

COMP – 8117 – 1 – R – 2022S

Project Team – 4

**Tras.h2o**

===============================

Final Project Report

**Object:**

|  |
| --- |
| **Document** |
| Final Project Report – Tras.h2o |

**Submitted To:**

|  |
| --- |
| **Project director** |
| Prof. Dr. Usama Mir |

|  |
| --- |
| **Project Client** |
| Ms. Saroj Dayal |

**Submitted By:**

|  |
| --- |
| **Team Members** |
| |  |  | | --- | --- | | Ayush Kaushik Mistry | 110087260 | | Harsh Bharath Nimmakuri | 110088953 | | Karan Amitkumar Zaveri | 110062029 | | Marc Smith | 110087601 | | Parth Ashokbhai Patel | 110071969 | | Pranjali Mukeshkumar Prajapati | 110087663 | |

Table of Contents

1. Project Description 4
   1. Project Preface 4
   2. Project Purpose 4
   3. Project Scope 4
2. Market Study 4
   1. Existing applications 5
   2. Target Users 6
   3. Unique Features 6
3. Software Requirements Specifications 6
   1. Objective 6
   2. Deliverables 6
   3. Project Team 6
   4. Expected Timeline 6
   5. Functional Requirements 6
   6. Non-Functional Requirements 7
4. Flowchart 7
5. Tools and Technologies 8
6. Architecture 8
7. Design Pattern – Atomic 9
8. Project Planning 10
   1. Sprint 1 10
   2. Sprint 2 10
   3. Sprint 3 10
   4. Sprint 4 10
   5. Sprint 5 11
9. Use Case Diagram 11
10. Gantt Chart 12
11. Sequence Diagram 13
12. Testing 14
    1. Introduction 14
    2. Testing Scope 14
    3. Out of Scope 14
    4. Testing Objective 14
    5. Roles and Responsibilities 14
    6. Test Methodologies 14
    7. Test Levels 14
    8. Suspension Criteria and Resumption requirements 15
    9. Test Completeness 15
    10. Testing resources 15
13. Cost Analysis 16
14. Risk Plan 17
15. User manual 18
16. Initial Expectations vs Final Product 18
17. Pros and Cons of the app 19
18. Difficulties Faced 19
19. APP UI 20
20. Future Scope 22
21. References 23
22. List of Figures 23
23. **PROJECT DESCRIPTION**
    1. **Project Preface**

The project mainly focuses on two problems. Firstly, people go to nearby coffee / smoothie vendors for their daily refreshments (coffee/health drink etc.). In most cases, people don’t carry their own cups but prefer the coffee/smoothie vendor to serve with a new one. Once done with it, they throw away the cup (with straw/lid sometimes). This results in thousands of cups being dumped every day.

Secondly, water is the most basic form of healthy/refreshing liquid item. Most people carry their own water bottle or sometimes buy a new plastic bottle in the stores to quench their thirst of water.

To tackle the above problems in a modern way, development of an app is proposed. On every user sign-in, the app enables the users to log the data on the cup they purchased. The log typically includes basic identifiers such as category of cup, purchased location, shop at which the purchases were made, quantity of the order etc. Along with this, the app also showcases the nearest water refill / fountain points in the nearby

* 1. **Project Purpose**

With the accumulated user data, the app will generate useful insights in the form of reports/ graphs on the quantity of waste saved that would have otherwise been dumped. In addition, the app guides users to the nearby water refill points which in turn can reduce the amount of water bottles purchased and thrown away. Theoretically, the reports generated should give a deep understanding of the scale of garbage generated at ground level, with respect to cups / water bottles only.

* 1. **Project Scope**

The Project is developed as a mobile application. Thus, in this modern age with unlimited number of mobile devices, the scale of this project is huge. It’s recommended for the devices having Android 7 or higher

1. **MARKET STUDY**

*“*It is estimated that Canadians consume 14 billion cups of coffee every year, and of that 30 per cent is consumed on the go. The coffee industry has responded to North America’s ‘*grab-and-go*’ culture by not only selling coffee, but also providing single-use, disposable cup for convenience*”* [1]. The number goes way beyond 20 billion disposable items with the inclusion of health refreshments, water bottles etc. Over half a million trees are lost every year due to disposable coffee cups. There is no existing application which tracks the details of cups being disposed daily and present the numbers to the people on a weekly/monthly basis. Most of the scenarios, they are represented in annual statistics. On the other side, most people prefer water as a healthy alternative, still most people buy plastic water bottles from supermarkets or vendor machines when needed. For people carrying a water bottle finding a refill point is difficult if it’s a new locality

* 1. **Existing Applications**
* **Loop** [2]

An application stationed in UK. It’s mainly purposed for collecting and ordering in reusable cups (*Loop cup*). When a cup is deposited in a Loop Return Point, the app deposits the order amount used to the user.

# Pros: Returns the ordered amount to the user,

Helps in reducing waste

# Cons: Only supported by McDonalds in UK,

Need Physical Machines to collect and process *Loop cups*

# Proposed idea features / advantages:

Shows the actual amount of waste being generated to the user,

Doesn’t store any financial data of user,

Doesn’t need any physical entities for functioning

* **Debris Tracker** [3]

An application operated by Morgan Stanley, to tackle the global plastic pollution. The user reports the various litter in different localities in the app. It is then used by scientists and researchers to understand the plastic pollution crisis.

# Pros: User can record various kinds of litter like plastic, metal, cloth etc.

National Geographic works on the data generated

# Cons: In depth statistics are not shown to the user

It’s a complete voluntary action by user with no rewards

# Proposed idea features / advantages:

The reports are shared with users along with several insights

Users are rewarded in form of health coupons to redeem

* **Publiek Water**[4]

A website which shows the water refill points mainly across Netherlands

# Pros: Wide range of coverage of water refill points in Netherlands

User community to add in new points

# Cons: Local app i.e., limited to Netherlands

# Proposed idea features / advantages:

Shows Water refill points in University of Windsor

Maintains individual user profile for each refill

* 1. **Target Users**

This is application is developed with minimalistic UI, making it effortless to use for all age groups. Recommended age groups would be from >=12

* 1. **Unique Features**
* The app has a user profile to track the user activities along with weekly leaderboards.
* Has a redeem coupon system to reward users
* No storage of user-sensitive information
* Refill functionality located mainly for University of Windsor

1. **SOFTWARE REQUIREMENTS SPECIFICATION (SRS)**
   1. **Objective**

To create a mobile application to satisfy the requirements stated in the project preface and comprising of the functional and non-functional requirements as stated in the Initial Software Requirement Specification.

* 1. **Deliverables**

- Working mobile application according to specification

- Application source code

- Final report

- User manual / Additional Documents

* 1. **Project Team**

The project team comprises of 6 members

* 1. **Timeline**

The project is on rails from May 22, 2022, to July 24, 2022

* 1. **Functional Requirements**
* **User Login/Sign-up:** The user should be able to login/signup in the app to use application or track his profile
* **Authentication:** Initial signup requires the app to authenticate the user with their email
* **Log data:** User can log the data of the trash in the app. The logging will be categorized into several fields like type, quantity, brand, location etc.
* **View Reports:** Weekly reports will be generated with the data collected along with user leaderboard to involve more participation
* **Redeem Coupons:** upon number of target entries, the user will be provided with coupons to be redeemable
* **Refill points:** The app shows the available water refill points in the locality of University of Windsor
* **User profile:** the app provides the user with a profile to track their log entries / refill meter and along with their coupons
  1. **Non - Functional Requirements**
* **Security Requirements**

No data, username or passwords will be stored in the application. The app should use an industry standard data exchange mechanism for exchanging usernames and passwords between the frontend and backend

* **Performance Requirements**

The application is required to run on basic model mobile devices available in today’s market, with minimum RAM and CPU

* **Software Quality Requirements**

**Testability:** Each feature must pass all test cases in order to be released as part of the production release

**Reliability:** The application must be launch, open and function correctly when needed

**Maintainability:**

The application must be designed and developed with best practices to ensure maintenance and support with regards to bug fixes and new feature requests

**Correctness:** The tally totals in the application must always produce the correct calculations

1. **Diagram

   Description automatically generatedFLOWCHART**

Figure 1: flowchart diagram

Above Figure 1 illustrates the flow chart representation of the working of the app ‘Tras.h2o’

1. **TOOLS AND TECHNOLOGIES**

* **Jira**

It will be used to manage and track the progress of the project, as well as the backlogs sprints, identified defects and change requests

* **Communication Platforms -** Microsoft teams and WhatsApp
* **Version Control -** Git and GitHub
* **React Native**

It’s used for client application development. Among many other mobile application development technologies, this is chosen because of higher chance of achieving project requirements with minimum learning curve

* **Django**

It is used as development tool for the backend. It provides much needed flexibility with respect to developing APIs and is ideal for rapid application development, given the short duration of timeline

Table 1: List of tools and technologies used

|  |  |
| --- | --- |
| Technologies | Tools / Languages |
| Programming | Python, Javascript |
| Framework | React Native |
| Activity monitoring Tools | JIRA, GitHub |
| Mobile Platform | Android |
| Database | PostgreSQL |

1. Diagram

   Description automatically generated**ARCHITECTURE**

Figure 2: Architecture

The above Figure 2 illustrates that the Tras.h2o application will utilize a client server architecture, with a frontend application communicating to the backend via the use of API

1. **DESIGN PATTERN - ATOMIC**

The application is developed with the help of React native on the front end. To maintain the growing complexity of the component structures, the team followed the ‘atomic design pattern’

A picture containing graphical user interface

Description automatically generatedFigure 3: Atomic design pattern

The above Figure 3 illustrates the five distinct elements of atomic design[5] are atoms, molecules, organisms, templates, pages. They map incredibly to form a well component-based architecture.

* **Atoms**

They can be defined as the basic building blocks. In other words, they represent the most basic interface elements such as buttons, icons. They serve as the supporting structures. Example of buttons, icons, text fields

* **Molecules**

Grouping atoms together to perform a task. An icon and a text field can be grouped together to form a user input field

* **Organisms**

They are a complex group of molecules, atoms or even other organisms. It can perform multiple tasks because of its complexity. They are represented for a single purpose. Example of a navigation bar

* **Templates**

They represent relation between the organisms. The clients can see the final design in place.

* **Pages**

They render different templates. They populate templates with unique contents. Example of a home page customized for different users

The main purpose is to segregate the contexts of each feature component, the atomic design pattern should be used to organize a React file structure. The isolation of side effects makes programming far more understandable and modular. Testing will be simpler with a single instance of a feature, which will enhance your application's overall quality assurance. The organization of your file structure in this pattern will help you simply determine and handle state as the complexity of your application and state management increases.

1. **PROJECT PLANNING**

In this project, software development with Agile framework is practiced. Scrum methodology [6] is implemented because of the ‘sprint’ planning technique which enables incremental feedback at each sprint. The total project is split into 5 sprints

Timeline

Description automatically generatedFigure 4: Agile Scrum Framework

The above Figure 4 illustrates the components involved in a scrum framework and their descriptions

Table 2: The project sprint timeline is as follows

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Sprint** | Sprint 1 | Sprint 2 | Sprint 3 | Sprint 4 | Sprint 5 |
| **Start date** | 2022-05-20 | 2022-05-30 | 2022-06-13 | 2022-06-27 | 2022-07-11 |
| **End date** | 2022-05-29 | 2022-06-12 | 2022-06-26 | 2022-07-10 | 2022-07-24 |

* 1. **Sprint 1**
* Project Idea Initiation
* Project Proposal Submission
  1. **Sprint 2**
* **Provisional Planning:**
  + Initiate User login functionality
  + Initiate Rewards functionality
  + Setting up docker container
* **Effective Planning:**
  + Initiate User Login / sign up functionality
  + Initiate User Log data functionality
  + Setting up docker container
  1. **Sprint 3**
* **Provisional Planning:**
  + Initiate User Profile functionality
  + Initiate Rewards functionality
  + Define the Django models
* **Effective Planning:**
  + Working on the Log data functionality
  + Initiated the User profile functionality
  + Django models specified
  1. **Sprint 4**
* **Provisional Planning:**
  + Initiate leaderboard functionality
  + Initiate Refill functionality
  + Define the urls and views
* **Effective Planning:**
  + Working on the Log data functionality
  + Initiated the Leaderboard functionality
  + Initiated the User profile functionality
  + Defined the urls and views
  + Testing on the developed modules
  1. **Sprint 5**
* **Provisional Planning:**
  + Initiate Report functionality
  + Testing on the functionalities
  + Prepare for the final presentation
  + Final Report
* **Effective Planning:**
  + Working on the Log data functionality
  + Working on the Leaderboard and refill functionality
  + Final Report preparation
  + Testing - manual and automation

1. **USE CASE DIAGRAM**

Diagram, schematic

Description automatically generated

Figure 5: Use Case Diagram

The above Figure 5 illustrates the use case diagram of the app with the main components as the user, admin and the system

1. **A picture containing chart

   Description automatically generatedGANTT CHART**

Figure 6: Gantt Chart

The above Figure 6 illustrates the gantt chart, which represents against the tasks performed during the project tenure along with the dates.

1. Diagram, schematic

   Description automatically generated**SEQUENCE DIAGRAM**

Figure 7: Sequence Diagram

The above Figure 7 illustrates the sequence diagram, showcasing the sequence of interactions between the objects in the Tras.h20 application.

1. **TESTING**
   1. **Introduction**

Tras.h2o is a mobile application developed in the concept of ‘Green Software Development’. The user creates boards onto the application and has the services to view the previous weeks reports or log the data or check the nearest water refill point. Each module is directed into its respective features. Reports show the previous weeks reports, Log data enables user to perform multiple input operations and nearest refill locations offers the list of nearest refill locations available to the user.

* 1. **Testing Scope**
* user login / signup
* ability to choose from the menu
* log data
* view reports
* Check nearest refill points
* Check user profiles
  1. **Out of Scope**
* Mobile hardware related issues
* Device operability system combability
  1. **Testing Objective**
* Making sure the deployed product is of high quality
* Application is free of bugs and errors
* Provide a functioning application to the user
* Making sure user has no difficulties in operating the application
  1. **Roles and Responsibilities**
* Quality Assurance Engineer(QAE): See to that the product quality is maintained and that the functionalities are tested, and defects are reported in accordance with the specifications
* Quality Assurance Manager: All the QAEs reports to the manager to keep a tab on the progress made
  1. **Test Methodology**
* Since the project is developed in agile framework (SCRUM), testing is also performed in agile methodology. Scrum is flexible in its sprint approach, so testing is performed in the final sprints
  1. **Test levels**

The following level of testing is followed to maintain the quality of the product

* Unit Testing: It divides the source code into separate components. Each of these tiny parts has its own functionality, and each one is examined to determine whether it satisfies the requirements
* Integration Testing: It combines individual components and tests them together if the functionality as intended. It is always performed after unit testing
* System Testing: It acts as back-box testing. It tests the overall systems functionality and checks for all the functional requirements
* User Acceptance Testing: It is a testing performed by customers and users. This testing determines if the application has met the business needs
  1. **Suspension Criteria and Resumption requirements**
* Suspension criteria
* The Shipped features are not working as expected
* The app is not opening
* The application crashes
* Application responsiveness is not as expected
* Resumption Requirements
* The features shipped are working as expected
* The app is opening
* Application didn’t crash
* Responsiveness of the application meets the quality standards
  1. **Test completeness**

Criterion resembling the test completeness of the system:

* Manual test cases
* Features testing
* Testing updates post defects
* Testing covering every possible input/user action
  1. **Testing resources**

Performed an extensive manual testing because of the knowledge gap in performing automated testing. Used an online industry certified tool ‘MobSF’ to static test the final ‘.apk’

* **MobSF**

Graphical user interface, application

Description automatically generatedIt is a static and dynamic analysis-capable, automated, all-in-one mobile application (Android/iOS/Windows) pen-testing, malware analysis, and security assessment framework. It is open source and is approved by OWASP (Open Web Application Security Project).

Figure 8: MobSF static report

The above Figure 8 illustrates the security certificate along with the app scores and information about the app provided by MobSF.

* **Jest**

A straight forward JavaScript testing framework is Jest. It is a framework for testing that is used to address problems in the codebase. Develop tests using this feature-rich API that are result-oriented, simple, and intuitive. It has good documentation, is extensible, and requires less setups.

Features of Jest:Text

Description automatically generated Easy Mocking, Code coverage, zero configuration, parallel tests, snapshots, Safe and Fast

Figure 9: Test cases output

The above Figure 9 illustrates the outputs of the automated test cases performed using jest.

1. **COST ANALYSIS**

The average salary of a software developer in Canada is $100,000 CAD per annum. So, for a single developer it will cost $8333 CAD per month. The average salary of a scrum master is $65000 CAD per annum, so per month a scrum master will cost $5416 CAD.

Table 3: cost analysis

|  |  |  |  |
| --- | --- | --- | --- |
| User\pay | Individual pay/month | Team pay/month | Total for 2.5mo |
| Developer \*3 | $8,333 | $24,999 | $62,498 |
| Scrum Master \*1 | $5,416 | $5,416 | $13,540 |
| QA/Tester \*2 | $5,000 | $10,000 | $25,000 |
| Total | $18,749 | $40,415 | $101,037 |

From the table 3, the cost of renting a server will be approximately $1,000 CAD per year. So, since we are a team of 6 members and will be granting approximately $101,037 CAD for the development of application

1. **RISK PLAN**

A risk is an unanticipated event or circumstance that, if it occurs, will have a substantial influence on the project's goals. The purpose of risk management plan is to identify and control all potential risks involved during the project’s initial phase to the final phase of the project. The following procedures make up the risk management life cycle

Diagram, schematic

Description automatically generated

Figure 10: Risk Management Life Cycle

The Figure 10 illustrates the steps involved in the life cycle process of risk management

* **Quantitative Risk Analysis:**

The technique of numerically analyzing the effect of identified risks on overall project objectives. A most common methos is the use of ‘Risk Matrix’

It is a matrix utilized for risk assessment to define the level of risk by considering the probability or likelihood against the consequence severity.

The harm severity can be categorized as catastrophic, critical, marginal, minor

The probability of the risk/harm occurring can be categorized as certain, likely, possible, unlikely, rare

Table 4: Risk Matrix

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Probability \ Severity | **Negligible** | **Marginal** | **Critical** | **Catastrophic** |
| **Rare** | High | High | Very high | Very high – 1 |
| **Unlikely** | Medium | High | High - 3 | Very high – 2 |
| **Possible** | Low | Medium - 4 | High - 7 | High |
| **Likely** | Low | Medium - 5 | Medium | High |
| **Certain** | Low - 8 | Low - 6 | Low | Medium |

From Table 4, the possible undesirable outcomes:

1. Product incomplete
2. Incomplete implementation of features
3. Product Rejection
4. Pricing
5. Feature malfunction
6. Login issues
7. Loss of user saved data
8. UI bugs

* **Risk Response Planning**

Risk Responses were planned as additional activities, time and budget reserves, or process adjustments. Other forms of risk responses should be established with the help of customers and the project manager. Each risk response plan should have a dedicated person who maybe a team member. They should be able to check on the risk and cooperate on the risk response implementation. This team member or the risk owner has the total duty for the risk. In case of any issues, the owner should raise it to the project manager

* **Risk Monitoring, Controlling and Reporting**

The team member or the risk owner is account for the following

* Keep a track on the assigned risk
* Updating on the progress of response implementation
* Updating any changes to the risks
* Identifying and entering any secondary risks.

1. **USER MANUAL**

The below steps to be followed for the usage of the application

Step – 1: Ensure the device requirements are satisfied

* Android Version – 7.0 or higher

Step–2: Install the application from the link: https://github.com/theonlysam/trash2o/blob/main/Frontend/ApkOutput/Trash20.apk

Step – 3: Open the app to find a User Login / SignUp menu

Step – 3.1: If a new user, proceed by clicking on ‘Sign-up’ and register to use the app

Step – 3.2: If existing user, Enter your username and password and login

Step – 4: On the main menu, choose the operation to be performed

Step – 4.1: If the choice is to log the item, click on ‘counter’

Step – 4.2: If the choice is to find a refill point, click on ‘find refill’

Step – 4.3: If the choice is to view report, click on ‘view reports’

Step – 5: Log the item data by entering various fields of brand, quantity, category

Step – 6: find the refill points by choosing the location given in the drop down

Step – 7: Find the latest reports in the view reports page

Step – 8: User profile section can be viewed/managed by clicking on the user profile icon

1. **INITIAL SOFTWARE REQUIREMENTS vs FINAL PRODUCT**

**Initial software requirements**

* An application with feature to login / signup
* The application must be cross platformed i.e. iOS and Android
* The app must be able take input from user regarding different items disposed in the categories of cups, brand, quantity, and date
* The app must show the nearest refill location with description, fountain, and refill status
* The app must have a report page which comprises various statistics of the data collected over a period
* User must access a user profile
* Must be provided with coupons upon 10 each target entries

**Final Product Delivered**

* An application features functionality to login / signup
* The app takes input from user reg different items disposed in the categories of cups, brand, quantity, and date
* The app must show the nearest refill location with description, fountain, and refill status and redirects users using google maps limited to the University of Windsor
* The app has a report page which comprises various statistics of the data collected over a period
* Access to the user profile showing the various user metrics
* Feature to redeem coupons upon reaching certain number of points (100, 200..)

1. **PROS and CONS OF THE APP**

* **Pros**

User can track their liquid inputs and compare with other users.

The app doesn’t store any user financial sensitive data.

Helpful in finding the nearest water refill locations.

Promotes user to use water bottles and decrease usage of disposable cups.

* **Cons**

The app is only available on Android platform.

The app doesn’t provide notifications upon updates

No proper ways to validate the user input – High chances of fake data input

1. **DIFFICULTIES FACED**

* Most of the team members are new to the technology of react native and Django, so reskilling became difficult.
* Improper infrastructure and system outages led to difficult in tracking of activities
* Functionalities like ‘Log Data’, ‘Refill locations and iOS dependency took considerate amount of time to resolve, with the amount of bugs

1. Graphical user interface, application

   Description automatically generated**A screenshot of a phone

   Description automatically generated with medium confidenceGraphical user interface, application

   Description automatically generatedAPP UI**

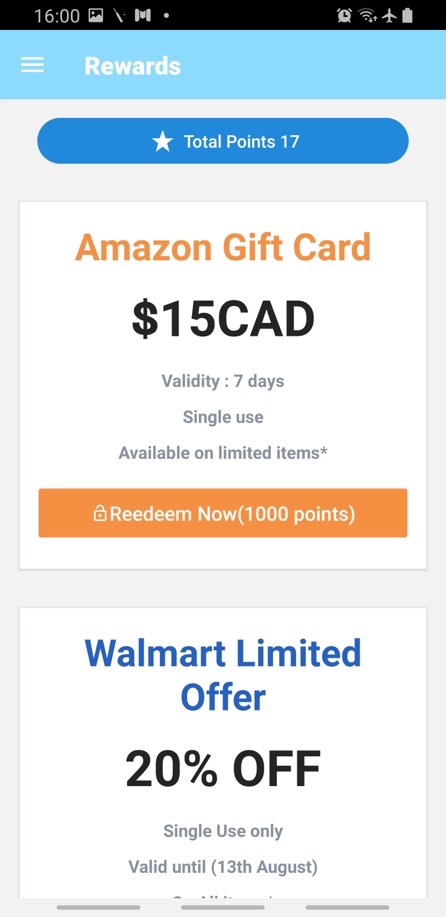
Figure11: Selecting container size Figure 12: Selecting brand Figure 13: Selecting Item

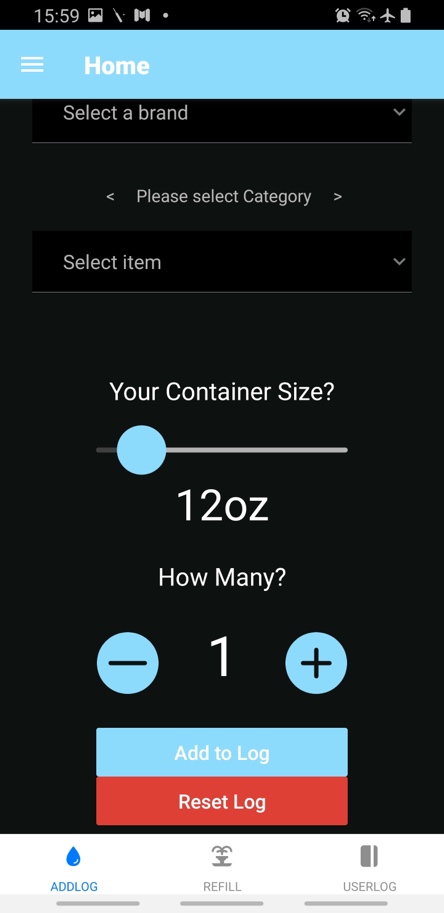
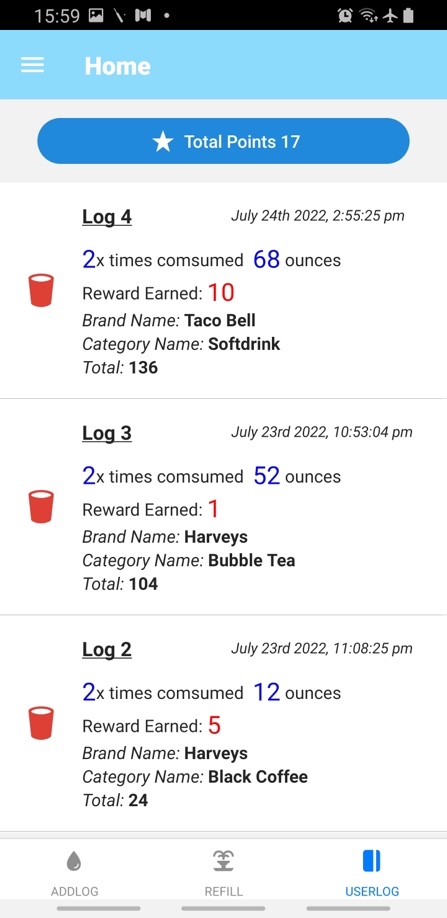
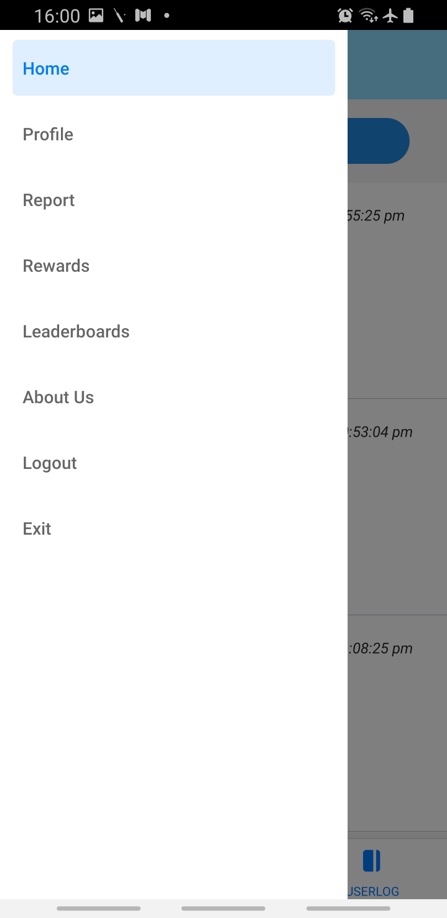
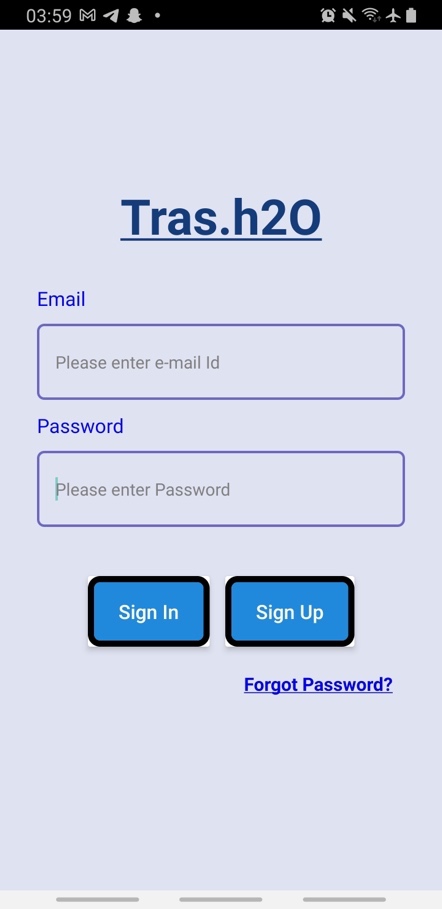
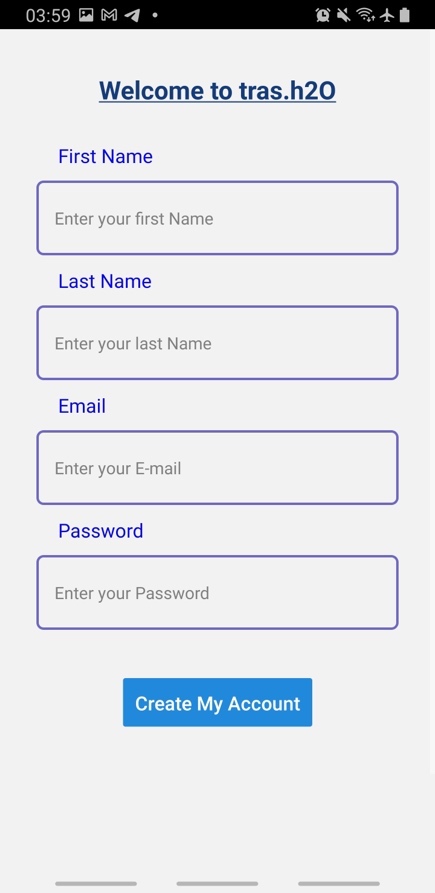
Figure 14: main menu Figure 15: user log Figure 16: app coupons 

Figure 17: Side menu Figure 18: Log in page Figure 19: Sign up page

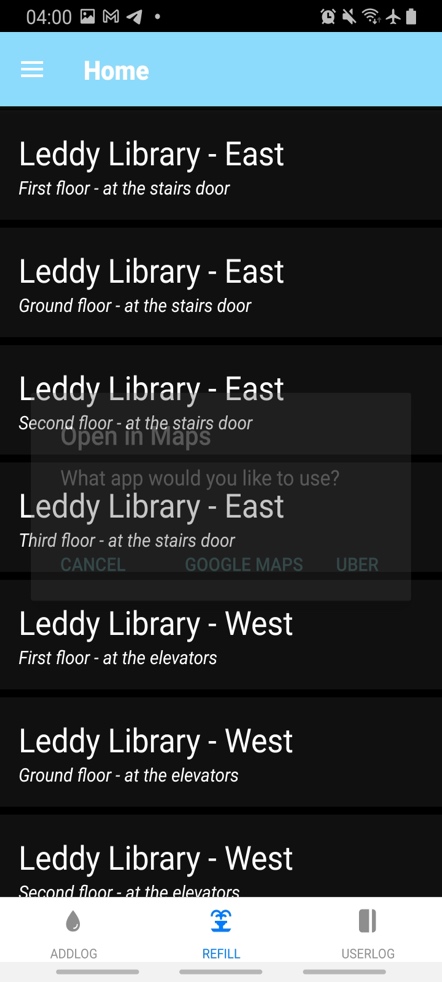


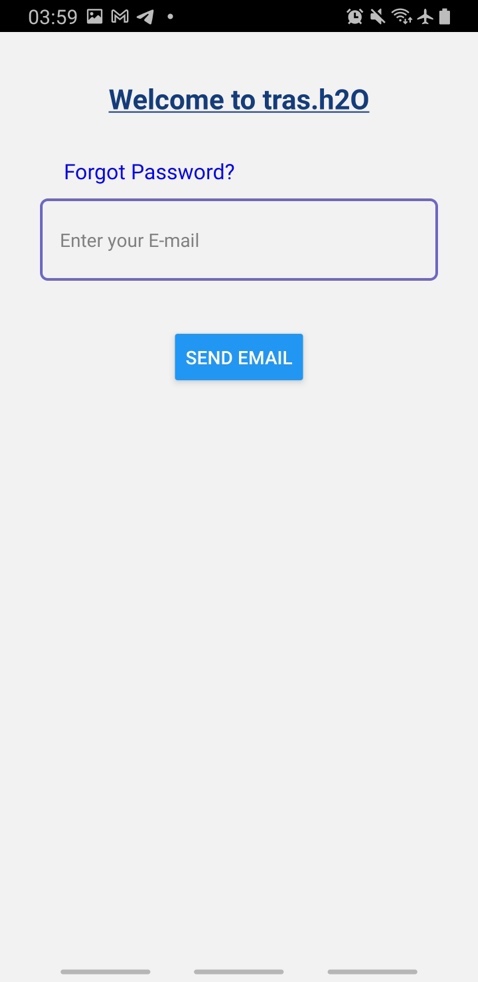
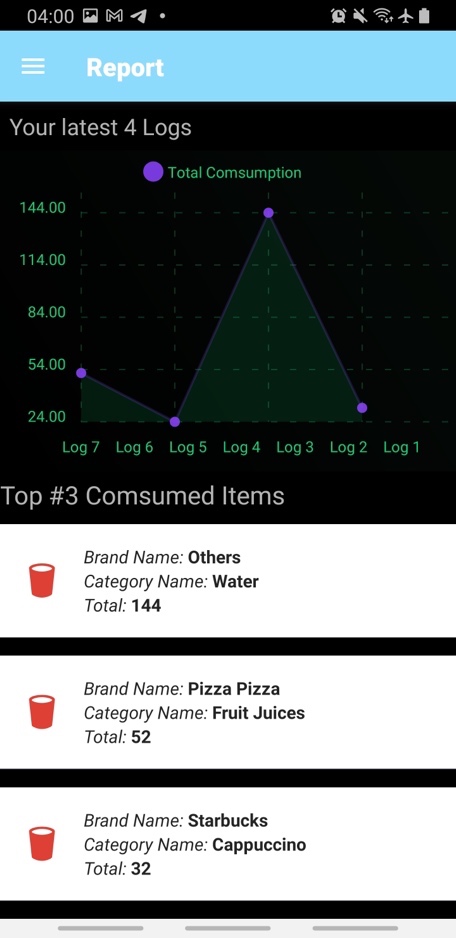
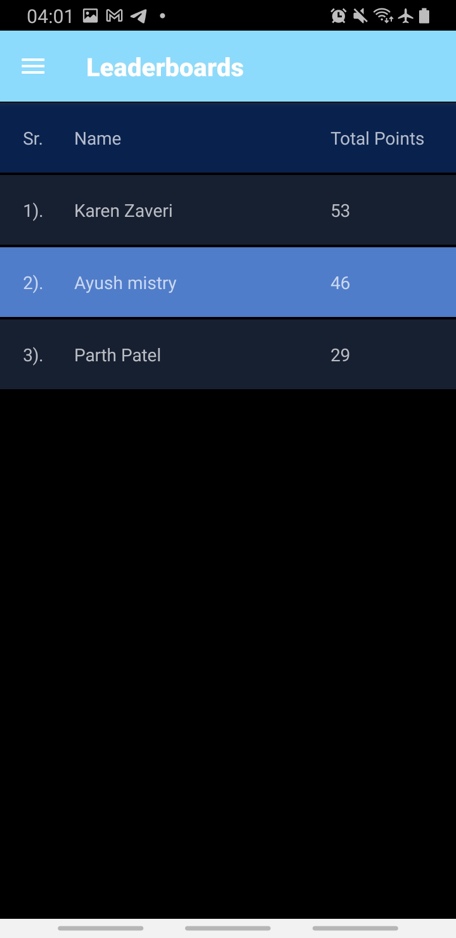
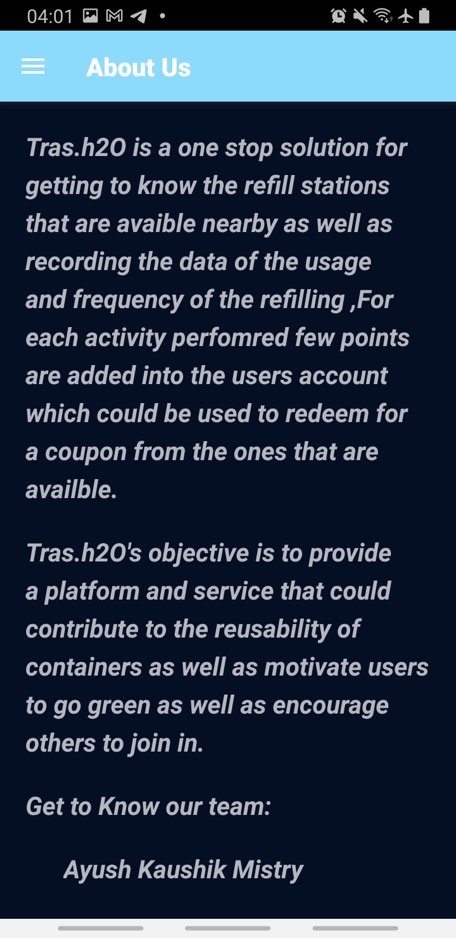
Figure 20: reset password Figure 21: refill locations Figure 22: User profile

Figure 23: Reports page Figure 24: Leaderboards Figure 25: About us page

The above figures 11-23, illustrates each screen display UI of the app along with their functionalities

1. **FUTURE SCOPE**

The application can be further improved / modified upon multiple points. Few are

* The user login / sign up can be extended from email only to social media accounts like Facebook, twitter, Twitch, Discord etc.
* The counter feature can be modified by adding location and comments features
* The refill location can be made mode dynamic by the guiding user to the location though location access
* A special link which enables users to share their work on social media platforms
* The App is specific to cups that are thrown away, this feature can be extended to different category of products like garbage, clothes or food and the app can be more reachable
* The coupons can be provided based on multiple tie ups with local vendors
* A live counter notification for total number of cups disposed and water refill used
* Making the app to function on the iOS platform
* Making the product more refined i.e., software and security updates

1. **REFERENCES**

[1] University of Manitoba October 2018, UM Today | Sustainability | Let’s Talk Waste: Disposable Cups, University of Manitoba, accessed 30May 2022, <https://www.student.unsw.edu.au/how-do-i-cite-electronic-sources>

[2] Loop Retailers, accessed 30 May 2022, <https://exploreloop.com/McDonalds>

[3] Debris Tracker | National Geographic Society, National Geographic, accessed 1 June 2022, <https://www.nationalgeographic.org/education/programs/debris-tracker/>

[4] Entrepreneur? Make your tap available! - Public Water, accessed 1 June 2022, <https://publiekwater-nl.translate.goog/?_x_tr_sl=auto&_x_tr_tl=en&_x_tr_hl=en&_x_tr_pto=wapp>

[5] Atomic Structure in React Native, Sandhya Deshmukh, accessed on 23 Jun 2022, <https://blog.consisty.com/react-native/atomic_structure/>

[6] Scrum, Claire Drumond, accessed on 19 Jun 2022, <https://www.atlassian.com/agile/scrum>

1. **LIST OF FIGURES**

|  |  |
| --- | --- |
| **Description** | **Page no** |
| Figure 1 : Flowchart | 7 |
| Figure 2 : Architecture | 8 |
| Figure 3: Atomic Design Pattern | 9 |
| Figure 4: Agile Scrum Framework | 10 |
| Figure 5: Use Case Diagram | 11 |
| Figure 6: Gantt Chart | 12 |
| Figure 7: Sequence Diagram | 13 |
| Figure 8: MobSF Static report | 15 |
| Figure 9: Test cases output | 16 |
| Figure 10: Risk Management Life Cycle | 19 |
| Figure 11: Selecting container size | 20 |
| Figure 12: Selecting brand | 20 |
| Figure 13: Selecting Item | 20 |
| Figure 14: main menu | 20 |
| Figure 15: user log | 20 |
| Figure 16: app coupons | 20 |
| Figure 17: Side menu | 21 |
| Figure 18: Log in page | 21 |
| Figure 19: Sign up page | 21 |
| Figure 20: reset password | 21 |
| Figure 21: refill locations | 21 |
| Figure 22: User profile | 21 |
| Figure 23: Reports page | 22 |
| Figure 24: Leaderboards | 22 |
| Figure 25: About us page | 22 |