| **A**  **PROJECT REPORT ON** |
| --- |
|  |
|  |
| Vehicle Management System |
| SUBMITTED IN  PARTIAL FULFILLMENT OF  **DIPLOMA IN ADVANCED COMPUTING (PG-DAC)** |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
| **BY**  **AbhiShek Shinde**  **Sanket Solanke**  **Tushar Vashistha**  **Yash Jawalkar** |
|  |
|  |
| **UNDER THE GUIDANCE OF**  **Shubham Borle** |
|  |
|  |
|  |
| **AT**  **SUNBEAM INSTITUTE OF INFORMATION TECHNOLOGY,**  **PUNE** |

| **SUNBEAM INSTITUTE OF INFORMATION TECHNOLOGY,**  **PUNE.** | | | |
| --- | --- | --- | --- |
|  | | | |
|  | | | |
|  | | | |
|  | | | |
| **CERTIFICATE** | | | |
|  | | | |
| This is to certify that the project | | | |
|  | | | |
| Vehicle Management System | | | |
|  | | | |
| Has been submitted by | | | |
|  | | | |
| **Abhishek Shinde**  **Sanket Solanke**  **Tushar Vashistha**  **Yash Jawalkar** | | | |
|  | | | |
|  | | | |
| In partial fulfillment of the requirement for the Course of **PG Diploma in Advanced Computing (PG-DAC Sep 2023)** as prescribed by The **CDAC** ACTS, PUNE. | | | |
|  | | | |
|  | | | |
| Place: Pune | | | Date: 22-FEB-2024 |
|  | | | |
|  | | | |
|  | | | **Shubham Borle** |
|  | | | **Project Guide** |
|  |  |

**ACKNOWLEDGEMENT**

*We would like to express Our special thanks of gratitude to our teacher Shubham Borle as well as our Director Nitin Khudale who gave us the golden opportunity to do this wonderful project on the topic Vehicle Management System, which also helped us in doing a lot of Research and we came to know about so many new things we are really thankful to them.*

Abhishek Shinde

Sanket Solanke

Tushar Vashistha

Yash Jawalkar

**ABSTRACT**

The Vehicle Management System (VMS) web application presented in this report represents a comprehensive solution designed to streamline and enhance the operational efficiency of businesses managing a fleet of vehicles. Our system provides a user-friendly interface catering to the needs of fleet managers.

The key features of the VMS include real-time insights into the fleet's status, presenting the manager with counts of available, unavailable, and vehicles under maintenance. This information empowers decision-makers to optimize resource allocation and maintain an agile fleet that meets the demands of their business.

Furthermore, the application facilitates order management by categorizing orders into approved, pending, and rejected, allowing managers to monitor the workflow seamlessly. The integration of driver management alongside order tracking adds an extra layer of control, ensuring that both vehicles and drivers are efficiently coordinated to meet operational requirements.

|  | **INTRODUCTION** | 1 |
| --- | --- | --- |
|  | 1.1 Introduction | 2 |
|  | **PRODUCT OVERVIEW AND SUMMARY** |  |
|  | 2.1 Purpose |  |
|  | 2.2 Scope |  |
|  | 2.3 User Classes and Characteristics |  |
|  | 2.4 Design and Implementation Constraints |  |
|  | **REQUIREMENTS** |  |
|  | 3.1 Functional Requirements |  |
|  | 3.1.1 Use case for Manager. |  |
|  | 3.1.2 Use case for Customer. |  |
|  | 3.2 Non - Functional Requirements |  |
|  | 3.2.1 Usability Requirement |  |
|  | 3.2.2 Performance Requirement |  |
|  | 3.2.3 Reliability Requirement |  |
|  | 3.2.4 Portability Requirement |  |
|  | 3.2.5 Security Techniques |  |
|  | **PROJECT DESIGN** |  |
|  | 4.1 Data Model |  |
|  | 4.1.1 Database Design |  |
|  | 4.2 Process Model |  |
|  | 4.2.1 Functional Decomposition Diagram |  |
|  | 4.2.2 Data Flow Diagram (DFD) |  |
|  | **PROJECT RELATED STATISTICS** |  |
|  | **CONCLUSION** |  |

**INDEX**

**LIST OF TABLES**

| **Section** | **Table Title** | **Page** |
| --- | --- | --- |
| **Fig 3** | **Complete Database** |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |

**LIST OF FIGURES**

| **Section** | **Figure Title** | **Page** |
| --- | --- | --- |
| **Fig 1-2** | **Customer & Manager Use Case Diagram** |  |
| **Fig 3** | **Data Flow Diagram** |  |
| **Fig 4 -9** | **Dashboard Screen Shots** |  |
| **Fig 10-13** | **Mobile Application Screen Shots** |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |

**INTRODUCTION**

In an era driven by rapid globalization and dynamic market demands, efficient fleet management has become a critical aspect of success for businesses relying on transportation. Recognizing the pivotal role that vehicles play in daily operations, our project introduces a comprehensive Vehicle Management System (VMS) web application designed to empower businesses with the tools needed to manage their fleets effectively.

The VMS addresses the multifaceted challenges faced by fleet managers by providing real-time insights into the status of vehicles, orders, and drivers. Through a user-friendly interface, managers gain access to a dashboard that offers a detailed count of available, unavailable, and under-maintenance vehicles, allowing for precise resource allocation and strategic planning.

Order management is streamlined through the categorization of orders into approved, pending, and rejected, providing a holistic view of the workflow. Additionally, the integration of driver management ensures a synchronized coordination of both vehicles and drivers, fostering operational cohesion.

This report unfolds the journey of conceptualizing, designing, and developing the VMS, delving into the technical aspects, methodologies, and user-centric considerations that shaped its creation. By offering a scalable, secure, and responsive solution, the VMS aims to not only meet but exceed the expectations of fleet managers, providing them with a powerful tool to navigate the complexities of modern-day transportation management.

**The goal of this project:** The primary goal of the Vehicle Management System (VMS) project is to provide businesses with a robust and intuitive web application that effectively addresses the challenges associated with fleet management. The project aims to empower fleet managers by offering a comprehensive toolset that enhances operational efficiency, optimizes resource utilization, and ensures seamless coordination between vehicles, orders, and drivers.

*Real-time Fleet Visibility*: Provide a platform that offers instant and accurate insights into the status of vehicles, enabling fleet managers to make informed decisions regarding their availability, maintenance, and overall utilization.

*Order Workflow Optimization*: Streamline the management of orders by categorizing them into approved, pending, and rejected, facilitating a clear and efficient workflow. This goal aims to enhance the overall order processing and fulfillment process.

*Driver Coordination*: Integrate driver management features to ensure a synchronized coordination between vehicles and drivers. This includes tracking driver assignments, schedules, and performance to optimize the workforce associated with the fleet.

*User-Friendly Interface*: Design an intuitive and user-friendly interface that caters to the needs of fleet managers, making the VMS accessible to users with varying levels of technical expertise. User experience considerations are prioritized to enhance usability.

**Product Overview and Summary**

**| Purpose:** The purpose of the Vehicle Management System (VMS) project is to develop a user-friendly web application that empowers businesses to efficiently manage their fleets. The VMS aims to provide real-time visibility into vehicle status, streamline order workflows, optimize driver coordination, and offer a scalable solution to enhance overall operational effectiveness. By addressing the challenges associated with fleet management, the project seeks to contribute to improved resource utilization, streamlined processes, and informed decision-making for businesses reliant on effective transportation.

**| Scope**: The scope of the Vehicle Management System (VMS) project encompasses the development and implementation of a comprehensive web application designed to facilitate efficient fleet management for businesses. The key components within the project's scope include:

*Fleet Status Management*:

1.Real-time tracking of vehicle availability, unavailability, and those under maintenance.

2.Instant access to the count and status of vehicles for timely decision-making.

*Order Management*:

1.Categorization of orders into approved, pending, and rejected for streamlined workflow.

2.Monitoring and tracking of order status to enhance order processing efficiency.

*Driver Coordination*:

1.Integration of driver management features for effective coordination with vehicle assignments.

2.Tracking driver schedules, and associated tasks to optimize workforce utilization.

*User Interface*:

1.Development of an intuitive and user-friendly interface for fleet managers.

2.Dashboard displaying key metrics and insights for quick and informed decision-making.

**User Classes and Characteristics**:

**1. Manager**:

Characteristics: Managers are responsible for overseeing the entire fleet operations. They require comprehensive insights into vehicle status, order management, and driver coordination.

Functionality: Access to real-time dashboards, analytics, and tools for efficient decision-making. They can manage orders, track vehicle status, assign drivers, and view detailed reports.

**2. Customer:**

Characteristics: Customers initiate transportation requests by placing orders through the VMS.

Functionality: Access to a user-friendly interface for submitting order details.

**| Design and Implementation Constraints**

**- User Interface**

In order to effectively incorporate each of the three focus areas of our application (Vehicle, Order and Driver Management) into the user interface without creating too much clutter, we opted to use a sidebar-driven design. When the app is initially opened, the user is taken to the “Dashboard” tab.

**- Sidebar-driven Design**: Relatively early in the development process, we came to the decision to separate each data aspect (Vehicles, Drivers, and Orders) into individual tabs within the app. Each of the four tabs is implemented in its own class, and is a subclass of the Fragment class. In addition, the tabs share a single Android activity, which is the Android class that handles all user interaction with the app. The main activity of the application manages the tab layout and tells the application which tab view to display when a user selects a certain tab. The purpose of using fragments is to eliminate the need to create a new activity every time the user switches between tabs, thus increasing both the temporal and spatial efficiency of the application. Structurally, each fragment is comprised of two components. The first is an XML file that defines the visual layout of the fragment. The second is the logic of the fragment, which contains various functions defining certain actions to take at different times. Each time a certain fragment is switched to or away from, these functions are called in a particular order, all of which make up what is called the fragment’s “lifecycle”. For example, when a particular fragment is displayed on the screen, the function onCreateView() is called, which is generally where most of the initialization process occurs

**Functional Requirements**

**| Use Case for Customer**:

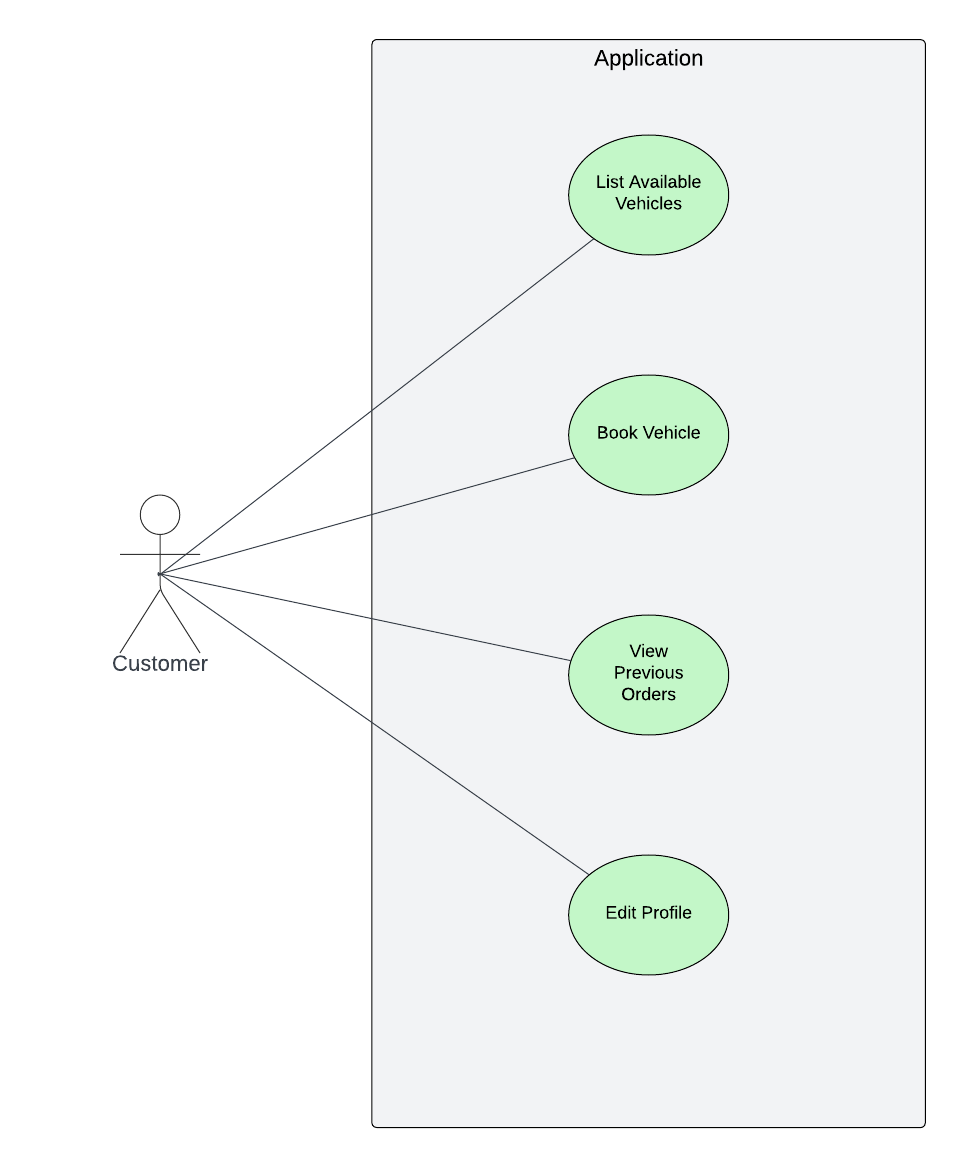


Fig. 1

**| Use Case for Manager**:

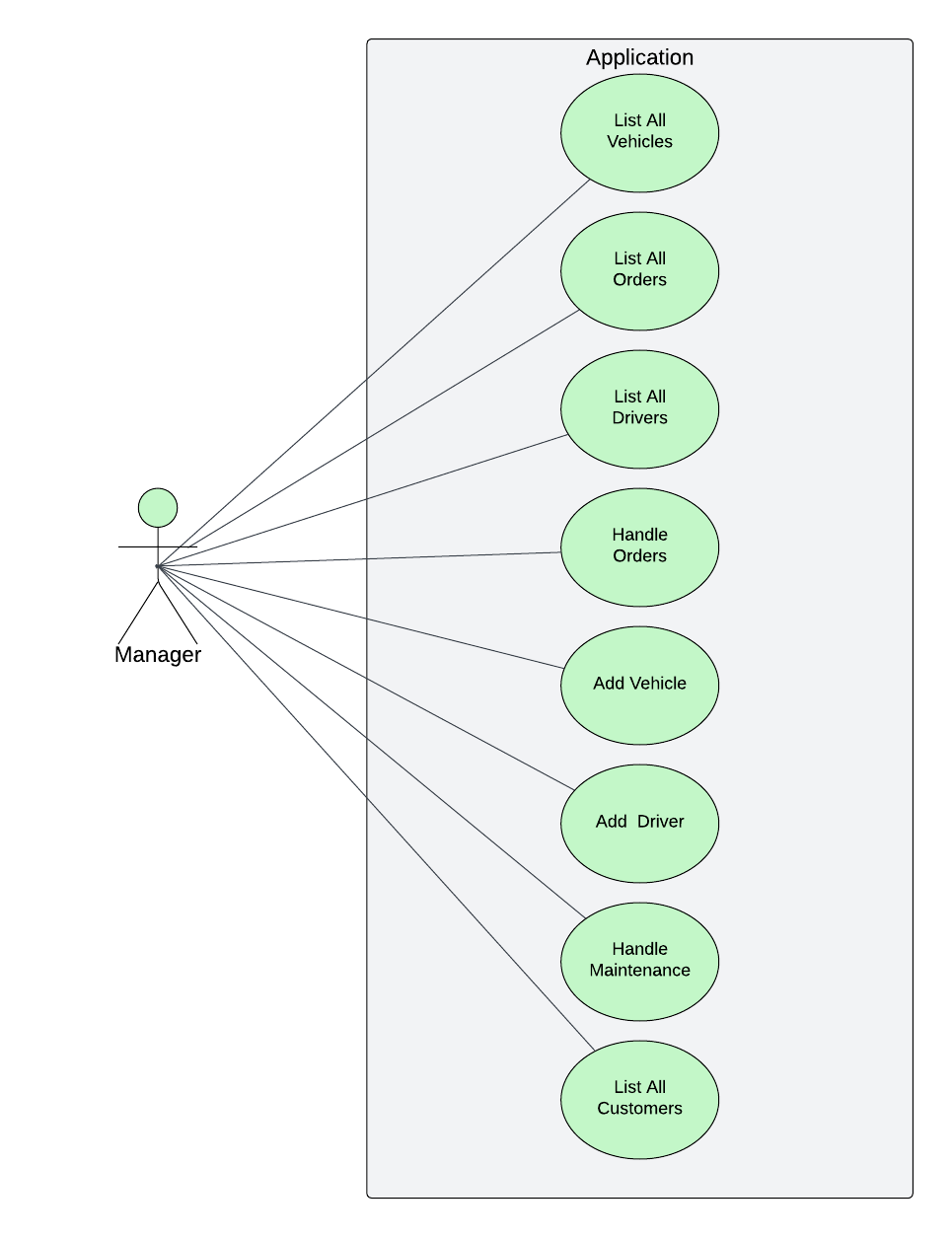


Fig. 2

**Non - Functional Requirements**

**Usability Requirement:**

The user interface should be intuitive and user-friendly, catering to users with varying levels of technical expertise.

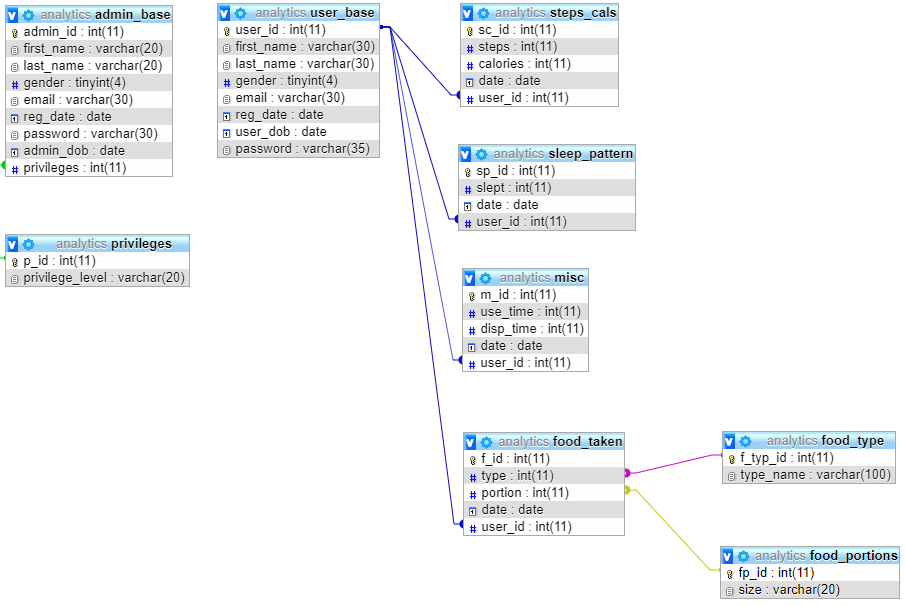
Accessibility: Ensure that the application is accessible to users with disabilities, adhering to relevant accessibility standards.

**Compatibility:**

The application should be compatible with different web browsers, operating systems, and devices to ensure a broad user reach.

**Data Model**

Database In order to effectively store the user’s step counting and sleep tracking history, we decided to use a SQL database, which is already built into the Android operating system. Because Google Fit integration automatically manages the step counting data, our database only has two tables, one for mood data and another for sleep data. The Database Schema for Sleep and Mood Data The first step in implementing the database was to create objects that could be used to store and retrieve information for each data type. More specifically, we created a sleep tracking class and a mood tracking class. Each class has two variables, as well as methods for setting and getting these variables. As seen in Figure 15, the sleep counting class contains a variable for the number of hours slept the night before, and the mood tracking class stores a number ranging from one to ten (where one represents the least happy, and ten represents the happiest). Each class also contains a variable representing the date at the time the data was stored. We then use these classes to periodically store the numerical data. For example, when the application detects that the user has woken up in the morning, it will calculate the number of hours that he or she has slept. It then creates an instance of the sleep counting class with that number of hours, and the date of the day before. It will then use that instance to create a SQL query that inputs the data into the database. Finally, it will use that same information to update the information displayed to the user. One of the features of the Google Fit API is that it conveniently stores the collection of fitness data that it records in the Google Fit Store. This is a cloud server that the data is sent to. The data can be easily accessed by sending a data read request to the server. If the phone is offline when the request is sent, it will store the request and wait until it has once again been reconnected to the internet to retrieve the data. Due to this integration of data storage built directly into the API, the SQLite database, which we had previously used, was no longer necessary for the storage of step tracking data. Instead, the app periodically retrieves aggregated step counting data from the Google Fit Store, and uses it to update the graph as well as the current number of steps for that day.



Complete Database (fig 3)

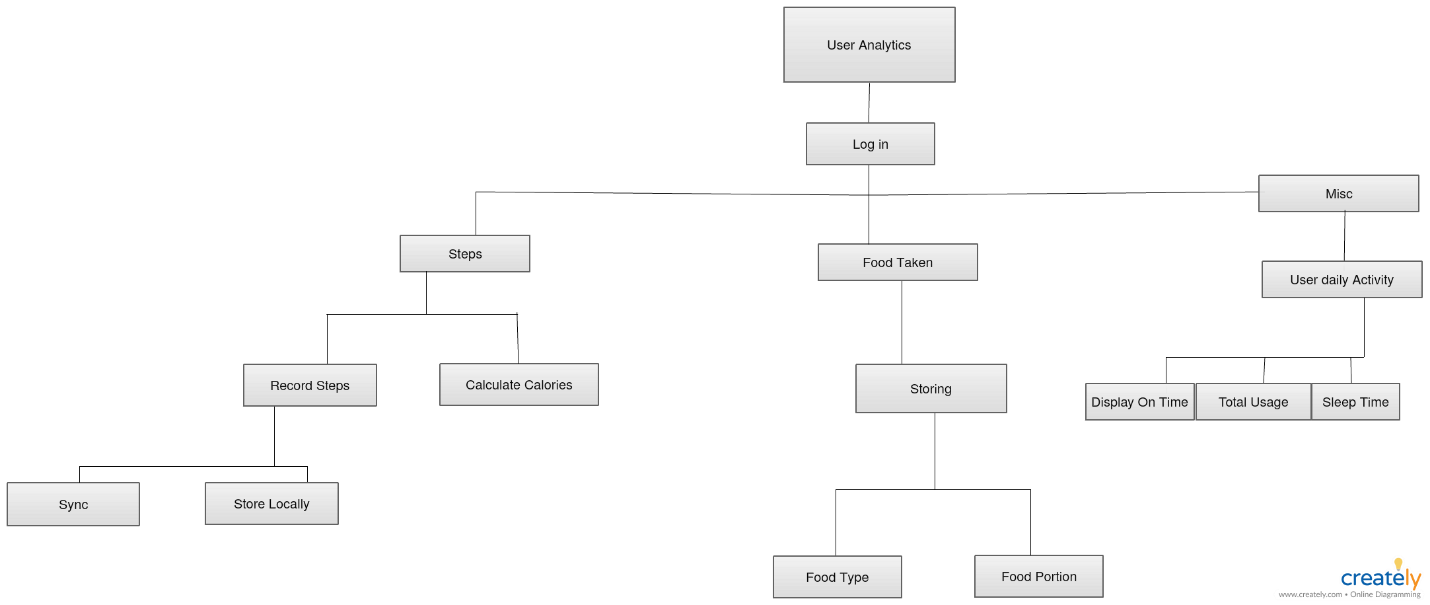


Fig. 3

**Screen Shots**

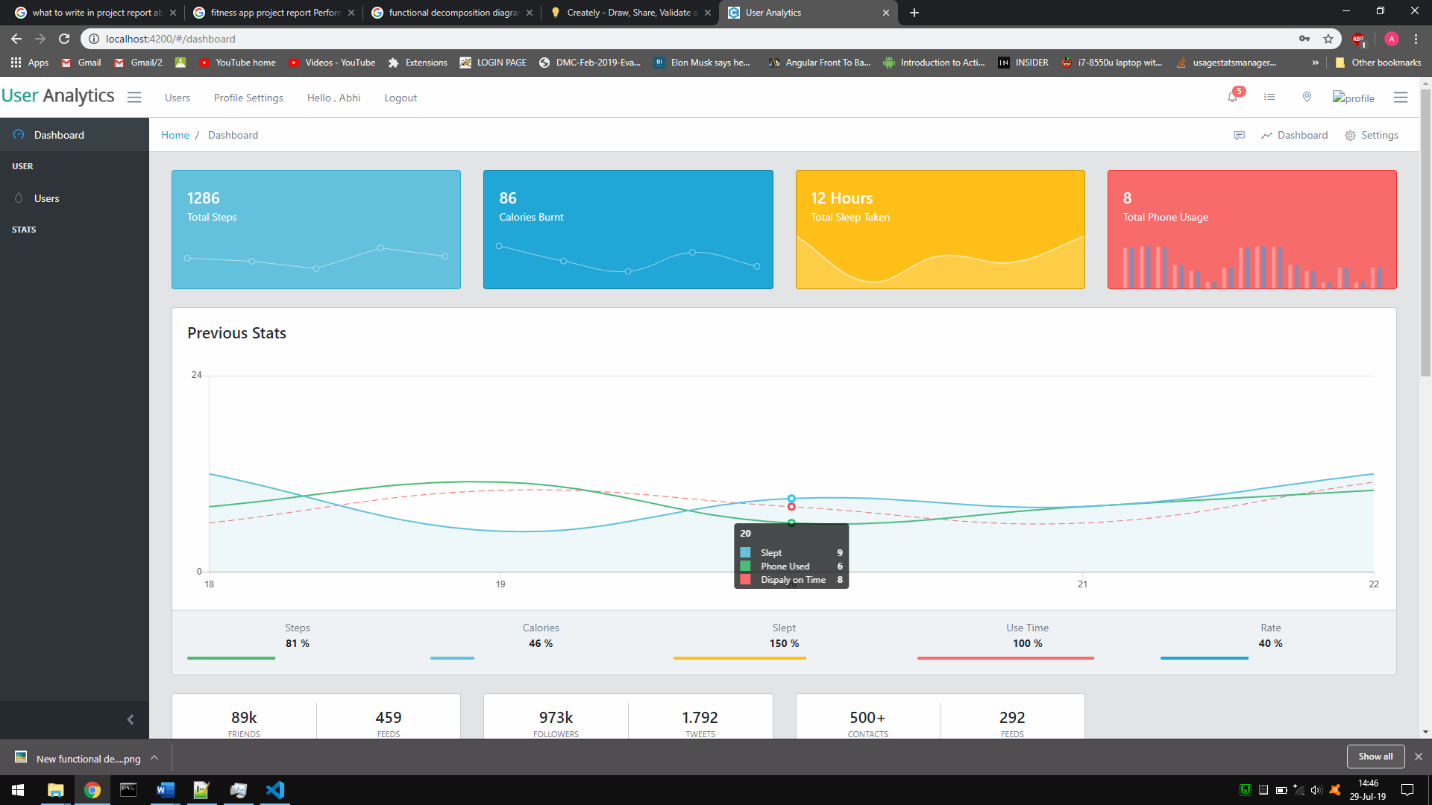


Fig. 4

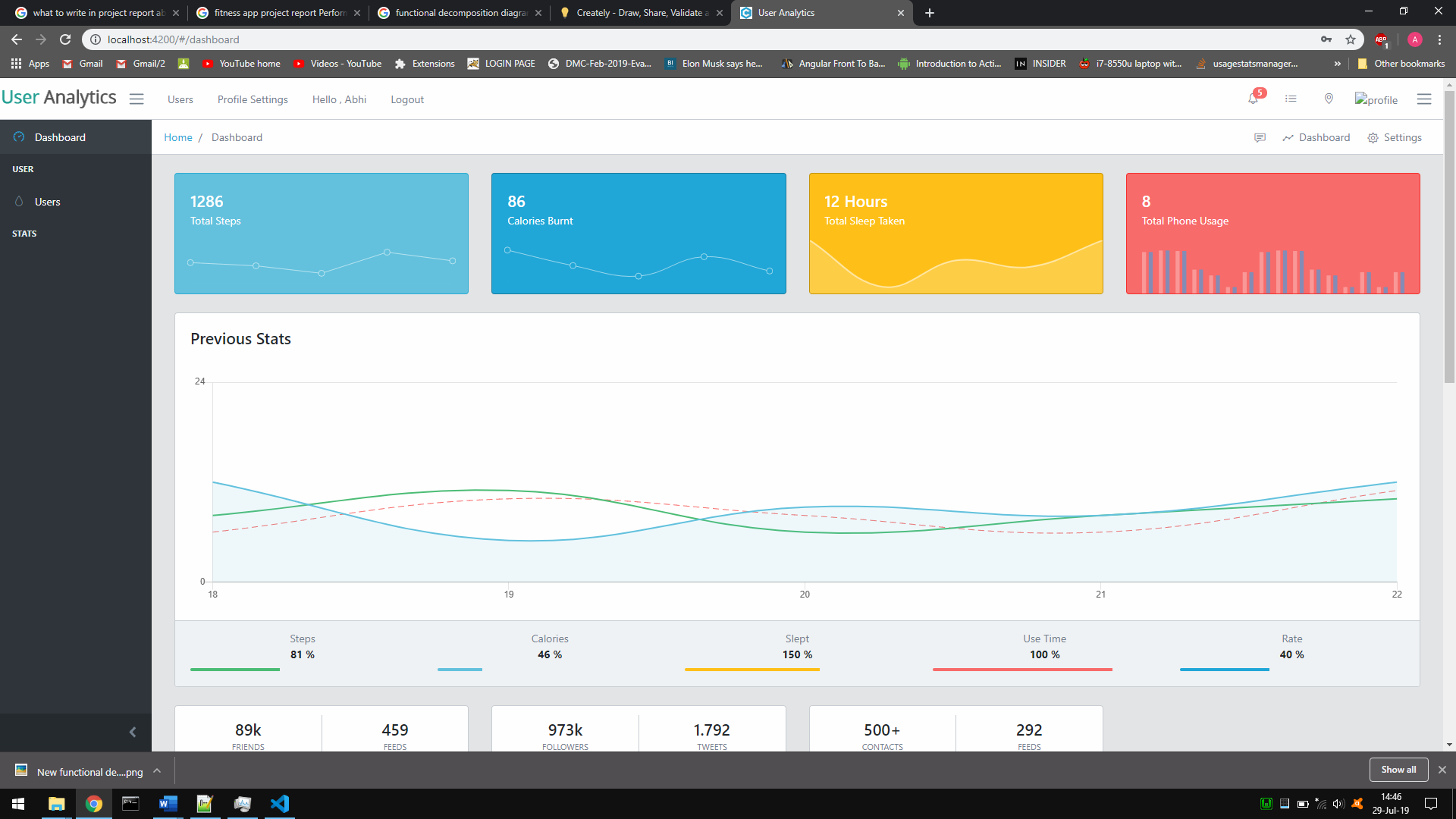


Fig. 5

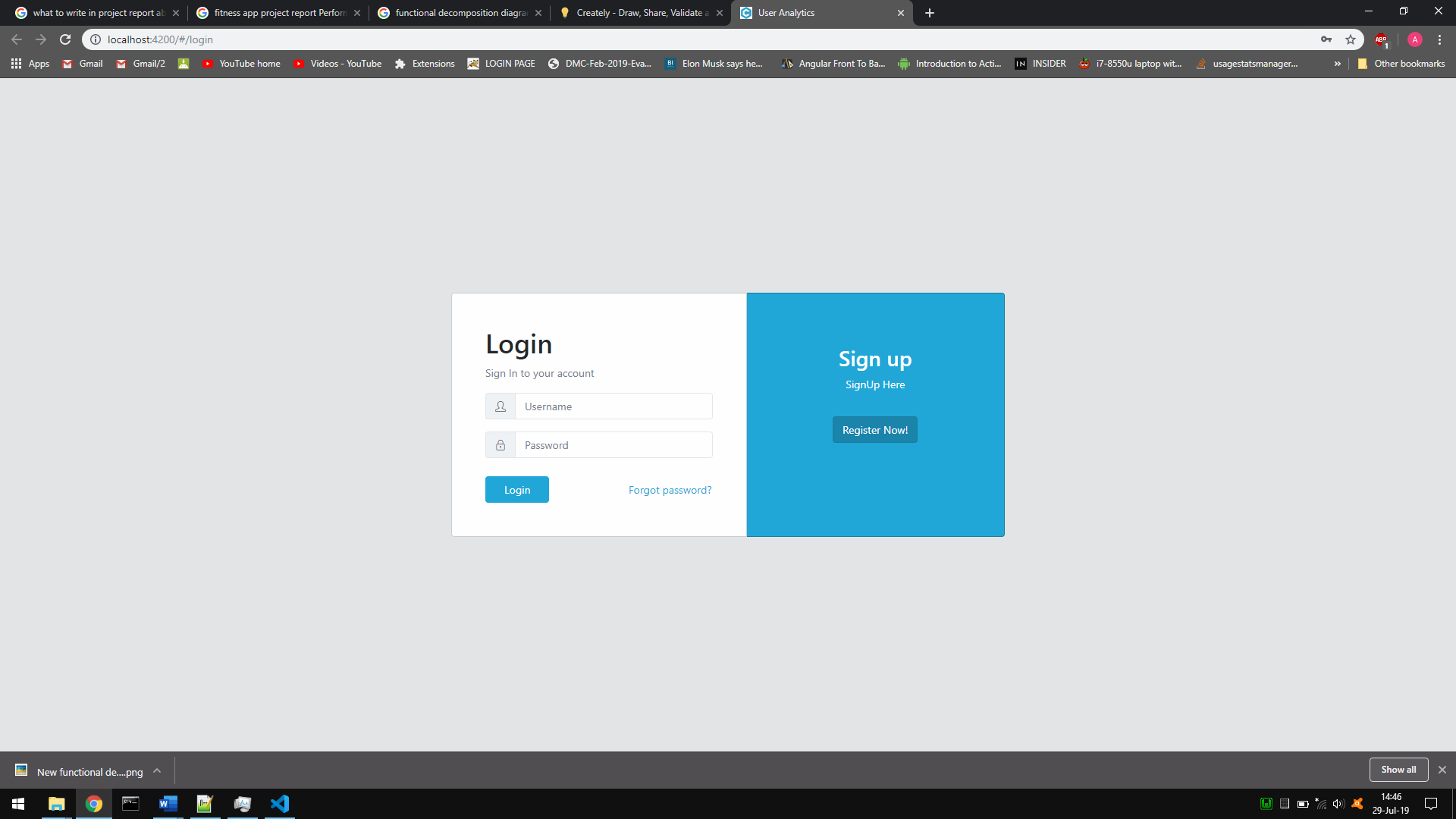


Fig. 6

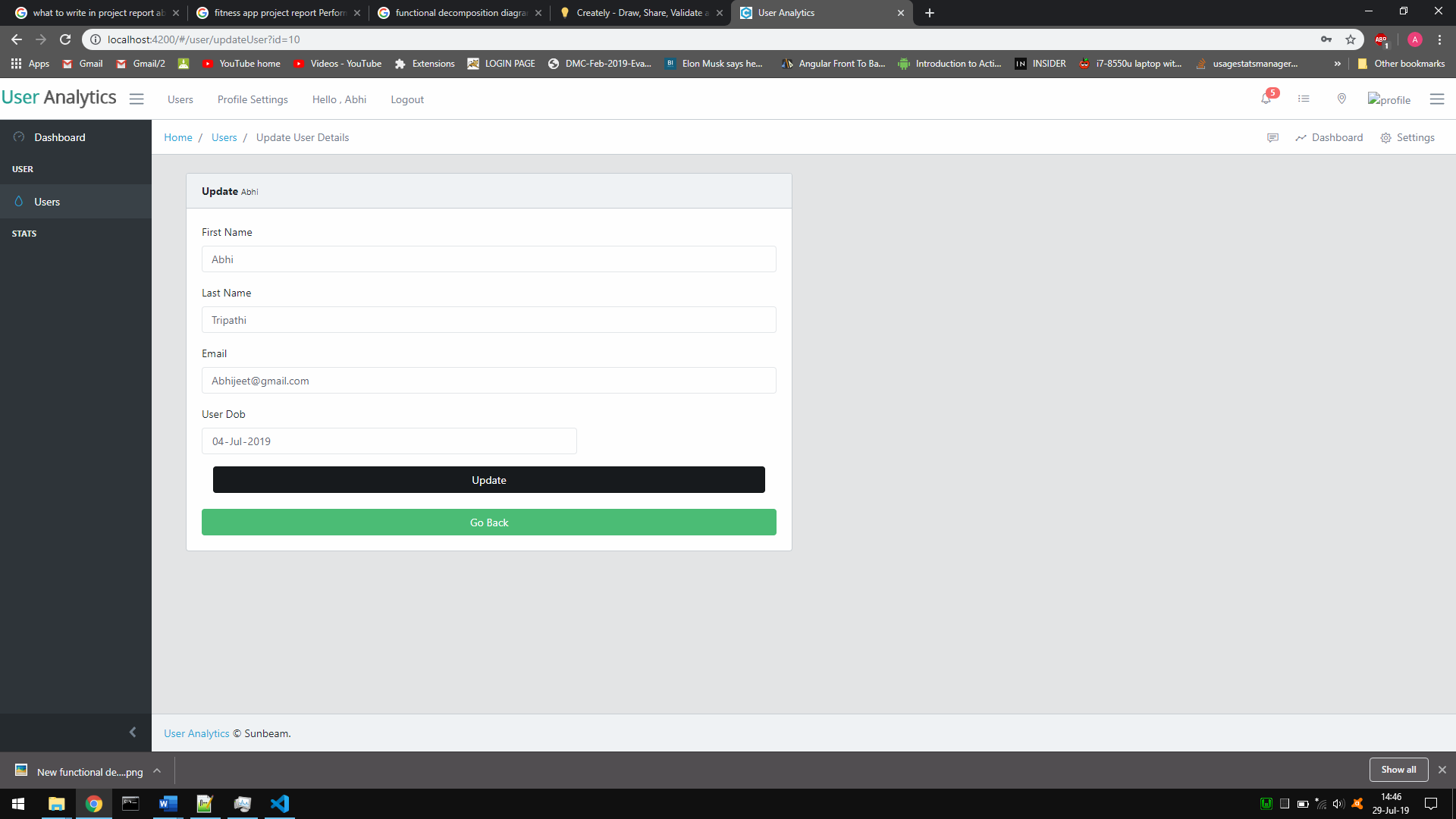


Fig. 7

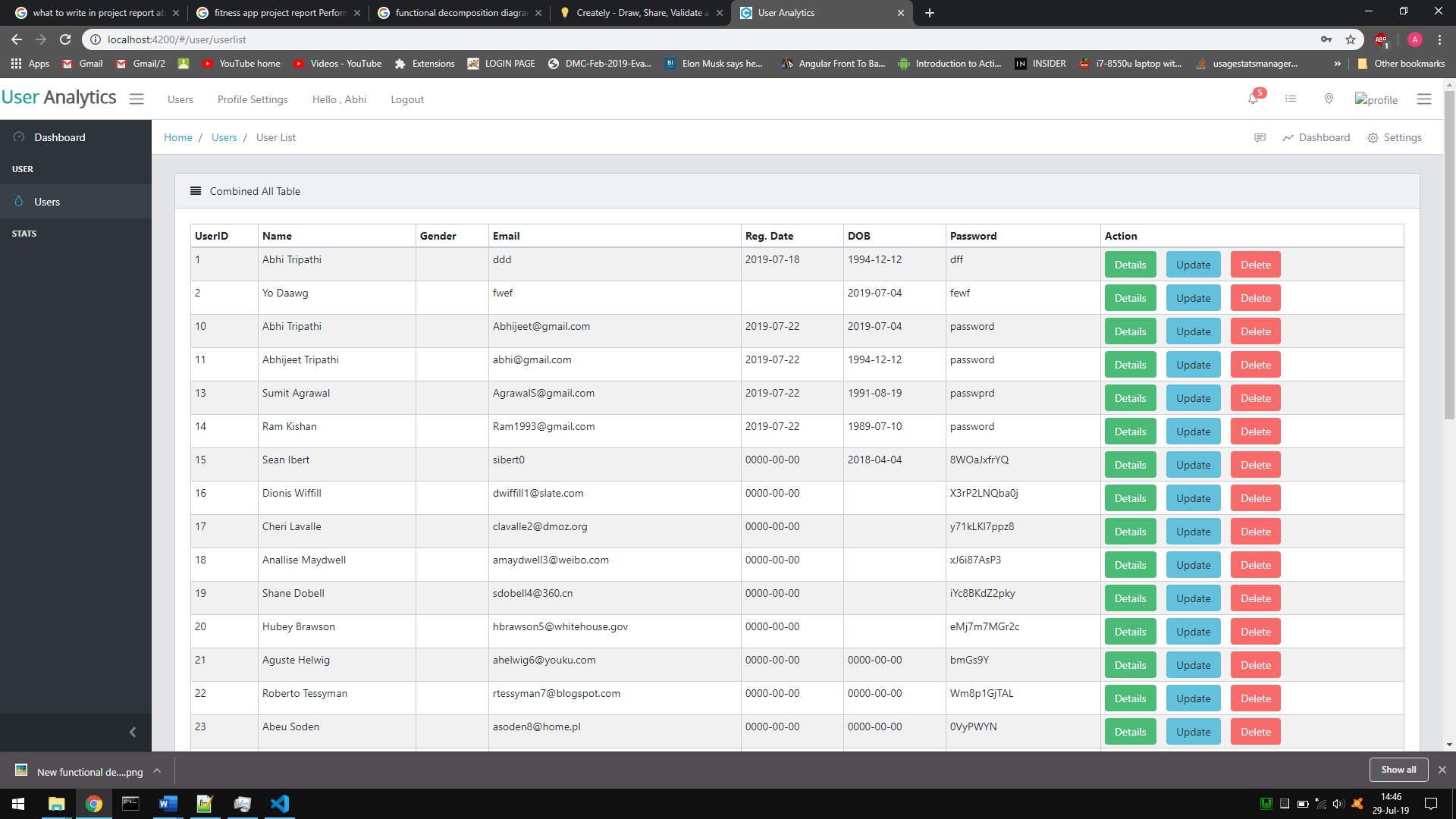


Fig. 8

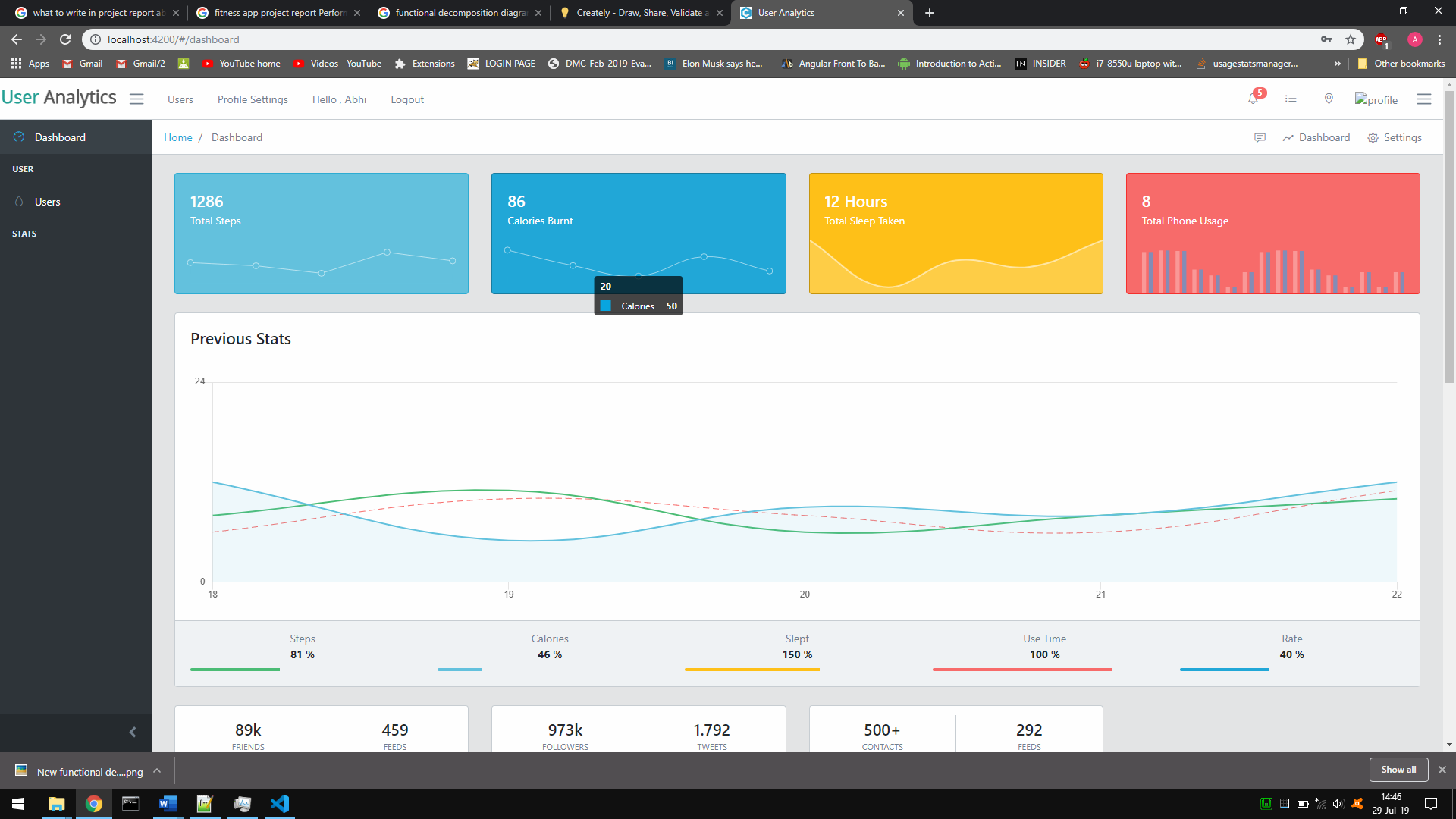


Fig. 9

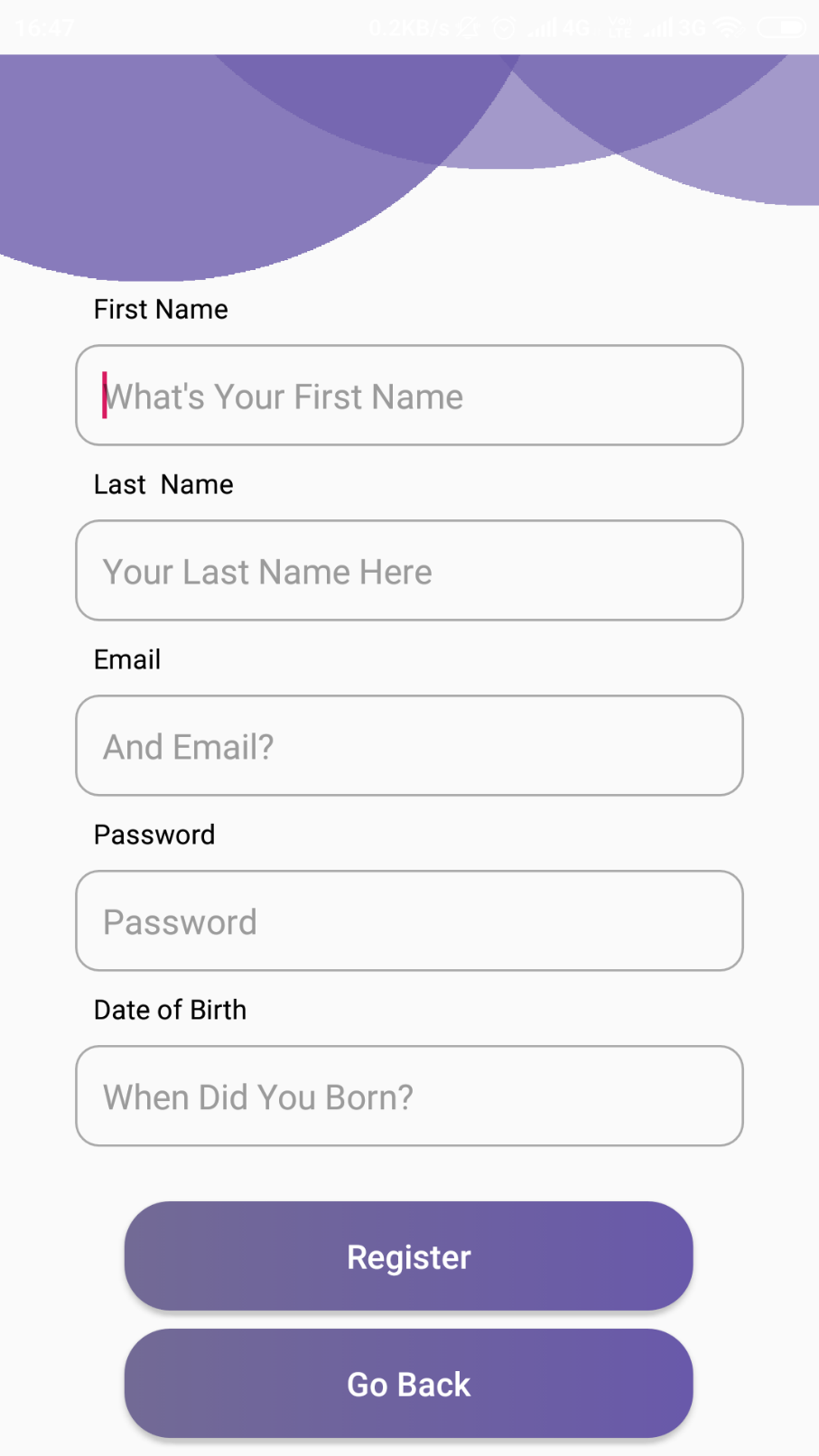


Fig. 10

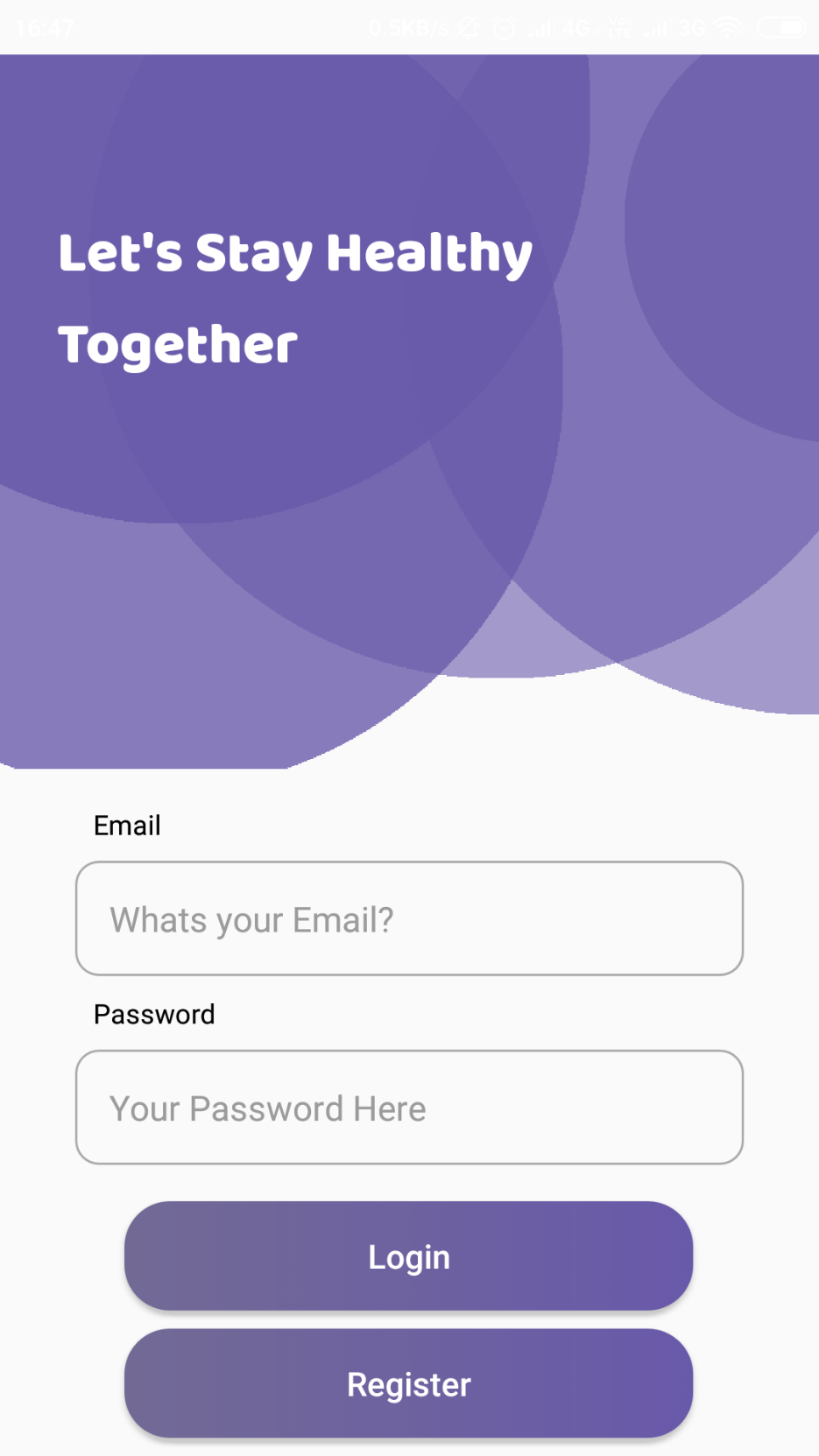


Fig. 11

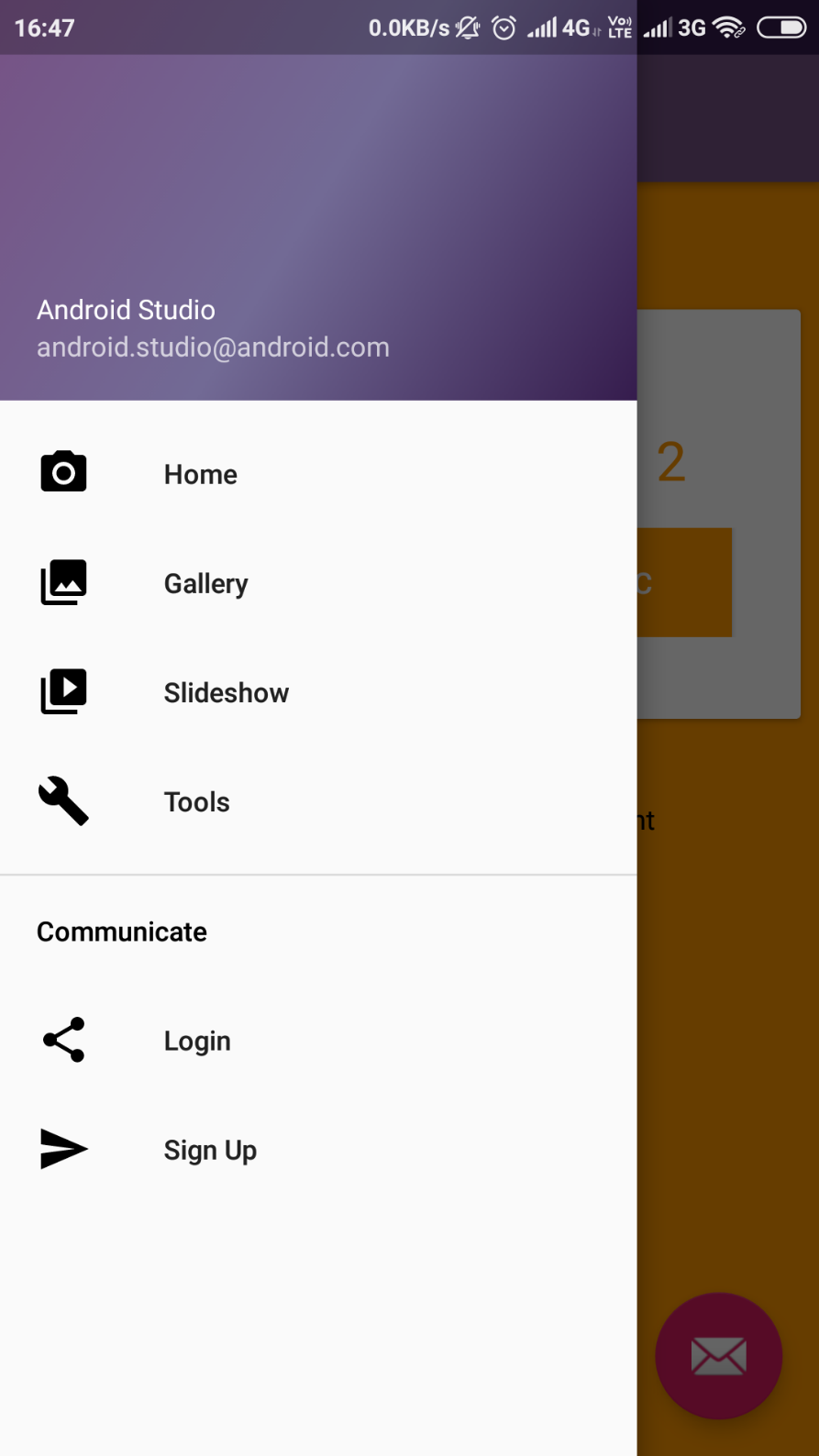


Fig. 12

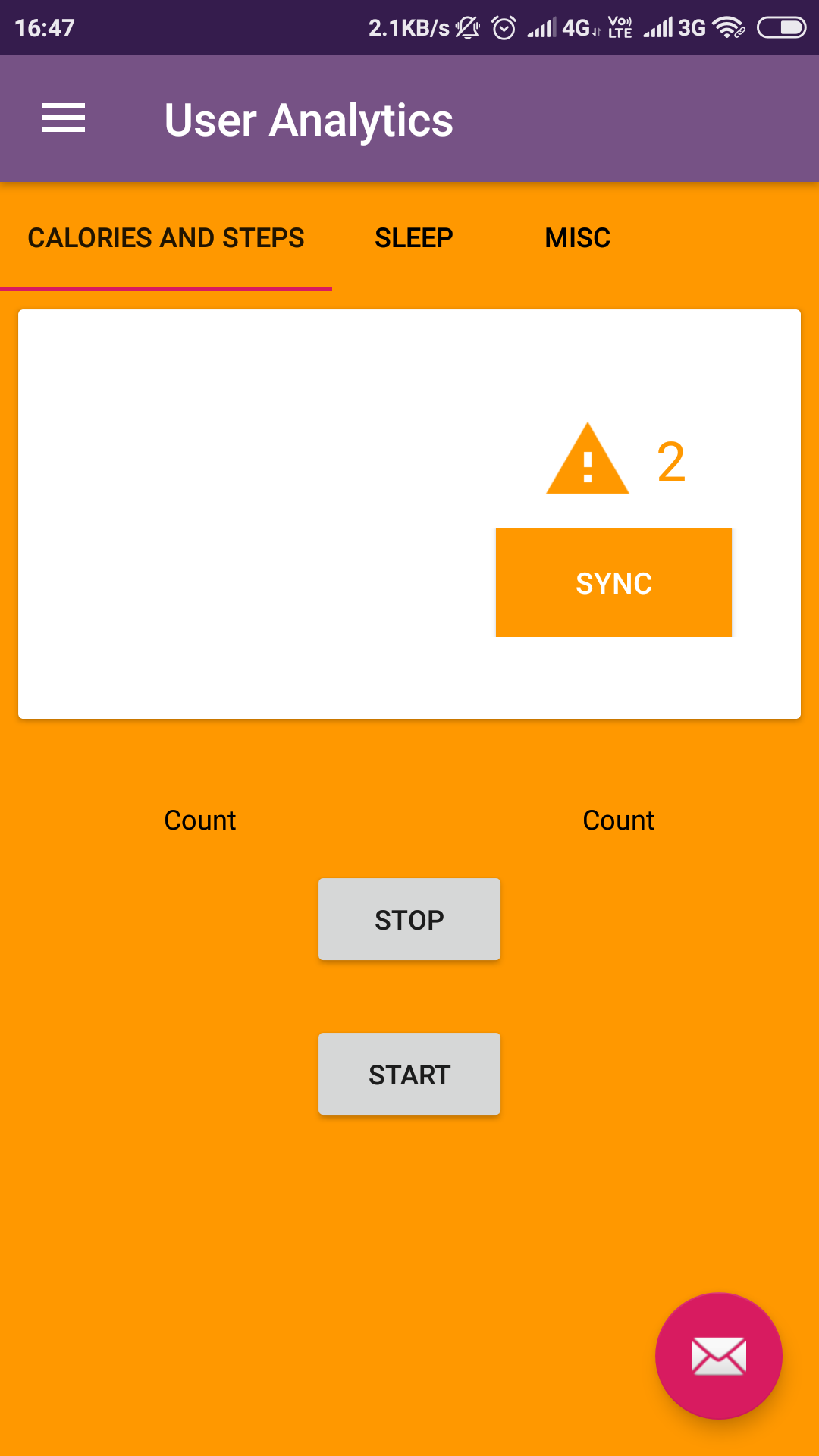


Fig. 13

**| Conclusion:**  Accomplishments While it was a challenge to develop, our project team successfully created a prototype wellness application for the Android platform capable of tracking, recording, and displaying data relevant to a user’s sleep, activity, and mood habits. While the full scope of the initial app design was not realized, all of the core data tracking functionality has been successfully implemented

**| Future Work** : While I feel that I’ve successfully implemented the desired tracking functionalities, there are still several elements that could be added to improve the effectiveness of the app. Another possible improvement is to enhance the accuracy of our tracking algorithms. While we have already devoted a significant portion of our development time to writing and testing these algorithms, theoretically, they can always be improved. The sleep tracking algorithm would likely be the most in need of improvement, due to the difficulty of analyzing sleep simply with sound and light levels. In this case, a biometric device that could track and record pulse or movement, would contribute tremendously to the accuracy of the collected sleep data.

**| References:**

Time - Alarm Clock. Retrieved from https://play.google.com/store/apps/details?id=com.azumio.android.sleeptime&feature=search\_r esult#?t=W251bGwsMSwxLDEsImNvbS5henVtaW8uYW5kcm9pZC5zbGVlcHRpbWUiXQ Fitbit Inc. (n.d.). Fitbit. Retrieved from http://dev.fitbit.com/ Fragments. (n.d.). Retrieved March 23, 2015, from

http://developer.android.com/guide/components/fragments.html Gehring, J. (n.d.). Documentation. Retrieved March 27, 2015, from http://www.androidgraphview.org/documentation Google. (n.d.). E.

Automation and developer API - Sleep as Android. Retrieved from https://sites.google.com/site/sleepasandroid/doc/developer-api Lopresti, A. L., Hood, S. D., & Drummond, P. D. (2013). A review of lifestyle factors that contribute to important pathways associated with major depression: Diet, sleep, and exercise. Journal of Affective Disorders, 148, 12-27 Michaud, C. (n.d.). Pedometer Using Accelerometer Sensor. Retrieved March 26, 2015, from

http://nebomusic.net/androidlessons/Pedometer\_Project.pdf PELUSO MAM et al. Physical activity and mental health: the association between exercise and mood. CLINICS 60(1): 61-70, 2005. ProtoGeo. (n.d.). Moves. Retrieved from

https://play.google.com/store/apps/details?id=com.protogeo.moves&hl=en Recording Fitness Data. (n.d.). Retrieved March 23, 2015, from https://developers.google.com/fit/android/record Sleepiness:

Cognitive and Emotional Effects. (n.d.). Retrieved March 23, 2015, from http://www.webmd.com/sleep-disorders/excessive-sleepiness-10/emotions-cognitive?page=3 Sonstroem RJ, Morgan WP. Exercise and self-esteem rationale and model. Med Sci Sports Exerc 1989;21:329-37. ViewPager. (n.d.). Retrieved March 23, 2015, from

http://developer.android.com/reference/android/support/v4/view/ViewPager.html 40 Working with

the Fitness History. (n.d.). Retrieved March 23, 2015, from https://developers.google.com/fit/android/history Build software better, together. (n.d.). Retrieved

December 15, 2014, from https://github.com/ Tudor-Locke, C., & Bassett Jr, D. (n.d.). How Many Steps/Day Are Enough? Retrieved November 15, 2014, from

http://www.health.utah.edu/peak/docs/Tudor Locke Paper.pdf National Sleep Foundation

Recommends New Sleep Times. (n.d.). Retrieved February 27, 2015, from http://sleepfoundation.org/media-center/press-release/national-sleep-foundationrecommends-new-sleep-times