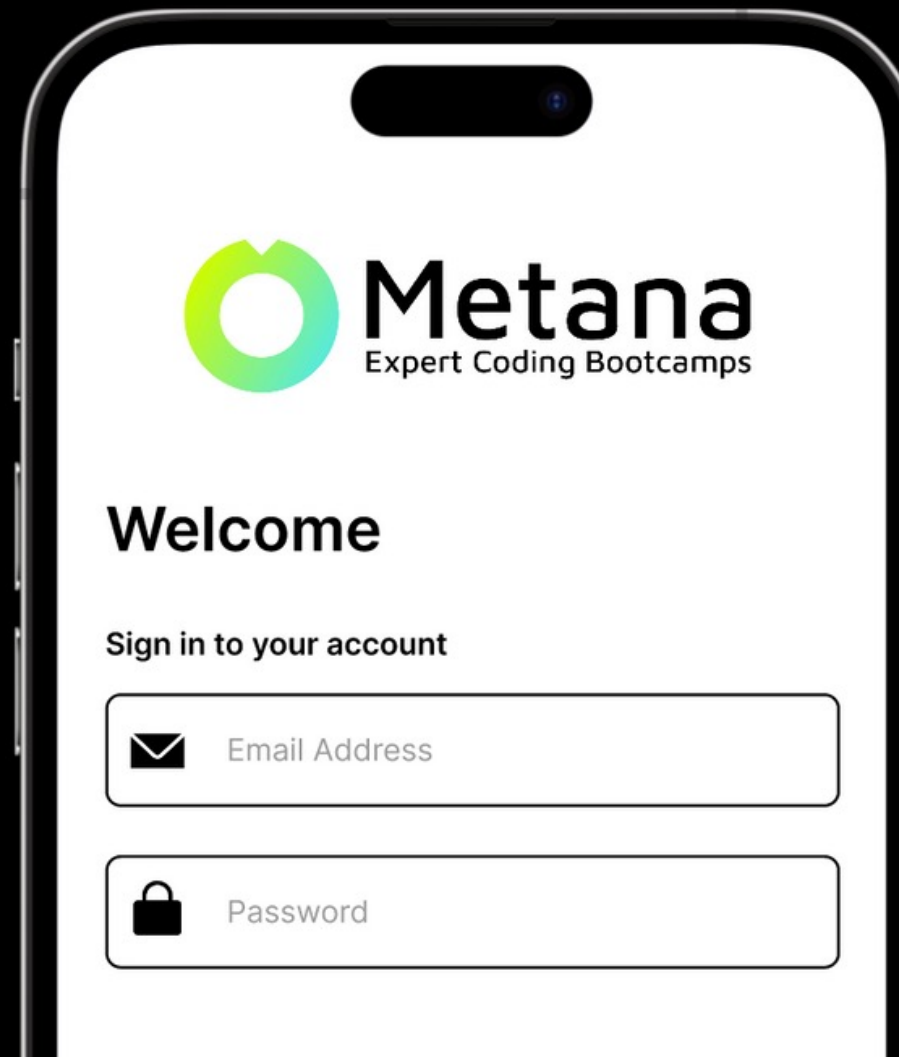


Unison App

Custom Marketing App: A Centralized Platform for All Your Marketing Needs



Imeth Vinnath

CB010675

SE Undgraduate

2023/08/10

Sri Lanka

Unsion App is an idea originated from working in an E-learning Start-up

After working for several months and associating with different Start Ups,
I noticed that an all-in-one organizational app, centralizing data customized to cater the business operations was ideal
to bridge inefficiencies that arise due to switching between platforms.

The idea is simple... Integrations + Centralization

About the App

Problem

During my experience working in a start up, I noticed that most start-ups are using a wide array of online tools as each of them provide a more granular control over different aspects of the business process.

What does it have to do with marketing?

A key part of marketing is to know what works and what doesn't. Dashboards and reports generated by analyzing record data in multiple platforms help marketing teams in visualizing and tracking the success of a marketing strategy

So what you really want is

An all-in-one App, designed for your business, Seamless integrations with all your marketing and sales needs. And your data is yours!



Features

Note: The features developed in this app, will be to cater to the business requirements identified in a specific company

- 01** Sign Up : Organization can facilitate the onboarding process of an employee by Signing up for the organizational email [reference](#)
- 02** Once logged into the app with organizational email, user can set up his/her accounts into app. (Slack, LinkedIn, Gmail, Hubspot, TypeForm)
- 03** The user can also then view all the events, tasks and manage them via the events view.
- 04** The app will have several dashboard including lead analytics which help the marketing team analyze performance of their strategies and overall marketing
- 05** Leads can be added as contacts, and you can also import contacts from your phone.

Design

- 01** Clarity: The app is easy to understand and use.
- 02** Consistency: The app has a consistent look and feel throughout.
- 03** Efficiency: The app is efficient and easy to navigate.
- 04** Simplicity: App is designed to be as simple as possible, without extraneous features or distractions that detract from the user's experience.
- 05** Emotion: The app is designed to be engaging and enjoyable to use.

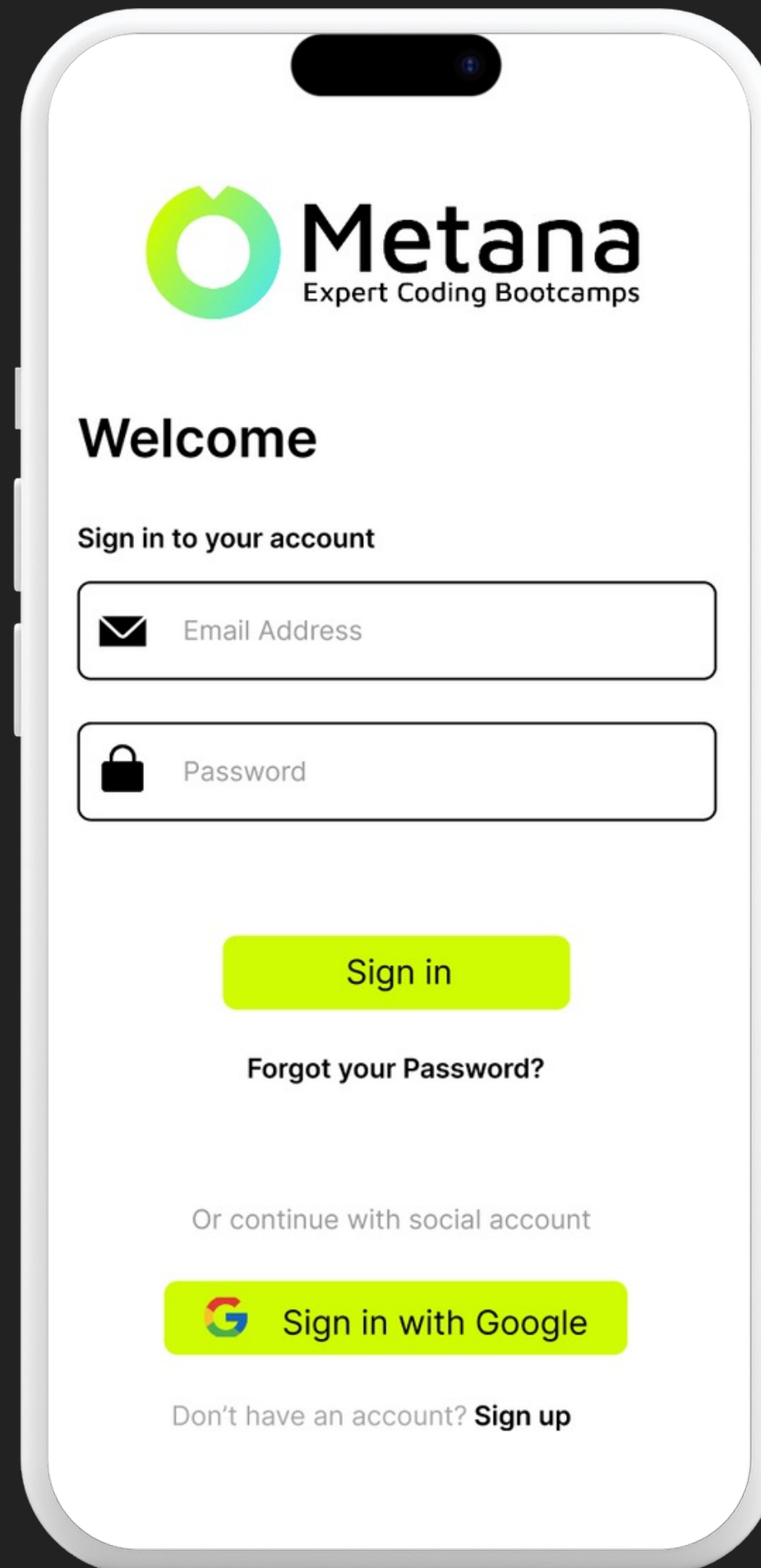
User Interactions

[See prototype](#)

If user has an organizational account, he can simply log in using the password or a social login (google)

If user, doesn't have an account, he can sign up for an account, after authorization from the admin, user will get the credentials via e-mail, where the user can enter the application using his newly created account

User can view messaging channels by clicking the message icon on the navbar and set up their slack, linkedin and hubspot accounts using the organizational email address



User Interactions

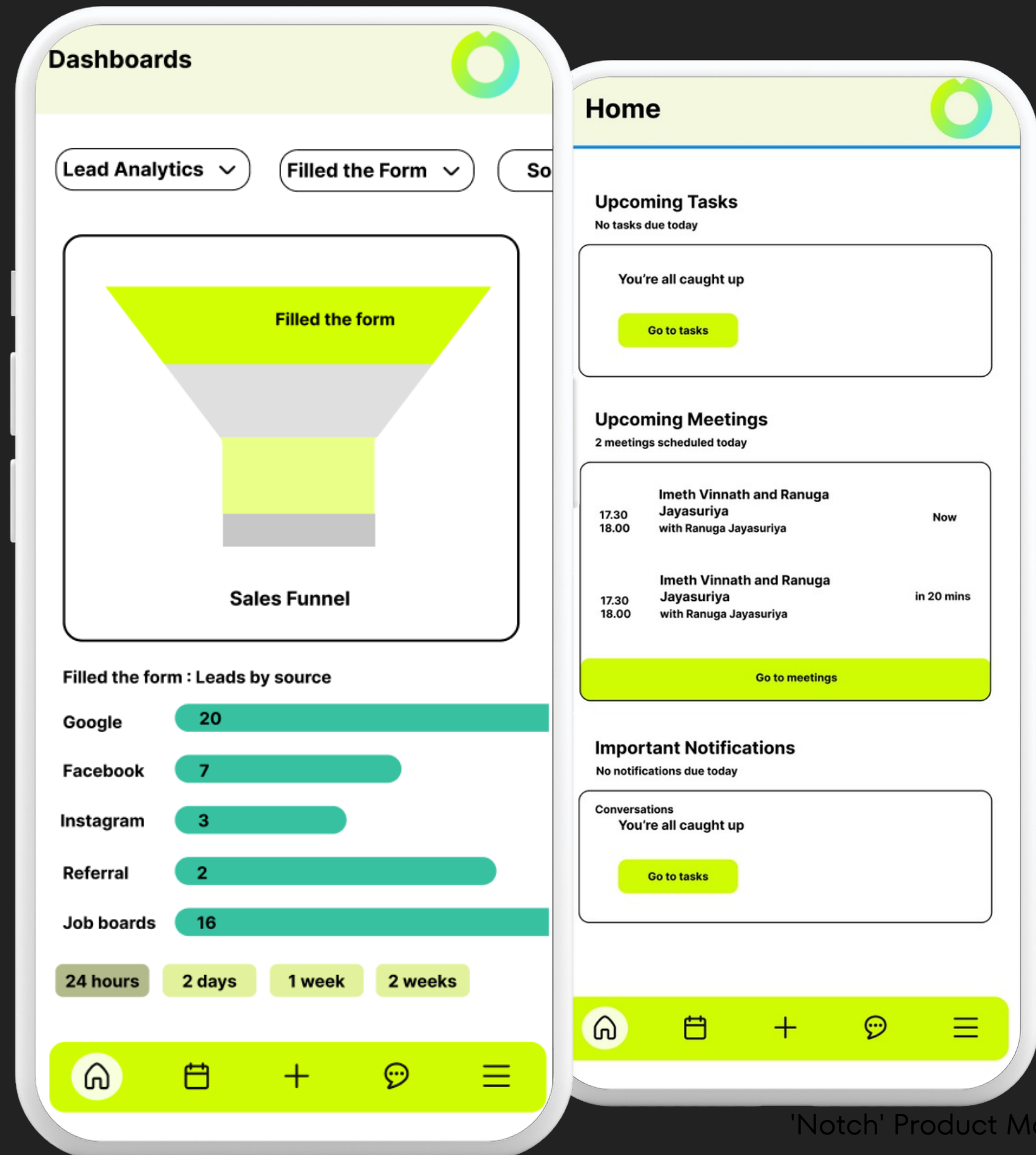
[See prototype](#)

Depending on the role of the user, he/she is able to view all accessible reports, analytics, tasks from the side menu

User is also able to import contacts and create deals and create tasks by clicking on the + icon on the navigation bar

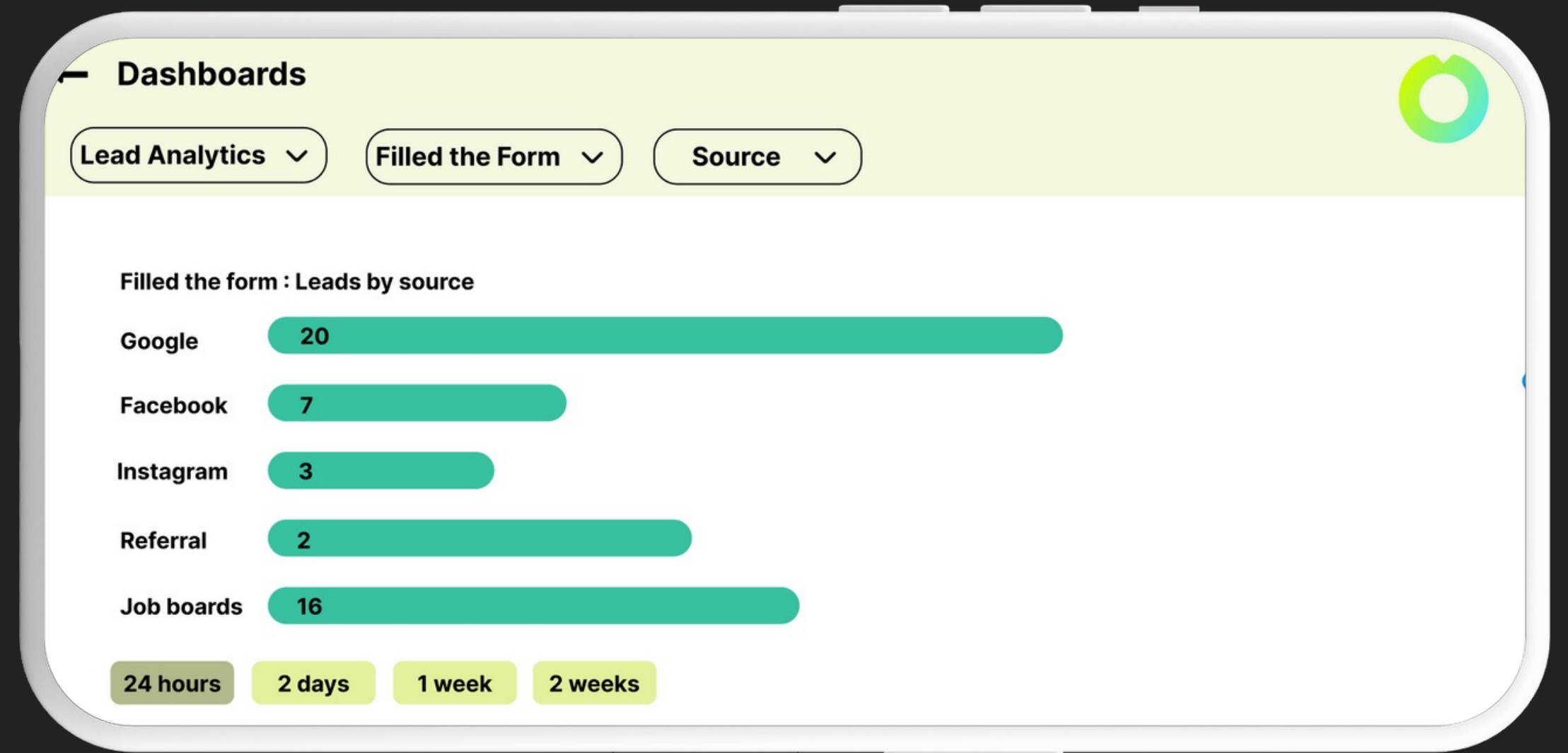
User can view messaging channels by clicking the message icon on the navbar and set up their slack, linkedin and hubspot accounts using the organizational email address

User's home view is mainly populated with tasks assigned and upcoming meetings that is synchronized from the users organizational email address



Mockup

Landscape design is adopted whenever the portrait restricts the view, such as displaying charts etc.



Transitions

- 01** The app has a loading animation that will be displayed in order to facilitate any latency issues, the design has been made to engage the user and not to worry about the speed.
- 02** When transitioning between screens it's mostly an instant or a dissolve animation.
- 03** When clicking buttons the button transition is always highlighted if screen is not changed, for the user to know which button he clicked on



Different approaches taken in modern mobile app development

Approach	Cost	Security	Tech Stack	Performance	Architecture	Cross-Platform Support	Development Time	Languages	Best Suited For
Native Apps	High (separate codebases for each platform)	High (platform-specific security measures)	Android Studio, Xcode	High (optimized for specific platforms)	Platform-specific	No (Separate apps for iOS and Android)	Long	Swift/Objective-C (iOS), Java/Kotlin (Android)	Apps requiring optimal performance, rich UI/UX, full access to device features
Hybrid Apps	Medium (one codebase for multiple platforms)	Medium (Depends on the WebView's security)	Cordova, Ionic, PhoneGap	Medium (depends on WebView performance)	WebView-based	Yes	Medium	HTML, CSS, JavaScript	Rapid prototyping, simple apps with less native functionality
Web Apps	Low (No app store deployment)	Varies (Depends on web security implementations)	Any web development framework	Low (compared to native apps)	Client-Server	Yes (Accessible from any web browser)	Short	HTML, CSS, JavaScript	Websites with app-like interaction, businesses needing short development times
Progressive Web Apps (PWA)	Medium (No app store deployment, but requires service worker development)	Varies (Depends on web security implementations)	Any web development framework with Service Workers	Medium-High (Can work offline and perform nearly as well as native apps)	Client-Server with Service Workers	Yes	Medium	HTML, CSS, JavaScript	Businesses wanting to combine website and app, without app store submission
Compile to Native (Cross-compiled)	High (Advanced frameworks)	High (Compiled to native code)	React Native, Flutter, Xamarin	High (Comparably close to native apps)	Varies depending on framework	Yes	Medium	JavaScript (React Native), Dart (Flutter), C# (Xamarin)	Apps that require near-native performance and cross-platform support with single codebase

Design and publishing process

Phase	Android	iOS
Design	Android offers Material Design guidelines. Designers can use these guidelines to build UI for Android apps. Tools like Figma , Adobe XD, Sketch can be used for design.	Apple provides Human Interface Guidelines for designing iOS apps. Designers often use tools like Sketch, Adobe XD, Figma for designing the UI.
Development	Android apps are primarily developed using Java or Kotlin in Android Studio.	iOS apps are primarily developed using Swift or Objective-C in Xcode.
Testing	Apps can be tested using the Android emulator or real Android devices via Android Studio.	Apps can be tested using the iOS simulator or real iOS devices via Xcode.
Preparation for Submission	Create a Google Developer account (\\$25 one-time fee), provide app details like name, description, and screenshots.	Create an Apple Developer account (\\$99/year), provide app details like name, description, and screenshots.
Building and Signing the app file	Build and sign your APK file in Android Studio.	Build and sign your IPA file in Xcode.
Uploading the app file	Upload your APK file to Google Play Console.	Upload your IPA file to App Store Connect.
Reviewing App Listing	Review and finalize your app's listing in Google Play Console.	Review and finalize your app's listing in App Store Connect.
Submission	Submit your app for review in Google Play Console.	Submit your app for review in App Store Connect.
Approval and Publishing	Once approved, the app will be available on Google Play Store.	Once approved, the app will be available on App Store.

References

Design principles

Direduryan, C. (2018). The Basics of UX/UI Design Principles. (Accessed: July 29, 2023) Available at: <https://www.linkedin.com/pulse/basics-ux-ui-design-principles-chris-direduryan/> .

Different approaches in mobile app development

Majchrzak, T.A., Biørn-Hansen, A. and Grønli, T.M., 2016, July. Comprehensive analysis of innovative cross-platform app development frameworks. (Accessed: August 4 2023) URL : https://www.researchgate.net/publication/317423949_Comprehensive_Analysis_of_Innovative_Cross-Platform_App_Development_Frameworks

Mobiloud. (n.d.) Native, Web, or Hybrid Apps? What's the Difference?. (Accessed: August 5 2023). Available at: <https://www.mobiloud.com/blog/native-web-or-hybrid-apps>

Documentation

Android Developers. (n.d.) App Basics. (Accessed: : August 9 2023,). Available at: <https://developer.android.com/guide>

Apple Developer Documentation. (n.d.) (Accessed: : August 9 2023). Apple Developer Documentation. Available at: <https://developer.apple.com/documentation/>

Flutter. (n.d.). Docs. (Accessed: *Date*). Available at: <https://flutter.dev/docs>

Apple Developer. (n.d.) Submit apps to the App Store. (Accessed: *Date*). Available at: <https://developer.apple.com/app-store/submissions/>

Android Developers. (n.d.) Publish your app. (Accessed: *Date*). Available at: <https://developer.android.com/studio/publish>

Google Play Console Help. (n.d.) Create an app. (Accessed: *Date*). Available at: <https://support.google.com/googleplay/android-developer/answer/113469?hl=en>

API documentation URLs

<https://api.slack.com/docs/conversations-api>

<https://discord.com/developers/docs/intro>

<https://developers.hubspot.com/docs/api/overview>

<https://developers.google.com/admin-sdk/directory/v1/guides/manage-users>