**Internship Project Report**

|  |  |  |  |
| --- | --- | --- | --- |
| **Sr. No.** | **Full Name** | **College Name** | **Mentor Name** |
| 1. | Akash C Sanjeev | Goa College of Engineering | Vishvanath Chili |
| 2. | Livio F. Mascarenhas | Padre Conceicao College of Engineering | Bharath Rajaram |
| 3. | Prateek Sharma | Goa College of Engineering | Bharath Rajaram |

**Contents**

**SR. NO.** **TITLE** **PAGE NO.**

1. Brief Overview 2

2. Introduction 3

3. Technologies Used 6

4. Use Cases 7

5. Details of Work Done 7

7. Challenges Faced 16

8. Future Scope 17

9. Conclusion 17

10. References 18

**BRIEF OVERVIEW**

The project aims to create a pair of Android applications that work coordinated to enhance the functionality of cluster displays. One application will serve as the primary cluster display, while the other will act as an extended display, augmenting the cluster interface. This innovative solution is designed to improve user experience and offer enhanced capabilities for Android-based cluster systems.

The primary cluster display application will function as the core interface, providing essential information and various widgets to the users. It will offer a user-friendly and intuitive interface for accessing data and interacting with various widgets present in the cluster system. Users can expect a seamless and efficient experience when using this application.

The extended display application will complement the primary cluster display by offering an extended screen thus expanding the cluster's capabilities.

Overall, this project aims to create a versatile Android-based solution for cluster displays, offering both a user-friendly primary interface and an extended display option with added functionality. The development of these applications will enhance the usability and utility of cluster systems, making them more adaptable to various user needs and preferences.

**INTRODUCTION**

In an era defined by ever-evolving technology and the increasing integration of smartphones into our daily lives, the concept of utilizing Android applications to enhance user experiences has become paramount. This project endeavors to explore the possibilities within this digital landscape by embarking on the development of a pair of Android applications with a unique and practical purpose.

As we all are aware that the current pace at which the Electric market for transportation is booming especially in India, there is a need to bring about new and practical innovations that can improve overall user experience and satisfy customer needs. This is where our project brings a similar innovation that has to offer something new in this ever-evolving Electric vehicle market.

The primary objective of this project is to create a symbiotic duo of Android applications, where one assumes the role of the primary cluster display, while the other functions as an extended display for the cluster interface. This innovative approach aims to push the boundaries of mobile application design and usability, opening new avenues for interaction and utility for Android device users.

if we go to see, especially for electric bikes, they have so much of data to be displayed and various widgets that provide the user with critical data. The only restriction here is that the user is not able to see multiple widgets at once as he is restricted to a single screen only.

This is where an extended screen would be of significant use, as it would allow the user to get some of those widgets onto it and that provide the user to see more data at once without the need of having to frequently interact with the main cluster screen to get the widgets that the user wants.

Further we discuss exactly what our project is offering and more about what our two main elements are, namely, the main cluster screen and the extended display, onto which our entire project revolves.

For this project we will be Building 2 Android Applications, for Main Screen (to run on Windows) and Extended Screen (to run on Mobile) each. Further there will be 8 widgets in total, in addition there will be a speedo-Odo widget that would be available only on the main screen.

The eight widgets would be of varied sizes i.e., small, medium, and large and based on which screen they are present, these would further differ in sizes.

When the extended device is not connected, the user will have the option to view all the 8 widgets in the three sizes on the main screen itself.

**Main Screen:**

* + The viewport size of the main screen is 1100 x 550 pixels.
  + The main cluster screen would run on windows.
  + The beginning screen would be the speedo-Odo which would contain data like speed, Odo, battery and range, bike modes, time, indications, headlight, park assist and network strength.
  + On swiping from the main screen, the user will be able to get the widgets with a minified version of the speedo-Odo.
  + On this screen, the layout is divided into 3 columns, the leftmost column would be for the speedo-Odo widget, and the rest two columns would contain all the other 8 widgets.
  + Out of these two columns, the first column would contain the first 4 widgets and the second column would contain the other 4 widgets. Though when the widget is in large mode, the main screen would be divided into two columns, one containing the speed-Odo and the other the large sized widget.
  + Initially these widgets would be in small and medium sizes, with the availability to have them in large size as well.
  + Within each of these two columns the user has the freedom to scroll the column to reveal the other widgets, the user can also drag the widgets to the other column (not the leftmost speedo-Odo widget), also on a single tap the widgets would toggle between small and medium sizes. When double tapped, the tapped widget would be displayed in large size wherein the other widgets would be displayed on the left side as small icons, which you can click to see them in large size and double tap again to get back to the original layout of 4 widgets. Now when the main screen app is connected to the extended screen, the user can long-press the widget to send it to the main screen.
  + Once the user has connected to the Extended Screen, the 4 widgets that are not currently being displayed on the main screen will be shifted to the extended screen and the main screen would be left with only 4 widgets.

**Extended Screen:**

* + The viewport size of the extended screen is 870 x 360 pixels.
  + Once the Main screen app is connected to the extended screen, the 4 widgets that are initially not able to be seen on the main screen will be shifted off to this extended screen. (speedo-Odo would not be present in the extended app).
  + On the extended screen the widgets will again be available in 3 sizes, small, medium, and large.
  + The extended screen's layout is divided into 3 columns containing 4 widgets. The first two columns will hold 2 widgets of medium size and the rightmost column would have the other two widgets of small size. When a particular widget is in full size, the layout would be divided into two columns, the left column holding the full-sized widget and the right side having the others.
  + Like the main screen app, the user can drag the widgets between the columns, double tap to toggle between the large size and the medium size and long-press to send it to the main screen and replace it with another widget from the main screen.
  + **Extended screen:**



Figure: Extended Screen

* + **Extended screen**, after double tapping a widget:

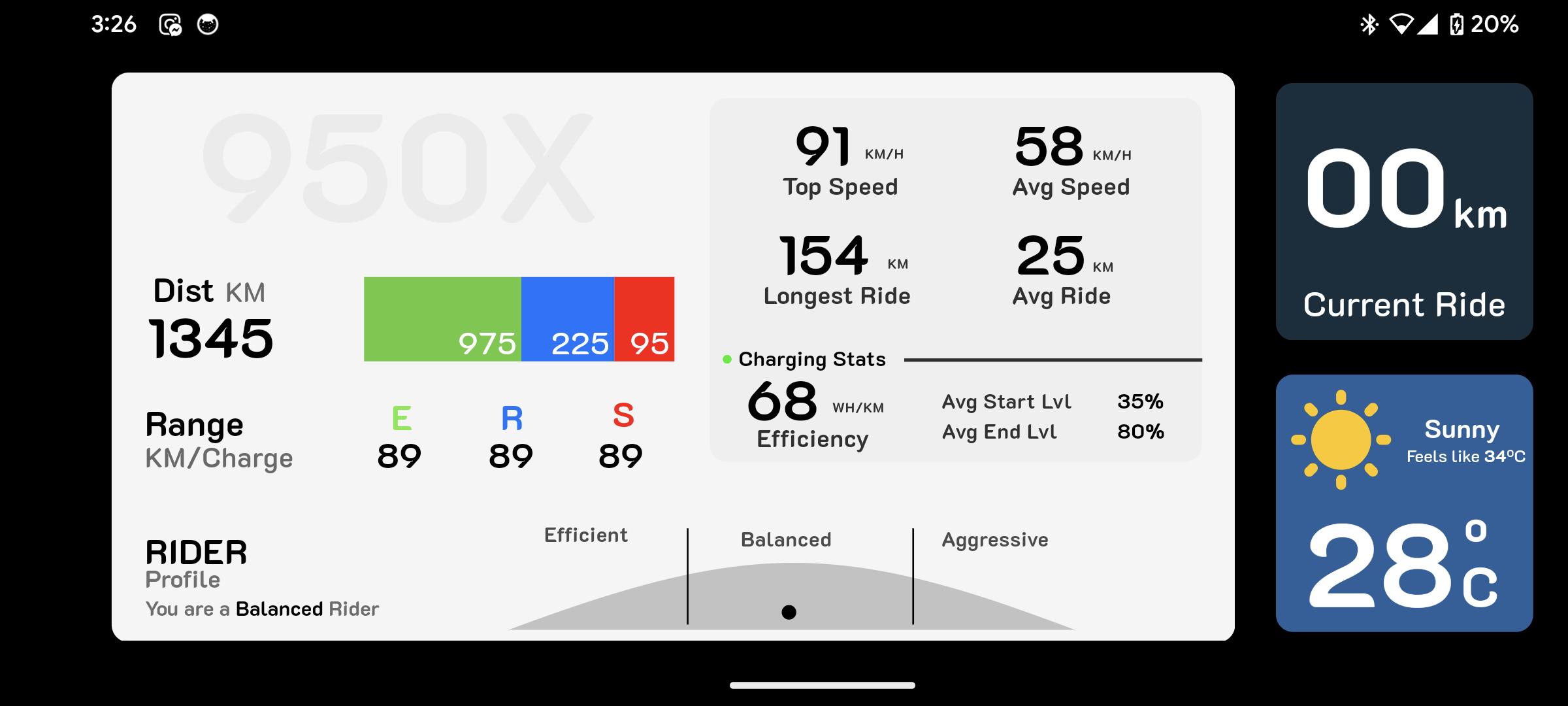


Figure: Extended Screen (with large sized widget)

As we delve deeper into this project, we will discuss the various user interface elements, the widgets, the communication between the primary and extended displays, animations, our journey of mobile app development, user experience enhancement, and creative problem-solving. Furthermore, we will investigate the potential for scalability and adaptability, envisioning how these applications could evolve to cater various device ecosystems.

This project would embark on an exciting journey into the realm of Android application development, where innovation, functionality, and user experience converge to redefine the boundaries of mobile technology. Together, we will unlock the potential of this dynamic pair of Android applications and pave the way for a future where our smartphones become even more versatile, intuitive, and indispensable in our daily lives.

**TECHNOLOGIES USED**

1. Figma (UI Design)

Figma is a vector drawing program. We can create scalable shapes that look great at any size and is great for collaborative work with the fellow teammates.

1. Flutter (Frontend and Backend)

Flutter is an open-source framework developed by Google for building natively compiled applications for mobile, web, and desktop from a single codebase.

1. Supabase (Realtime Database)

Supabase is an open-source platform that provides a set of backend services and tools for building web and mobile applications.

**USE CASES**

* Users can have multiple widgets on the screen at the same time, hence more data is available to the user as compared to the conventional single cluster screen where the user is only restricted to having a single screen thus limiting the experience.
* Instead of having every widget cluttered in one screen, users can arrange it according to personal preferences, which improves overall customer satisfaction.
* In the case of bikes this can act as a larger main display.
* Users having multiple widgets at once allows the riders not to frequently leave the bike handles.

**DETAILS OF WORK DONE**

The work for this project was divided into 3 major parts:

1. **Figma**: To design various UI elements such as the layouts of the two screens, various widgets with varied sizes.
2. **Flutter**: Flutter which uses Dart, was used to develop the frontend and the backend of our project.
3. **Communication Protocol**: This would play a significant role in providing communication between the two android applications.

**Problem Statement:**

To develop a pair of Android applications, with one serving as the primary cluster display and the other functioning as an extended display for the cluster interface.

* When extended display is connected to the cluster app, it will act like an extended display for the two-wheeler, else if not connected, the main cluster will be working independently.
* The various widgets which are present on both the screens, the user can interact with them in multiple ways.
* Use of TCP/IP for communication between the displays.

Our project was divided into 3 phases:

* **Phase 1**: Brainstorming, Deciding Widgets and Communication Protocols.
* **Phase 2**: Widget Designs, Layout Designs
* **Phase 3:** Static Flutter Application, Dynamic data retrieval from supabase ,UI enhancements and bug fixes.

**Phase 1:**

* **Brainstorming:**
* The Problem Statement was understood and noted. We discussed all the viable solutions to the noted problem statement.
* We discussed the design expectations and requirements, considered all the possible communication protocols that could be used, looked for various flutter animations that could be implemented and many such topics related to the problem statement were discussed.
* Finally, we all decided on some approaches, noted them, and divided the work equally within ourselves.
* **Layout Designs:** 
  + Choosing the correct layout was important as it would be the base on which the logic would be built and the designs of widgets and their sizes.
  + Among the various layouts that we discussed, some had design flaws, while others did not align with what we had in our mind, so we had to reject them. We finally went with the one which was the most practical and viable according to the requirements. The selected layout would work great on both the apps and was aligned with the communication logic that we had in our mind.
  + The selected layout on both the screens would be almost similar and is divided into 3 columns.
  + Following is the layout of the initial **Main Screen**:

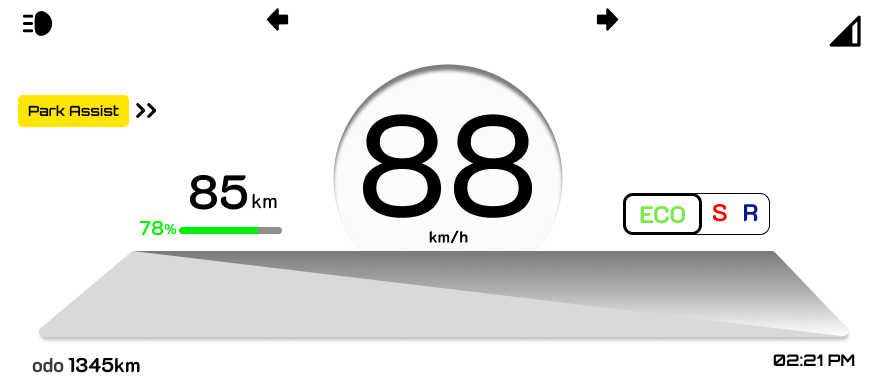


Figure: Initial Main Screen

* + The following is the layout after the user swipes the initial main screen revealing the fixed speedo-Odo at the left and the other widgets:

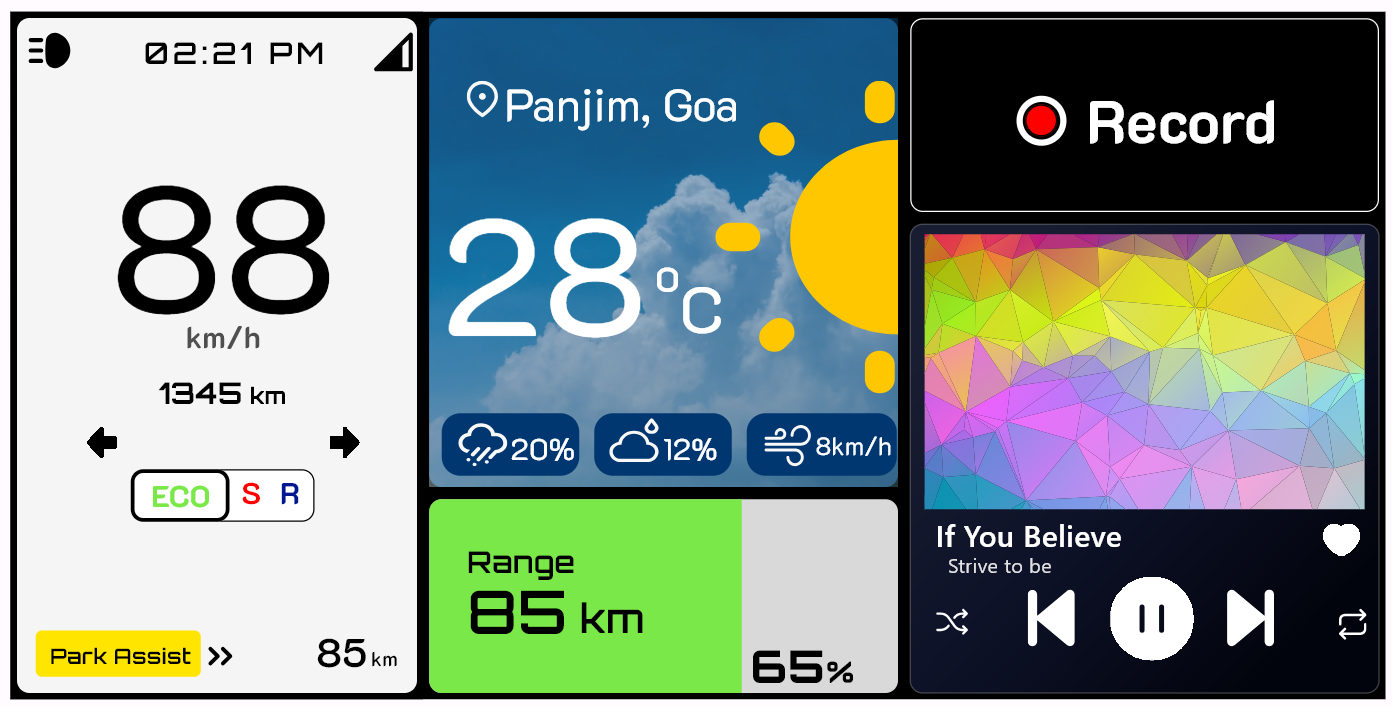


Figure: Main Screen with Widgets

* Following is the layout for the **Extended Screen**:

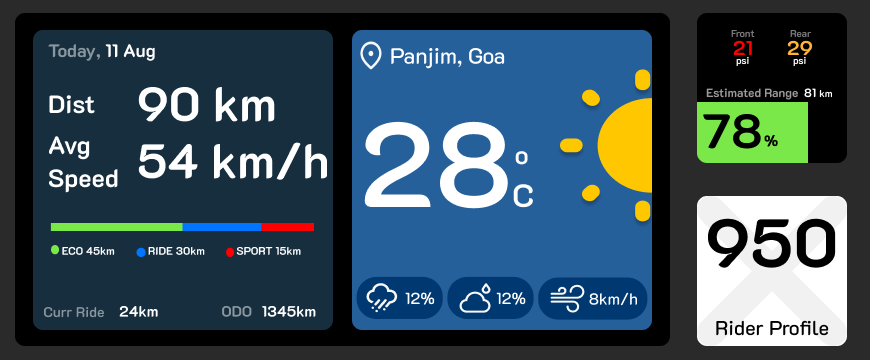


Figure: Extended Screen

* Following is the layout when a particular widget is double clicked, and the widget is seen in its large size and the rest widgets are to be seen on the right side and other widgets on the left.

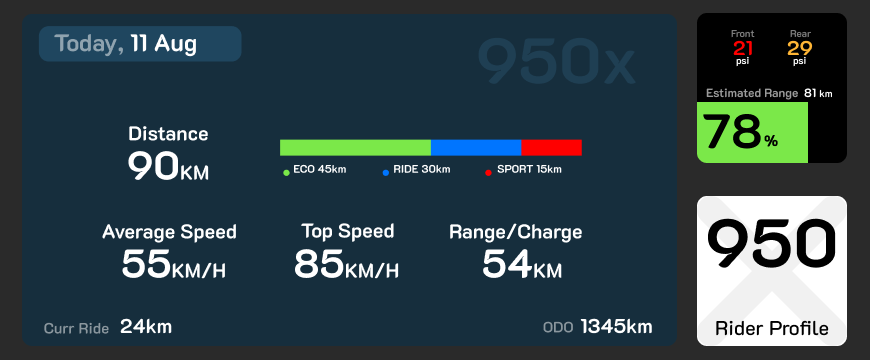


Figure: Extended Display (with large sized widget)

* **Widgets:**
* The next step after designing the layouts, was to decide what widgets were to be made. For this we browsed the internet looking for what widgets are currently offered by EV (electric vehicles) bike companies, we also discussed the same with our friends who owned EV bikes to understand more and to know what more we could offer.
* Finally, we listed down 8 widgets that we thought were important and at the same time would offer some look and feel.
* Following are the 8 widgets:
  1. **Navigation: Allows** the user to navigate through geographical locations.
  2. **Music Player:** Allows the user to control music related components.
  3. **Weather:** Gives the user various weather-related data such as temperature, precipitation, humidity probability.
  4. **Bike Stats:** Gives the user various bike related data such as battery percentage, battery health and condition, estimated range for various modes and tire pressure details.
  5. **Rider Profile:** This widget gives the user an almost complete insight of what the user has achieved with his bike, details such as total distance covered, top and average speed, charging related stats, rider profile, ranges the user was able to achieve on various bike modes.
  6. **Documents:** Allows the user to carry various important documents digitally.
  7. **Ride Stats:** This widget allows the user to get an insight of his day’s ride statistics, it contains data like, today’s total distance (mode-wise), average speed, top speed, odometer.
  8. **Trip Recorder:** This widget allows the user to explicitly start a trip session and to record the trip and thus be able to get various trip details such as total time and distance, his avg and top speed throughout the trip, the range he was able to get on various modes, etc.
* **Communication Protocol:**
  + The next major step after the basic layout and widget design was to set up communication between the cluster and the extended display, we had many options:
    - Bluetooth: Bluetooth is a wireless communication technology that enables short-range data exchange between electronic devices (BLE and Bluetooth classic.
    - Sockets: Sockets are a fundamental communication mechanism in computer networks and operating systems. They provide a programming interface for processes (applications or services) to communicate over a network, either on the same machine (local communication) or between different machines (remote communication).
  + We went forward with socket programming as it is more reliable, and we can share larger data on it.
  + We implemented TCP/IP for communication using socket programming in dart.
  + Data used for communication:

|  |  |
| --- | --- |
| **Data** | **Interpretation** |
| DeviceBrand deviceName c | Connection request |
| [widgetId ...] [parentId ...] w | Initial widget transfer from Cluster to Extended |
| ParentId parentId h | Highlight the parent widget |
| ParentId parentId hc | Cancel all the highlights |
| WidgetId parentId d | Widget as data |
| WidgetId parentId a | Widget as ack |
| Disconnected DeviceBrand deviceName | Connection closed |

**Phase 2:**

1. **Widget Designs**

* Once the decision of which widgets were to be made was complete, we started with the Figma designs for the widgets. Following are the designs:
* **Extended Screen Widgets**

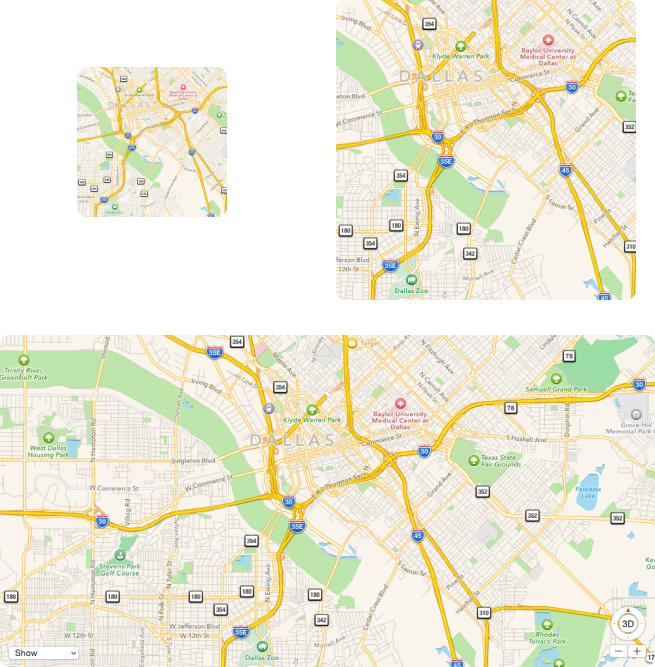
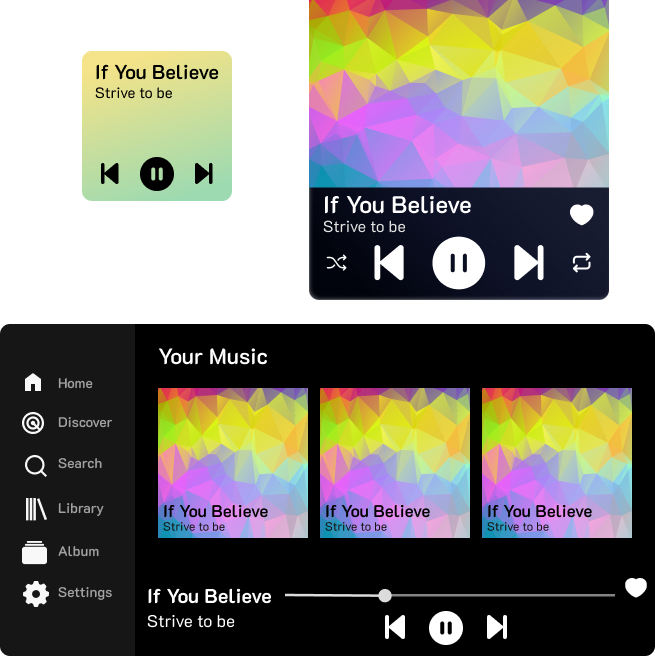
 

Figure: Navigation Figure: Music Player

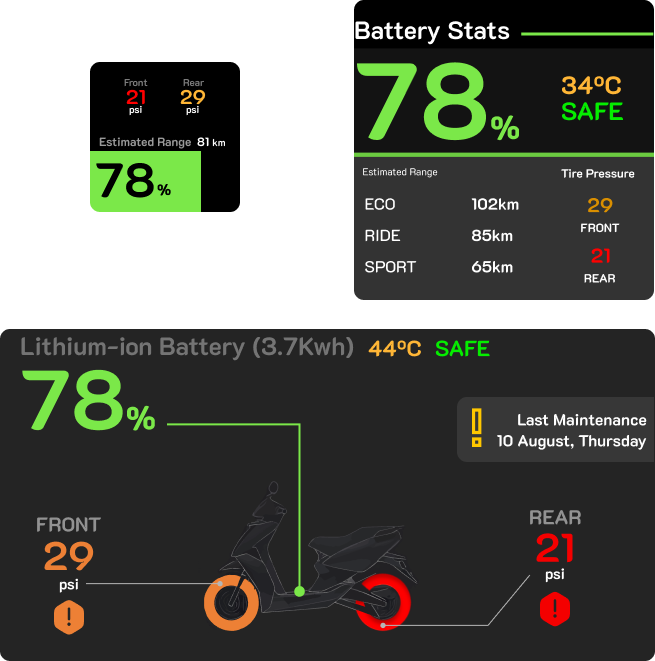
 

Figure: Weather Figure: Bike Stats

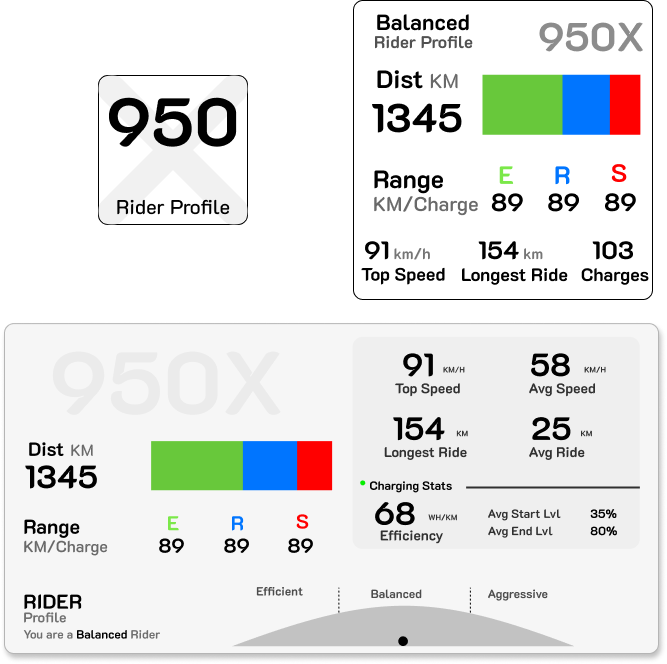
 

Figure: Rider Profile Figure: Documents

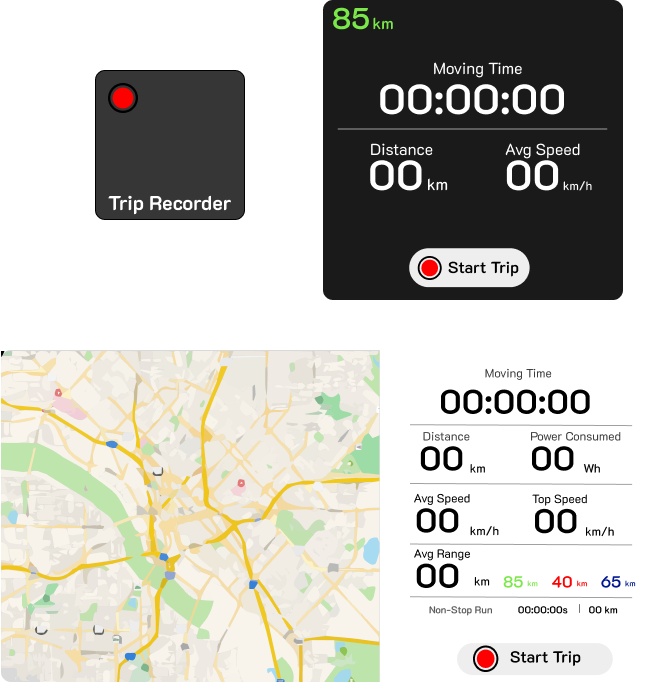
 

Figure: Ride Stats Figure: Trip Recorder

* **Main Screen Widgets**

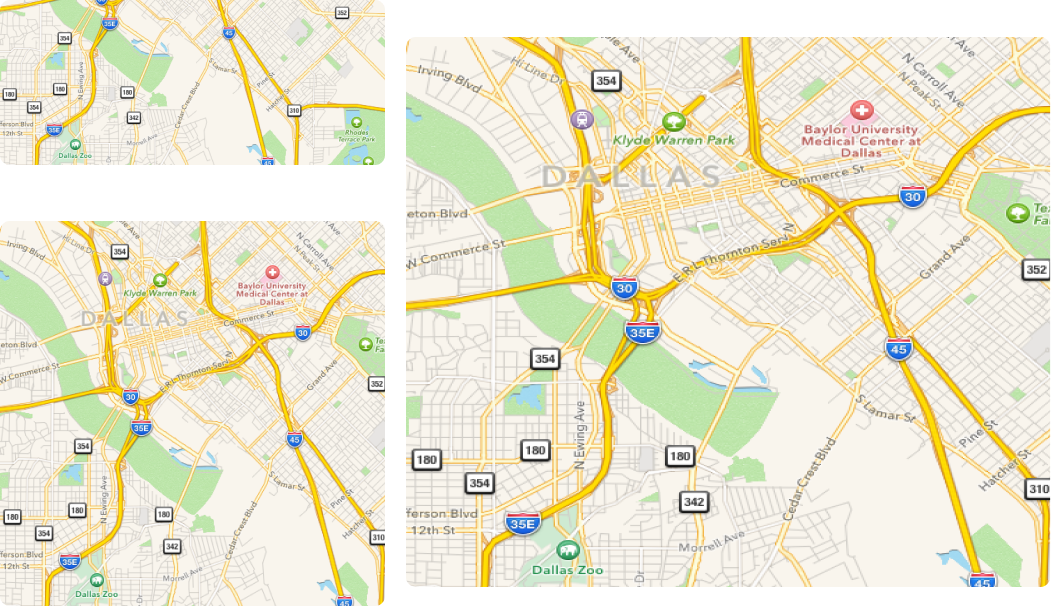
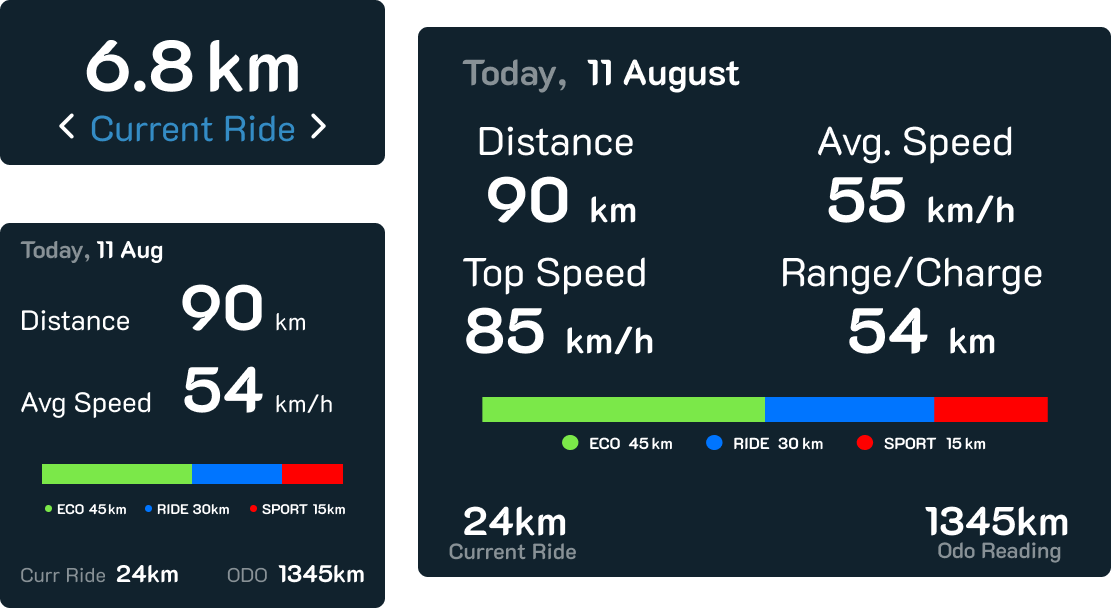
 

Figure: Navigation Figure: Ride Stats

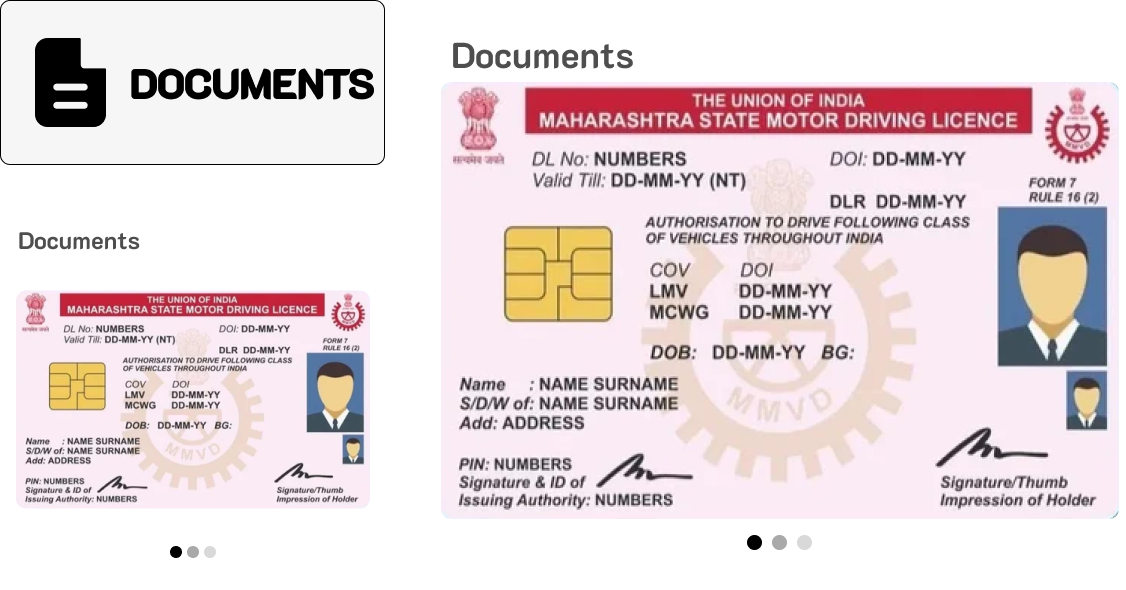
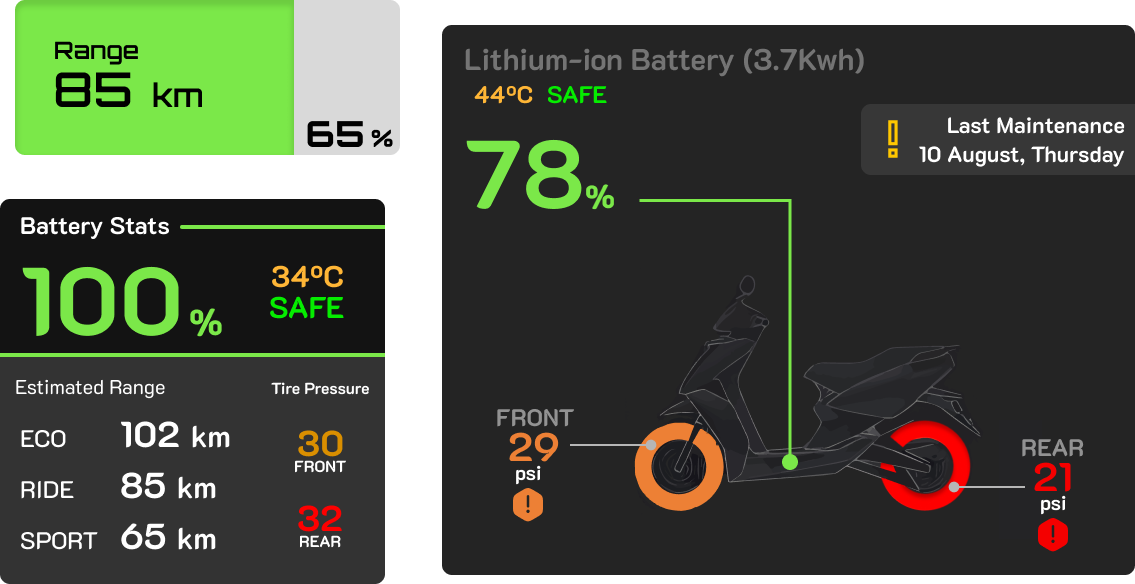
 

Figure: Documents Figure: Bike Stats

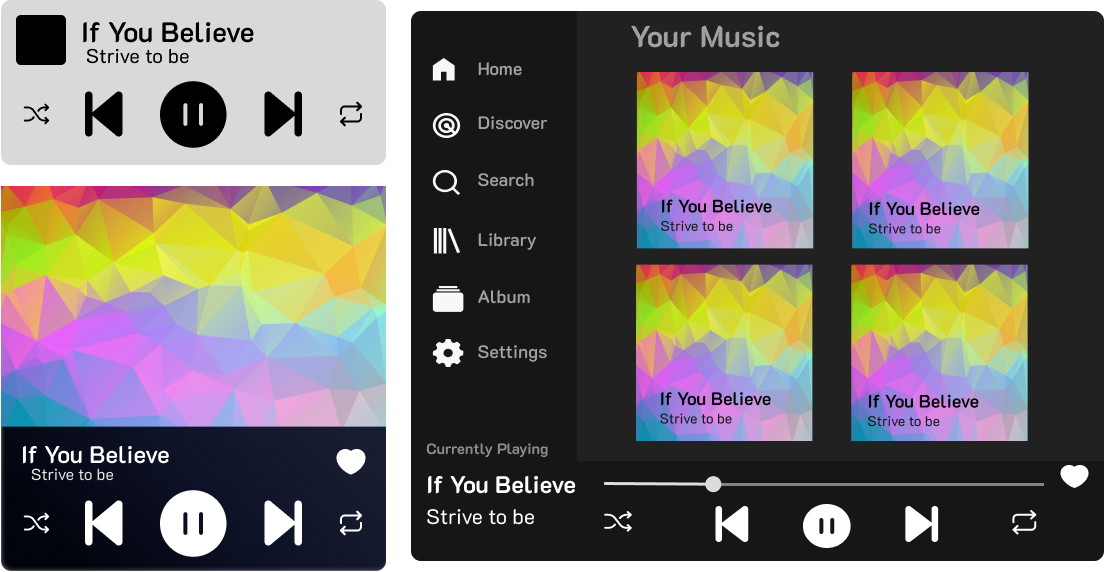
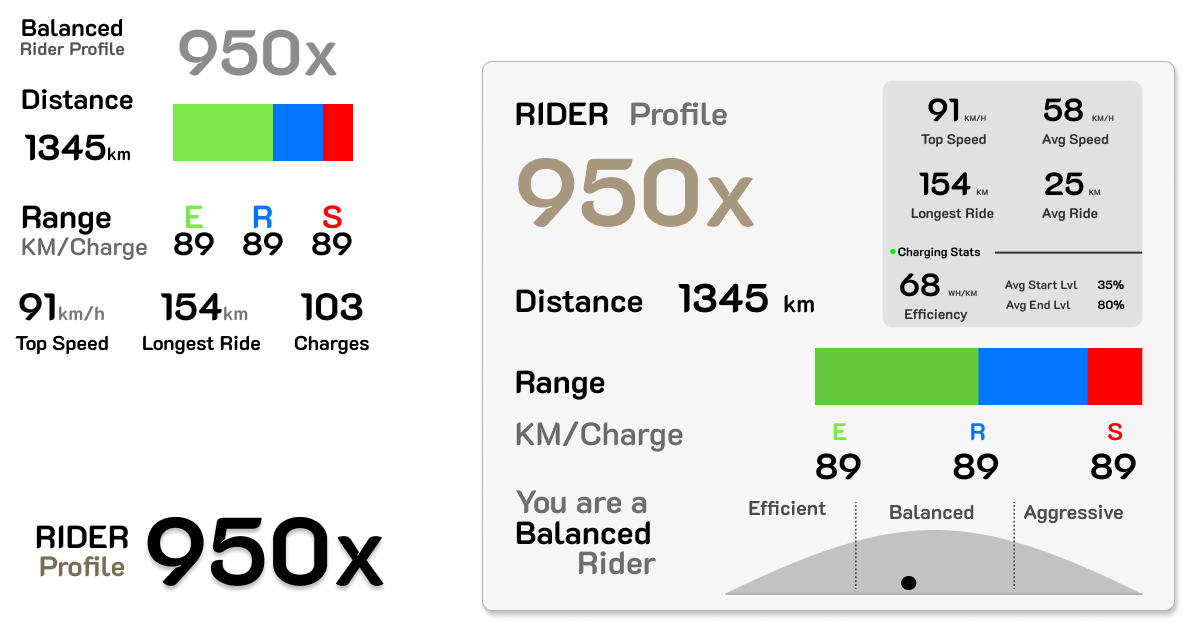
 

Figure: Music Player Figure: Rider Profile

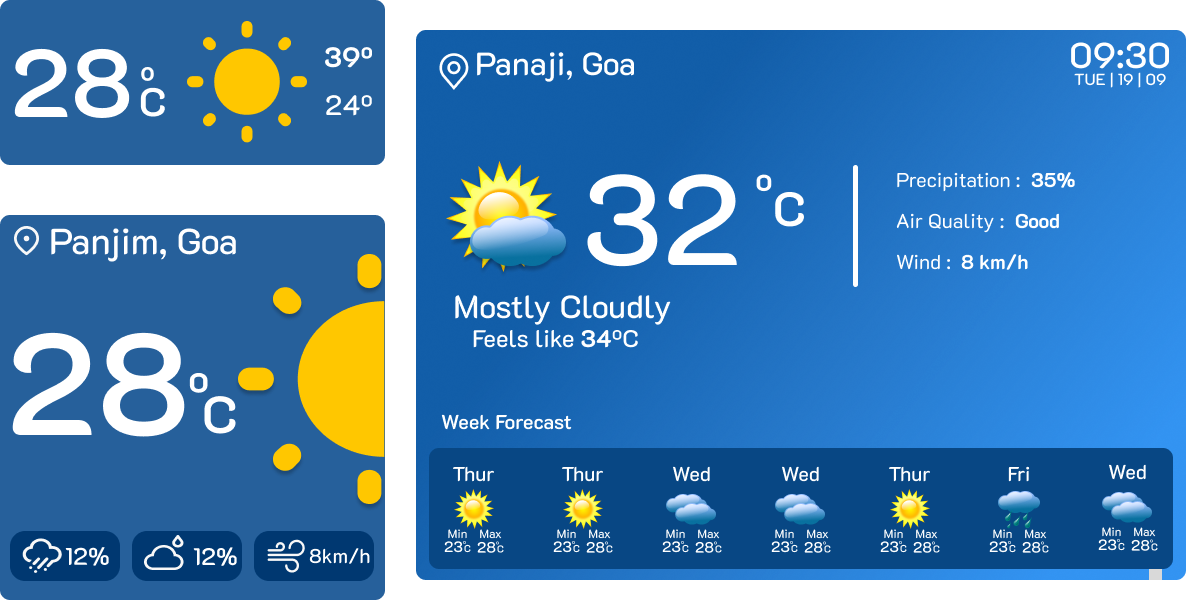
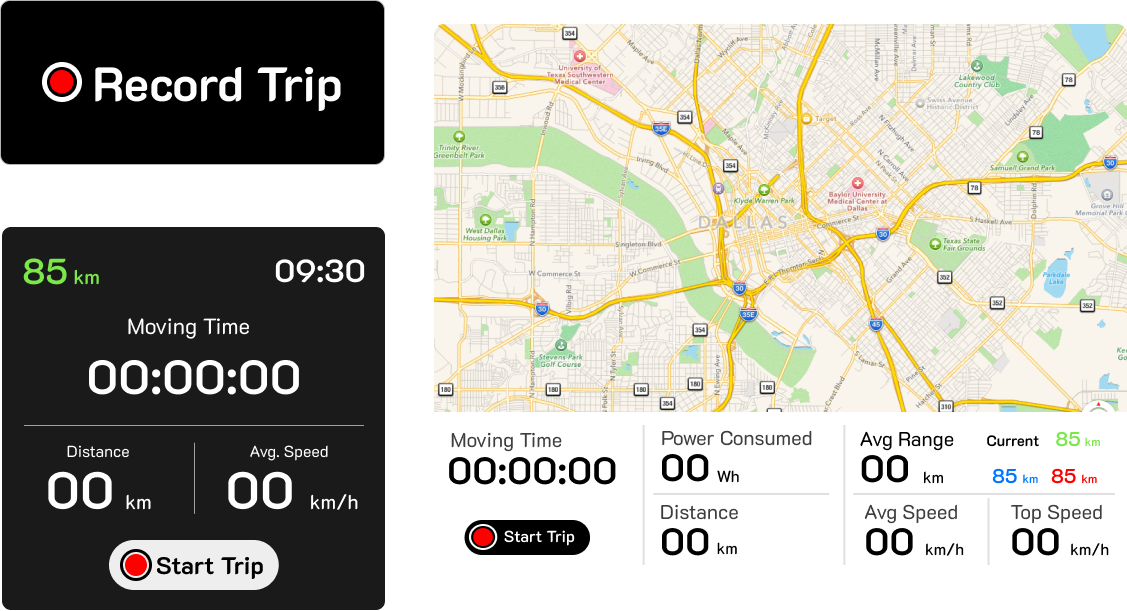
 

Figure: Weather Figure: Trip Recorder

**2. Basic Flutter Application**

* This version of the flutter app was only a statice version of the Figma designs.
* The widgets code for flutter was generated using a plugin from Figma called to Code (HTML, Tailwind, Flutter, SwiftUI)
* Here widgets could be resized by taping on them.
* The transition on between was achieved using AnimatedCrossFade widget.
* When the extended is not connected there is a scroll which was achieved using flutter package carousel\_slider.

**3. Dynamic data retrieval from supabase**

* Supabase is an open-source replacement for firebase (did not use as it is not compatible with Windows's (Operating System).
* Supabase database uses PostgreSQL and also provides us with real-time database in the same.
* Here we used supabase as a backend service that is responsible for listening to the changes in the database and making the changes accordingly. Currently we have implemented it on only a single widget, but we can extend it to other widgets thus making it dynamic and more appealing.

**CHALLENGES FACED**

* Major Challenges faced while designing layouts and widgets:
  + As the two screens had different viewport sizes and layouts, the widget sizes and designs were to be made such that they could very well align with the logic and layout and at the same time look appealing. As a solution we had to think of various design changes we could make, aligning various UI components within the widgets and trying multiple designs before finalizing each widget.
  + Designing the layout for the main screen was straightforward but thinking about a layout for the extended screen was quite a challenge. As the fixed speedo-Odo widget was not to be added to the extended screen, it was a challenge to fix the similar widgets in the space in way that the widgets would still look appealing.
  + To solve the above challenge, we tried multiple different layouts and arrived at a layout that looked good, the only challenge there too was that it needed different sized widgets. So, we designed the required widget sizes and finally the layout for the extended screen looked great.
  + While developing the basic flutter app the major challenge was making the animation smooth this was doing using the Animated widgets in flutter and changing the widgets width instead of changing the visibility.

**FUTURE WORK**

* Communication with Bike Intruments
  + For now, whatever data that is being displayed on the screens is hardcoded. Thus, if we can have a communication protocol that allows the mobile app to communicate with the bike instruments, allowing to get various bike related data such as, speed, battery levels and other details, then the app could be perfectly accurate and dynamic.
* Enhancing the UI and Advanced Animations
  + We were able to achieve some great UI and animations in our apps, but we had some more ideas that could have further elevated the user experience even more like an even more intuitive UI, more enhanced animations when the user interacts with the UI in multiple ways.
* Using Bluetooth as a mode of Communication.
  + Providing multiple modes of communication based on if the extended is connected or not.
  + If the extended device is connected the communication can be Bluetooth if not it can be TCP (Transmission Control Protocol).

**CONCLUSION**

In summary, our project represents an important turning point for cluster display technology. We successfully designed a stable and visually appealing mobile cluster display application by seamlessly combining Figma, Flutter, and TCP/IP communication. This software not only delivers real-time data updates, but it also allows for smooth connectivity with the car Cluster software Server, boosting the user experience in the electric car industry.

Looking ahead, we see several promising chances for continued development. Advanced data visualization techniques, customization and personalization features, the incorporation of external APIs to expand functionality, the investigation of voice and gesture controls for improved user interaction, ongoing performance optimization, and the inclusion of user feedback for continuous improvement are a few of these.

These progressive activities will make sure that our cluster display stays at the cutting edge of innovation and provides an outstanding user experience that satisfies the changing requirements of the mobile car application landscape. By attempting to give consumers a comprehensive and user-centric tool that is yet flexible and responsive to their demands, our initiative paves the way for further developments in this field.

**REFERENCES**

Socket Programming (dart):

<https://api.dart.dev/stable/3.1.3/dart-io/Socket-class.html>

<https://api.dart.dev/stable/3.1.3/dart-io/ServerSocket-class.html>

<https://medium.com/flutter-community/working-with-sockets-in-dart-15b443007bc9>

Supabase with flutter:

<https://supabase.com/docs/reference/dart/introduction>