Weight: 2105 lbs
Specialty: Food

Requirement: Localization (in this case can be run in Spanish)

OXOX

Hudson Alvarez
Camionero

Página De Inicio

Asistencia

Hola Hudson,
Espero que estés disfrutando del viaje!

El camión se ha detenido durante más de 1h 32m. ¿Tienes problemas?

Reportar Situación

Estado Actual

Interrumpido

Destino Burlington, VT
Conductores Lucas Melo, Diego Armando

Distancia: 873 Millas Restantes

Tiempo: 10:04 Horas Restantes

Current Location
Route 65

Map | Satellite

Foam Brewers - Burlington

OLD NORTH END

CENTENNIAL

University of Vermont

Accessibility: That application passes usability requirements.