



CZ4079 Final Year Project

JUSTGIVIT

A digital platform for giving

Project Index

SCSE20-0175

Supervisor

Dr Own Noel Newton Fernando

Supervisee

Sam Jian Shen (U1821296L)

**School of Computer Science and Engineering
Nanyang Technological University**

| Version Control | | | |
|-----------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-------------------|
| Version # | Changes | Reason | Date (DD/MM/YYYY) |
| 1 | 1. Added cover page 2. Added draft content page 3. Added initial research materials and citing | 1. Initialise documentation format 2. Draft out possible content for the final year project report | 21/08/2020 |
| 2 | 1. Added initial information about background, objective, scope, expected tools, draft roadmaps 2. Added Appendix | 1. Formulate Initial Ideas, Concepts, and Plans 2. Draft out potential extra content key points | 22/08/2020 |
| 3 | 1. Added Motivation Content | 1. Provide stronger reasoning for the needs of this problem locally. | 24/08/2020 |
| 4 | 1. Updated and solidified background, objective, and scope 2. Added Project Schedule from draft Project Planner and convert them into RoadMap and Gantt Chart 3. Added Initial Concept Figure | 1. Update information base on feedback from Supervisor 2. Established a plan for the project for the whole duration. 3. Added initial visual concept of the project | 31/08/2020 |
| 5 | 1. Added Meeting Minutes | 1. For self-reference and keeping track of the meeting details. | 02/09/2020 |
| 6 | 1. Added Elicitation requirement (Functional Requirement, Use Case diagram) 2. Non-Functional requirement | 1. Formulate functions for the application | 15/09/2020 |
| 7 | 1. Added Elicitation requirement (Use Case description) | 1. Formulate the behavior between the users and system | 16/09/2020 |
| 7.1 | 1. Vetting, refactor and improve the structure, improve clarity. | - | 17/09/2020 |
| 8 | 1. Added LO-FI v1.0 design 2. Added Draft Design Architecture 3. Added Work Breakdown Structure and Feedback Channel 4. Added concepts and technical knowledge | 1. Formulate Initial Design Interfaces. 2. Formulate the general design of data flow and behavior in the system 3. This helps ensure work progress and concerns are tracked easily 4. Adaptation of certain architecture, software, style, etc | 02/10/2020 |
| 9 | 1. Added Technical description 2. Added Setup Information | 1. Shared the experience regarding the software and tools I used in this project. 2. References for the target audience | 03/10/2020 |
| 10 | 1. Added HI-Fi Design 2. Added Database design | 1. Develop a better precision UI layout to be implemented | 20/10/2020 |

| | | | |
|-------------|--------------------------------------------------------------------------------------------------------------------------|------------------------------------------------------------------------------------------------|------------|
| | | 2. Develop the objects needed for the database | |
| 10.1 | 1. Vetting, refactor and improve the structure, improve clarity. | - | 21/10/2020 |
| 11 | 1. Added Assumptions, Constraints, and Project Environment Content 2. Added Data Dictionary 3. Added System Design | 1. Ensure all facts are listed needed for this project 2. Ensure there is minimal ambiguity | 11/11/2020 |
| 12 | 1. Added UI (Web) 2. Finalise report structure for an interim report | 1. Display UI features | 20/01/2021 |
| 12.1 | 1. Vetting, refactor and improve the structure, improve clarity. | - | 21/01/2021 |
| 13 | 1. Added UI (Mobile) | 1. Display UI features | 20/02/2021 |
| 14 | 1. Added Test Case | 1. To ensure good quality software | 09/03/2021 |
| 15 | 1. Added What's Next Content | 1. Ideas to add-on or improve on this project | 11/03/2021 |
| 16 | 1. Added Challenges 2. Added Conclusion | 1. Sharing challenging experience of this project 2. Wrap up about this project | 15/03/2021 |
| 17 | 1. Added Abstract and Acknowledge key points | 1. Acknowledgement and overall information about this project. | 16/03/2021 |
| 18 | 1. Added Result | 1. Showcase workable function. | 13/04/2021 |
| 18.1 | 1. Vetting, refactor and improve the structure, improve clarity. | - | 14/04/2021 |
| 19 | Final Wrap Up | 1. Ready for submission! | 16/04/2021 |

Legend

[X]: Enclosed square bracket refers to citation or supported claims references link to the source where X represents a whole number.

{X}: Enclosed curly bracket refers to image link or any entity link that retrieved over the internet sources where X represents a whole number.

Contents

| | |
|----------------------------------|----|
| Abstract..... | 7 |
| Acknowledgment | 8 |
| Introduction | 9 |
| Background | 9 |
| Motivation..... | 10 |
| Objective..... | 15 |
| Scope..... | 15 |
| Project Schedule | 16 |
| Roadmap..... | 16 |
| Gantt Chart | 16 |
| Project Track | 20 |
| Work Breakdown Structure | 20 |
| Assumptions..... | 21 |
| Constraints..... | 21 |
| Project Environment..... | 22 |
| Internal..... | 22 |
| Software..... | 22 |
| Hardware | 22 |
| External | 23 |
| Technical Specification..... | 24 |
| Technical Description..... | 24 |
| Experiences | 24 |
| Data Dictionary | 25 |
| Functional Requirement | 26 |
| Cross-Platform | 26 |
| Web platform exclusively..... | 29 |
| Mobile platform exclusively..... | 30 |
| Non-Functional Requirement | 30 |
| Web and Android | 30 |
| Use Case Diagram | 31 |
| Use Case Description | 32 |
| Design Specification | 53 |

| | |
|---------------------------------------|-----|
| System Design..... | 53 |
| System-Level | 53 |
| Software-Level | 59 |
| Database Design..... | 60 |
| UI Design..... | 62 |
| Consideration..... | 62 |
| Tools Used..... | 65 |
| Initial Concept..... | 66 |
| LO-FI | 67 |
| Hi-FI | 68 |
| Implementation Specification..... | 71 |
| Foundation Phase Implementation | 71 |
| System Design Implementation..... | 73 |
| System-Level | 73 |
| Software-Level | 74 |
| Database Design Implementation | 75 |
| Directory Organisation..... | 76 |
| UI Design Implementation..... | 79 |
| Public UI | 79 |
| Normal UI..... | 84 |
| Company UI..... | 97 |
| Administrator UI (Web Only) | 98 |
| Test Cases..... | 102 |
| Walkthrough | 105 |
| What's Next?..... | 118 |
| Challenges..... | 119 |
| Conclusion..... | 120 |
| Appendix | 121 |
| Disclaimer..... | 121 |
| Credits | 121 |
| Access guide..... | 122 |
| Source code guide..... | 123 |
| Test guide..... | 128 |

| | |
|-------------------|-----|
| Mobile UI..... | 129 |
| Public User | 129 |
| Normal User | 132 |
| Links | 139 |
| References | 140 |

Abstract

Waste has always been a prominent problem to solve with the limited land of Singapore. There has been plenty of existing state-of-the-art solutions that can aid in solving this problem, but it was never enough. This motivates the development of this software application called “JustGivit: A digital platform for giving”. This application helps further contribute to reducing the waste through indirect means such as donating and exchanging goods instead of disposing them directly into the trash bin. For donating, it digitizes charity operations in terms of donating unwanted and reusable goods locally in a more efficient manner and offers to reach out to wider target audiences. For exchange, it has helped reach out to those people who need them and promote reusability. This report will focus on evaluating the application’s key software development life cycle (SDLC) content from design to release of production processes and experiences.

Acknowledgment

I would like to express my special thanks of gratitude to my supervisor, **Dr. Own Noel Newton Fernando**, for the endless support and aid he provided me towards the completion of this final year project. He has offered meaningful and helpful guidance since project day one that allowed me to headstart the project with ambitious and novel ideas. He has always maintained a positive mindset and outlook towards my progression and achievements regardless of the magnitude. Apart from that, he has been understanding, empathic and a great listener toward my struggles throughout this journey especially handling the project concurrently with Multi-Discipline Project (MDP) module. His continuous efforts and positive attitude towards me have made the success of this project possible.

All in all, this project journey has provided a positive experience. It allowed me to understand the basic software life cycle, and also acquire the basic understanding of being a full-stack developer. I am looking forward to bringing this project experience towards solving real-world problems. Once again, thank you very much!

Introduction

Background

All around the world, extensive waste is a huge problem. In Singapore, we have approximately 28.8% [1] increase in total waste over the last decade since 2008 as shown in Fig.1. Unfortunately, Singapore's landfill has limited space which lasts up to the year 2056 (36 years) [2]. By then, the waste has to be deposited elsewhere which imposes a bigger problem for our future generations. On top of that, a study has shown that Singapore has emerged as the top 5th in solid waste produced rate per kg/capita/day in the world which is about 5 times more than the global average as shown in Fig.2, top 1st in East Asia and Pacific region [3]. With state-of-the-art technologies, we as a small nation can contribute and minimize this problem through innovation. One such innovative system is called "Justgivit", it is a digital platform for giving, where users can exchange or donate unwanted reusable waste to those who need it instead of throwing them away. It is an indirect approach to reduce waste.

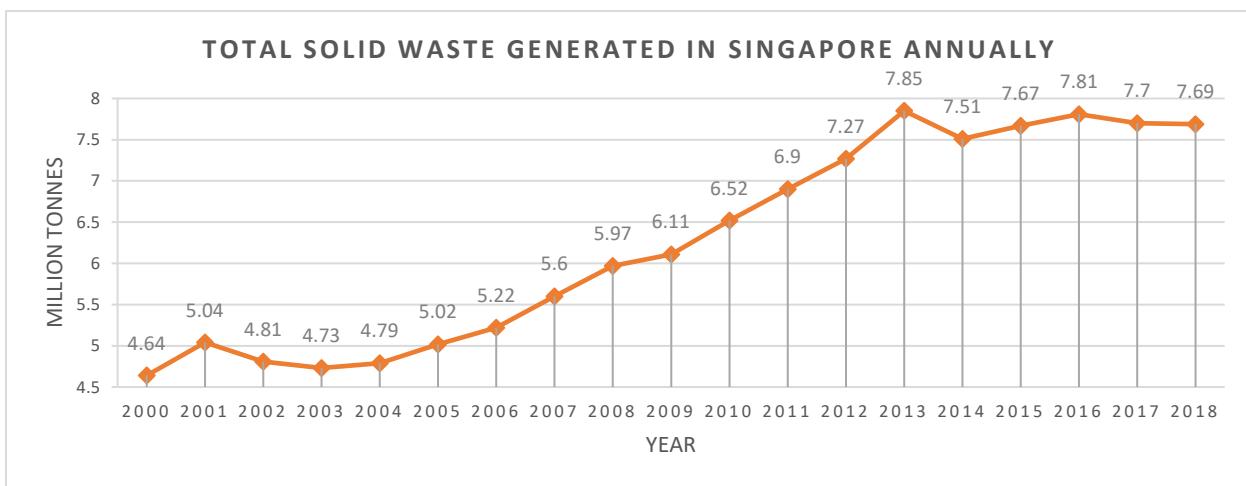


Fig 1, Line graph that shows the increase in total solid waste in Singapore from 2000 to 2018.

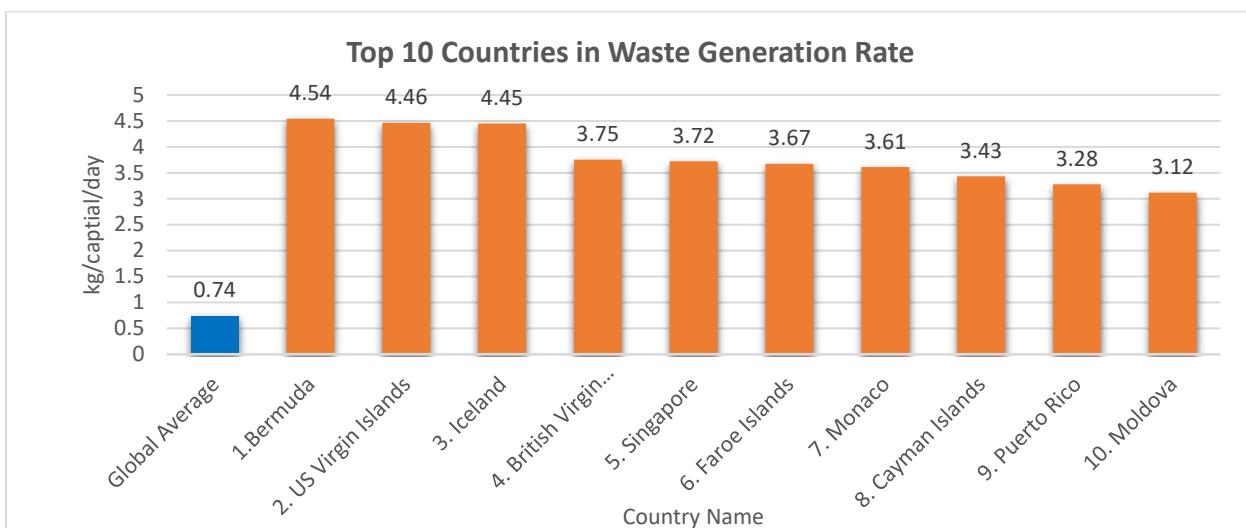


Fig 2, Bar chart that shows that Singapore rank 5th most generated solid waste per kg/capita/day in the world in 2016.

Motivation

Besides the problem statement, in Singapore, our day-to-day life going about donating and exchange operation have plenty of room for improvement. Below we will briefly discuss the current state of donation and exchange, and highlight some case studies about it respectively. Lastly, we will also discuss why the crowdsourcing model is still feasible today and its benefits to society.

Donate

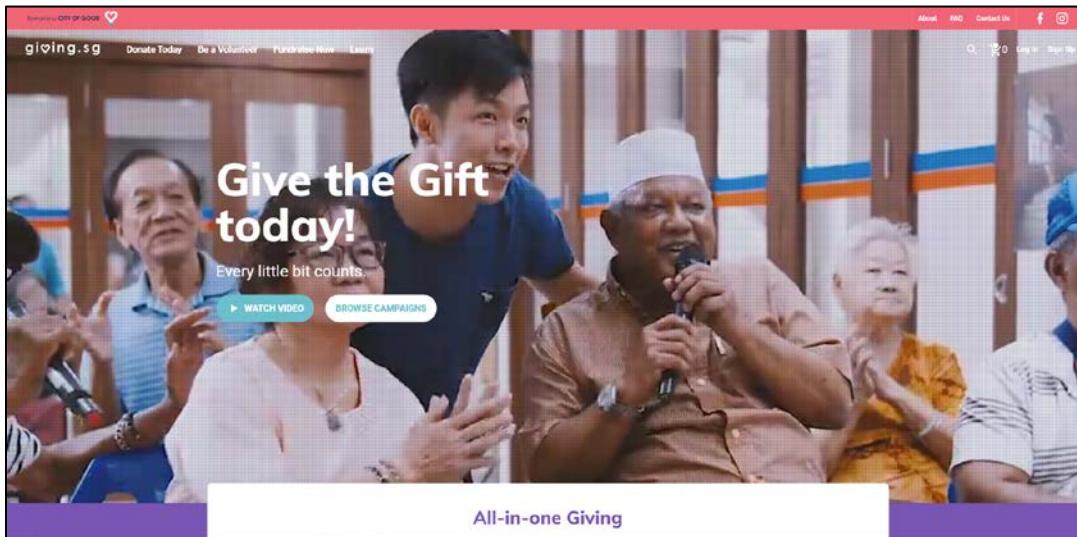


Fig 3, the main page of giving.sg {1}

Studies have shown that altruistic values and efforts to help others and a sense of goodwill play another important role in encouraging individuals to recycle items in Singapore [4]. Currently, unwanted and reuseable goods donation has been operated in a non-digital way, it does not have a centralised system like giving.sg shown in Fig 3 which focuses on the financial aspect of donation. 2 main cases provide the need for digitization in our context.



Fig 4, A commonly distributed leaflet from a charity organization {2}

Case A, HDB residents may receive a charity-related leaflet occasionally as shown in Fig 4. The leaflet approach of donation has flaws. Firstly, it does not hold reliability until someone contacts them and clarify their identity or searching the government's charity website {3} where it shows all the valid registered organization. Besides, based on the publicly available 2020 crime infographic document,

there has been a surge in scam cases, an increase of 65.1% between 2019 and 2020 [5] which typically targets vulnerable groups such as the elderly.



Fig 5, typical security personnel guarding private properties. {4}

Case B, condominium residents may not have the benefit of accessing such services as this has to be approved by all independent security company representatives of these private properties. This may contribute to a large portion of reusable waste being disposed of as trash.

Exchange

Exchanging use products are considered a positive impact most of the time [6]. Below we share some of the similar products available today. We will discuss some features that distinguish them from existing products.

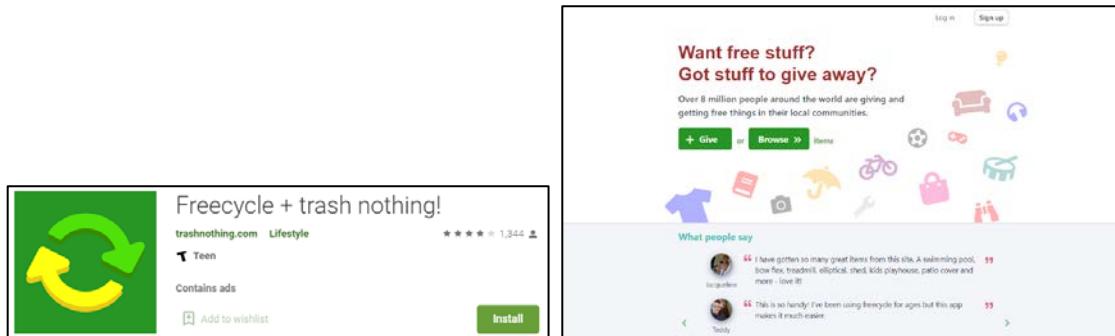


Fig 6, 'Trash Nothing' mobile application shown in Google PlayStore (Left) {5}, 'Trash Nothing' web main page (right) {6}

A product such as 'Trash Nothing' as shown above shares similar features with our product. It exchanges unwanted reusable products to those who need them. However, this product allows items to be placed anywhere without any form of tracking mechanism. This results in a breach of security and reliability. Besides, due to culture and demographic differences in Singapore, it was not directly designed for Singapore. This application is design for worldwide use. Thus, certain features may not be applicable in Singapore. The exchange goods can be placed anywhere in Singapore, which may not be reliable or safe enough for some target audiences.

This application supports the cross-platform capability to reach out to a wider audience, with low investment and still offer a great service to their clients [7]. Cross-platform development also offers great opportunities for the rapid development of high-fidelity prototypes of the mobile application [8].

This allows them to implement their application in one step for a range of platforms, avoiding repetition and increasing productivity [9]. However, the web interface is not optimized as it was still in beta phase development.

Since their API is open source {7}, it is easier to compare with our product. Below are the differences in terms of features between our products.

1. More than just exchange item(s), introduce a donation system that supports charity organisations.
2. Better reliability (QR scan, trackable item status, monitor system by the administrator)
3. Better security (Designated meeting locations)
4. Ease of Use (Auto-fill designation with GPS, Optimise UI usage dependent on the platform)

Crowdsourcing

Crowdsourcing is a process of connecting with a large group of people via the internet, to reach out for their knowledge, expertise, time, or resources. In our context, solving this problem alone is not sufficient. What if we can connect to the world with like-minded people, who are also ‘thinking’ of solving this problem but didn’t think it matters until someone else started it!

“It is no longer about one designer, one client, one solution, one place. Problems are taken up everywhere, solutions are developed and tested and contributed to the global commons, and those ideas are tested against other solutions.” [10]. We can take Bruce Mau and the Institute Without Boundaries claims to describe one of the characteristics of crowdsourcing.

Nowadays, more and more businesses are adopting an open innovation strategy to make extensive use of external valuable ideas and expertise by sponsoring innovative and co-creation activities. Studies have shown that university students are the important potential source of crowdsourcing contributions and have been regarded as seedbeds for innovation [11].

Besides that, thanks to the widespread availability of mobile devices, coupled with the increased accessibility to the wireless internet have caused the proliferation of mobile crowdsourcing applications. [12]. Mobile devices also come with other embedded hardware such as sensors, cameras, and GPS-supported hardware that may allow more possible ways that help unfold the potential of crowdsourcing [13]. Thus, expand to the mobile platform is worth it [14] since it may help to provide better solutions and allows wider accessibility toward our target audiences.

Challenging problems might benefit from solutions from others in different places around the world and the use of crowdsourcing for the resolution of some of these problems may speed up the possibility of finding the best solution. In this project, we don’t just stop at providing solutions about waste, we also enhance how the donation and exchange process in a centralised system that serves a common goal. Our current focus here in this project is rather basic but the end goal is to open crowdsource from both experts and ordinary people to allow efficient improvement of this application. It is low cost, rapid and it has the potential to accelerate development and research due to the amount of work that can be concentrated on a single challenge in a short period. Below are 2 case studies where that describe the relevance of this crowdsourcing model today.

Case A, COVID-19 pandemic has revealed the limitations of the usual organizational processes and the need for greater efforts in managing people and resources to respond adequately to an unexpected and devastating pandemic like this but like other similar events that might occur [15-17]. Due to the flexibility of crowdsourcing, the ‘get it done’ style of work can effectively respond to this issue. Besides, it is not only able to identify solutions to the specific challenge, potentially enhances and improves the existing problem that was not being solved before [15].

Case B, My Starbucks Idea {8} is open innovation, its goal is to increase engagement with customers and giving customers insight into what the company is doing. Top-voted may stand a chance to implement their innovation. This has benefited both Starbucks and the customers. My Starbucks Idea helps the company stay aware of the customers’ needs and it creates loyalty, customer engagement, transparency, and trust [18-20].

Based on these 2 cases, we can see that crowdsourcing can potentially bring this idea into the real world faster as it impacts the end-users needs. For case A, it was the one’s recent solution that solves past solutions which benefit wider users. For case B, it brings a win-win situation for all users. Thus, crowdsourcing remains relevant today.

The key goals of the suggested open crowdsourcing as follows.

1. Broadening participation
 - Recruiting participants to get the best ideal solutions to this problem
2. Marketing
 - This project can be used for brand recognition among participants
3. Fund leveraging
 - Stimulate other organizations to sponsor similar projects.
4. Participant education
 - This allows people who want to learn or explore technical skills and at the same time be part of the contributor to solve a real-world problem.

Below are some of the benefits that JustGivit can bring to Singapore's society with the crowdsourcing model.

1. Gain popularity quicker such that end-users are aware of such application existence.
2. Quicker job opportunities (Collector and Administrators)
3. Indirectly contribution towards NEA's waste management
4. End-users may feel happier toward giving people who need it.
5. Crowd feedback from end-users needs to improve the JustGivit system.
6. Transparency between end-users and JustGivit system.
7. Streamline work process, more efficient use of manpower.

The benefits are worth pursuing this project and bringing this innovative idea to adapt toward impacting the people for the greater good of all.

Objective

The scope of this project “Justgivit” is to build a software system with **cross-platform technologies** that utilize our **smartphones and desktop computers** to minimize our background issues. To achieve this, there are **2 possible ways**, there will be **reusable, unwanted** items to be donated to charity organizations to aid the needy people or to be exchanged with needy people.

Scope

This section will briefly highlight the scope of the system. “JustGivit” consists of various internal and external systems that are catered to different roles of users namely, **donor, exchanger, administrator, collector, and company**. Each role plays an important role to fulfill the objective of this project. Below will briefly describe their roles and the systems used for this project.

1. The **company** will be responsible for requesting needs into the system, such that donors can donate to them respectively.
2. The **donors** will be volunteering to donate reusable and appropriate entities towards charity organizations or needy people.
3. The **exchangers** are free to exchange based on both party agreement and meetup via chat, with recommended designated location after the verification process.
4. The **internal system** is where users can interact with the system through their android (mobile) device or desktop using the Windows operating system with internet access available. There will be an authentication system to access the “Justgivit” private system. There will be a feedback mechanism to ensure continuous improvement of the system. The platform also aims to have an interactive user interface and appealing visual user experience to navigate the respective screens using Flutter.
5. The **external system**, we will be using utilizing Firebase to store our dynamic and credential database. We are using 3 different services such as authentication, Firebase’s Firestore, and Firebase’s Firestore for this project. Apart from that, we are also using Google Cloud Platform (GCP) to provide various APIs for Google Maps services.

These roles will not be implementing or fully implemented. It is still a required concept to assist and explain the donation and exchange process.

1. The **administrator** will oversee monitoring the system properly to prevent the system from being compromised.
2. The **collectors** are tasked to collect the items requested by the donors.

Project Schedule

Roadmap

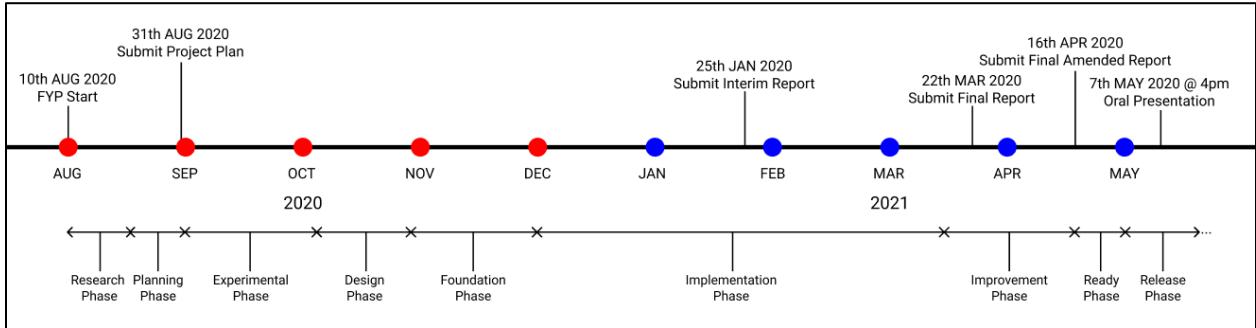
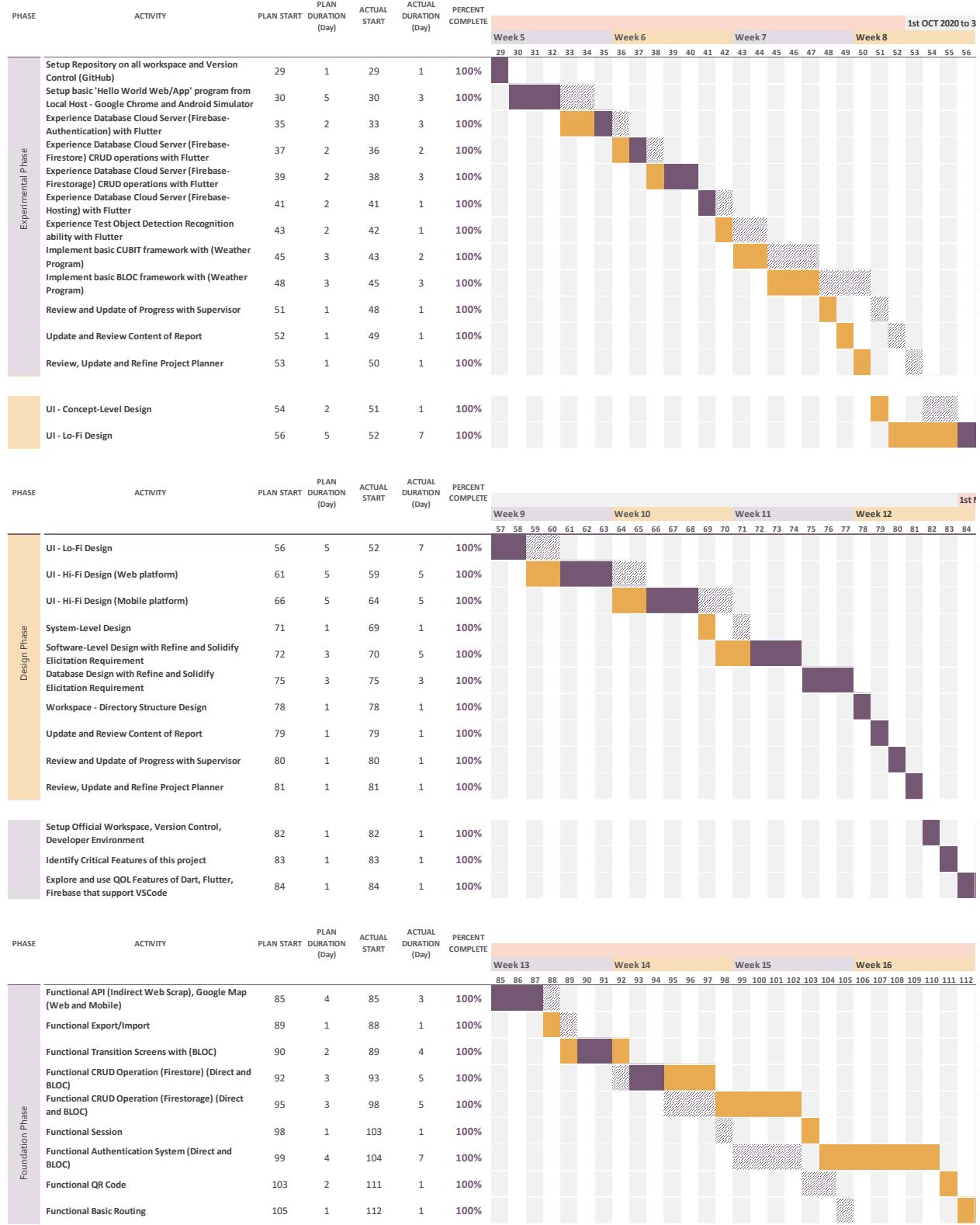
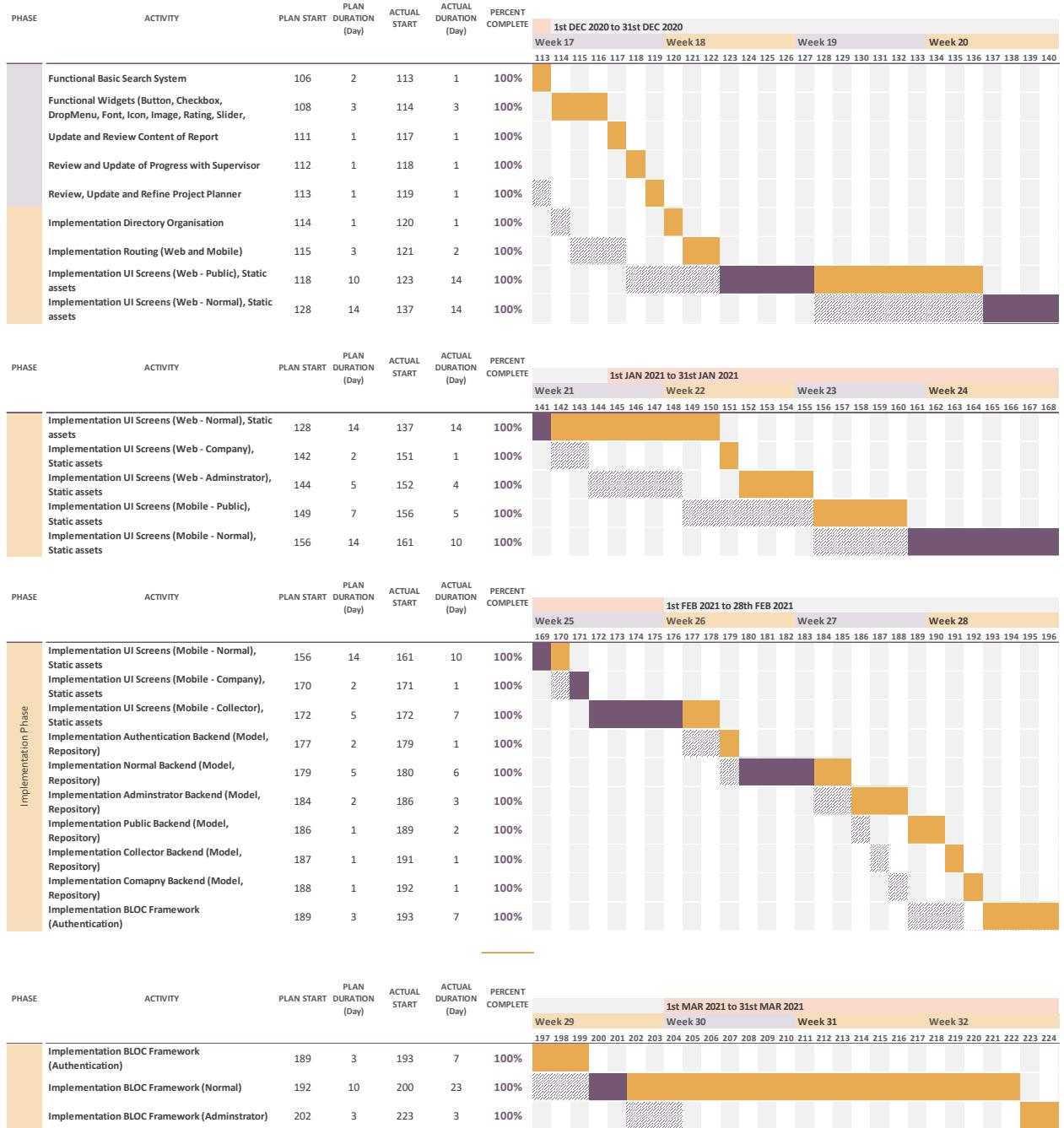


Fig 7, Roadmap of the project from the start to the end of FYP

Gantt Chart







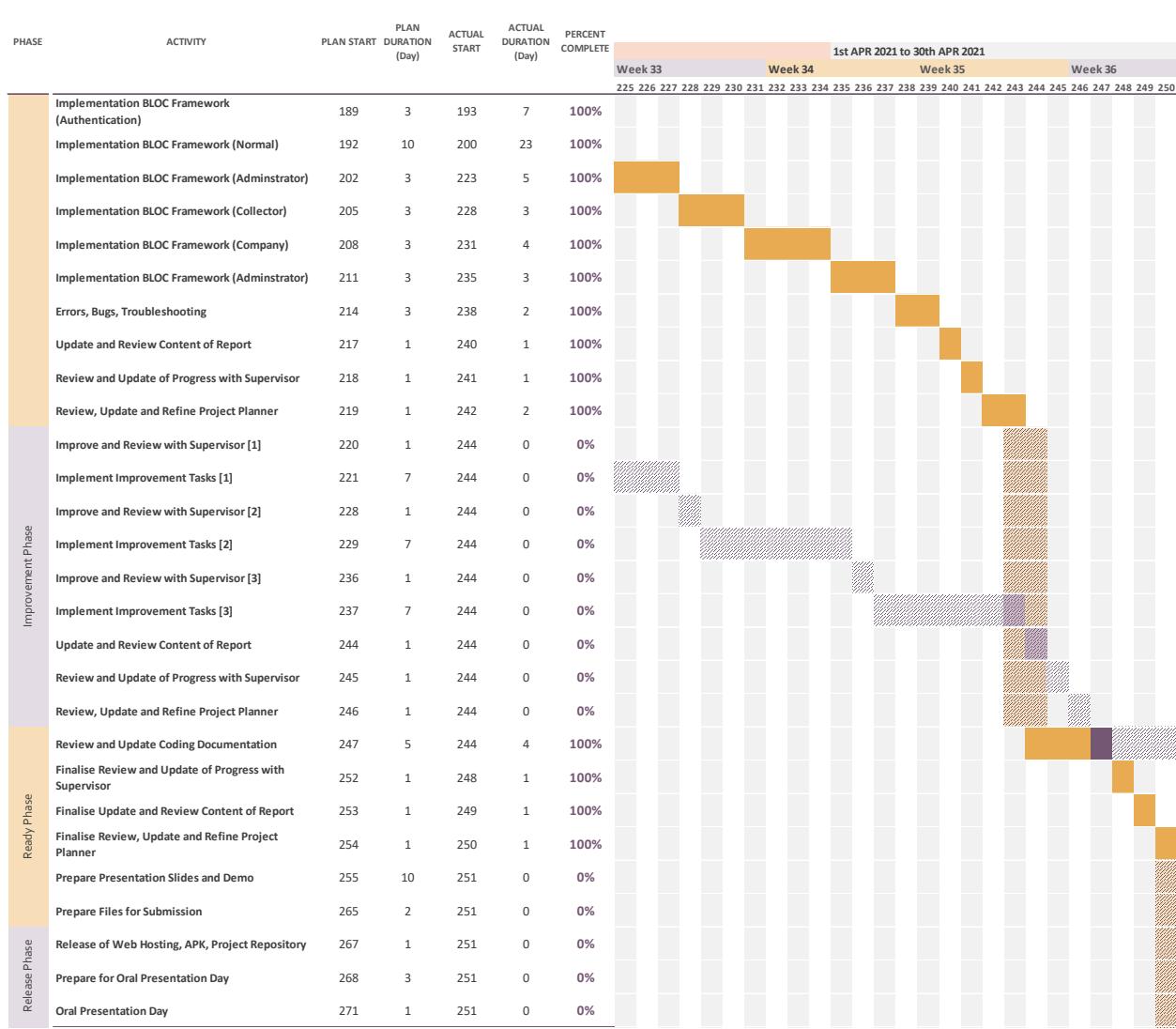


Fig 8, Gantt Chart of the project from the start to the end of the FYP

(Updated until amended final report submission 16th Apr 2021)

Project Track

This section helps to keep track of the tasks between the supervisor and supervisee such as Meeting Minutes, Work Breakdown Structure, and Feedback Channel

Work Breakdown Structure

It is an excel document that shares the development progress with the supervisor. This happens from the foundation phase until the ready phase.

| Phase | Task ID | Classify | Task Title | Task Description | Status | Date Completed | Remarks |
|--------------|---------|------------|------------------------------------------------------------------|-------------------------------------------------------------------|-------------|-----------------|----------------------------------------------------------------------------------------|
| Template | | | | | | | About A |
| Setup | T - 1 | Sample | GitHub Repository | Short Desc A (Expected Action) | Not Started | 11.08.2020 | |
| | A - 1 | N/A | Backend Setup | Setting Up Webserver to support multi-platform | Not Started | 11.08.2020 | Authentication (Normal E-mail and G-mail) |
| | A - 2 | N/A | Generate APK | Allows Firebase's remote API services | Not Started | 15.08.2020 | Initialize after approval of proposed design |
| | A - 3 | Deploy | Hosting Setup | Allows Tester to test mobile app | Not Started | 15.08.2020 | Initialize after approval of proposed design |
| | A - 4 | Full-Stack | Create App Icons like Apple's Interface or List View | Hosting for Web Access to test web | Not Started | 15.08.2020 | Initialize after approval of proposed design |
| | | | | Allows navigation of functional modules easily | Not Started | 17.08.2020 | Initialize after approval of proposed design |
| Experimental | D - 1 | N/A | Setup Function Correctly | Learn how to set up function correctly | Not Started | 10.09.2020 | |
| | D - 2 | N/A | Execute Function on Device Web Access | Learn how to set up web access up | Not Started | 10.09.2020 | |
| | D - 3 | N/A | Execute in Samsung S9 | Learn how to execute the mobile platform | Not Started | 11.09.2020 | |
| | B - 4 | Full-Stack | A Functional Authentication System | Login, Logout, Show Error | Not Started | 11.09.2020 | |
| | B - 5 | Full-Stack | Image Recognition System | Show confidence target(s) with trained Models (Default Model). | Not Started | 11.09.2020 | |
| | B - 6 | Full-Stack | Public and Private View with Authentication | Show was was viewable base on non-auth and auth view | Not Started | 17.09.2020 | |
| | B - 7 | Full-Stack | Mobile Application | Ability to download and install on other mobile device | Not Started | 22.09.2020 | |
| | B - 8 | Full-Stack | Generate APK (test) | Ability to download and install on other mobile device | Not Started | 18.09.2020 | |
| | B - 9 | Full-Stack | Create basic BLOC software pattern | Test a widely used pattern cleaner code and rigid structure | Not Started | 28.09.2020 | |
| | B - 10 | Full-Stack | Simple Coffee Ordering App | To expose the knowledge of firestore CRUD in real time | Not Started | 18.09.2020 | |
| Foundation | C - 1 | Frontend | Section Page (Rename classify as subscreen) | Show working section page | Not Started | 22.11.2020 | |
| | C - 2 | Frontend | Clickable / Touchable Icon | Show working icon | Not Started | 15.11.2020 | |
| | C - 3 | Frontend | Clickable / Touchable Text | Show working interactive Text Content | Not Started | 16.11.2020 | |
| | C - 4 | Frontend | Clickable / Touchable Image | Show working interactive Image | Not Started | 17.11.2020 | |
| | C - 5 | Frontend | Drop Menu | Show working dropdown menu | Not Started | 29.01.2021 | Long list need better widget to support render |
| | C - 6 | Frontend | Loading Screen | A loading animation base on JustGivit theme | Not Started | 15.11.2020 | |
| | C - 7 | Frontend | Listing Objects in body content | Show listing in right timing for desktop | Not Started | 17.11.2020 | |
| | C - 8 | Frontend | Show Textarea | Show working textarea | Not Started | 18.11.2020 | |
| | C - 9 | Frontend | Show Toolbar | Scrollbar should scale correctly | Not Started | 19.11.2020 | |
| | C - 10 | Frontend | View Table Content | Hover should respond a pop out / fade in / changes to the element | Not Started | 06.11.2020 | Not working well for Web when window resize |
| | C - 11 | Frontend | Hover Trigger (Desktop) | A overlay for effect when hovering open and close | Not Started | 15.11.2020 | |
| | C - 12 | Frontend | Pop-Up/Alert/Overlay Window | Show different controls in the same page | Not Started | 15.12.2020 | It need bloc mechanism |
| | C - 13 | Frontend | Multi-tags show different content in the same page | Drag and Drop and Exist when trigger a action | Not Started | 15.12.2020 | |
| | C - 14 | Frontend | Notify Number | Ability to setup pin-point in a map | Not Started | 18.12.2020 | Snackbar effect done together with testing UI |
| | C - 15 | Frontend | Singapore Map with Pin-Point | Step version / Fluid style | Not Started | 03.11.2020 | Done together with GPS system |
| | C - 16 | Frontend | Fill Bar Effect depend on value | Showcase the branch effect | Not Started | 14.01.2021 | Downscale to non-dynamic but clickable char |
| | C - 17 | Frontend | Ability to generate a nice tree base on a set of relation values | Cancelled | Cancelled | Time constraint | |
| | C - 18 | Frontend | Show QR code | Ability to convert from STRING to QRCode | Not Started | 30.11.2020 | QR Package offer such function |
| | C - 19 | Frontend | Open Camera | Changes base on user interaction | Not Started | 01.12.2020 | Open camera with GPS system |
| | C - 20 | Frontend | Overlay Camera | To show the result after image processing | Not Started | 14.11.2020 | Time constraint |
| | C - 21 | Frontend | Implement Page Not Found | Bind tab function into | Not Started | Discontinued | For Mobile it serve as user encounter fatal error |
| | C - 22 | Frontend | Section Page | Resizing image to Standard | Not Started | Discontinued | Changes to design requirement therefore no |
| | C - 23 | Backend | Image Calibration | Searching base on uniqueness and default | Not Started | 27.11.2020 | Workable with image picker package |
| | C - 24 | Backend | Send Email | Going to pages back and forth | Not Started | 01.12.2020 | Simple Search Done |
| | C - 25 | Backend | Routing Page (Push-Pop) | Workable Firestore function (text data) | Not Started | 14.11.2020 | Is part of the implemented design in AA |
| | C - 26 | Backend | Send and Received data from Firestore | Workable storage function (image data) | Not Started | 25.12.2020 | |
| | C - 27 | Backend | Send, Received data from Storage | A login and logout function using non-Gmail | Not Started | 28.12.2020 | |
| | C - 28 | Backend | Authentication System (Normal Email) | A login and logout function using non-Gmail | Not Started | 28.12.2020 | |
| | C - 29 | Backend | Authentication (G-Email) | A login and logout function using Gmail | Not Started | 28.12.2020 | |
| | C - 30 | Backend | Public and Private View | Control base on authentication and other permission | Not Started | 26.11.2020 | Reduce complexity but will be implement if able to keep track of auth status correctly |
| | C - 31 | Backend | Downloads APK | Download APK for mobile user | Not Started | 27.12.2020 | |
| | C - 32 | Backend | Listing Changes and Control | Update Listing In Real Time | Not Started | 29.12.2020 | Work only for firestore , not firestore , prop |
| | C - 33 | Backend | Create Random Spread Tree-Like with Weights using Seed | A algorithm for reward system | Not Started | 04.01.2021 | Time constraint |
| | C - 34 | Backend | Algorithm to generate QRCode STB base on a given data | QR code for payment verification purpose | Not Started | 04.01.2021 | Identified the code responsible for such operation |
| | C - 35 | Backend | GPS Address from phone | All permission from Mobile OS | Not Started | 10.12.2020 | Mobile offer such function and proven work |
| | C - 36 | Backend | GPS get XY coordinates | To display pin-point or require GPS as dependencies | Not Started | 05.12.2020 | Proven workable |
| | C - 37 | Backend | Gov (Web Scrap / API) list of register charity website | To verify new charity organisation | Not Started | 18.12.2020 | Conclude to use downloadabit static data |
| | C - 38 | Backend | Camera permission from phone | unlock ability the use camera | Not Started | 27.12.2020 | |
| | C - 39 | Backend | Camera capture for image processing | Take image and process for model | Not Started | Cancelled | Time constraint |
| | C - 40 | Backend | Image processing Algorithm | Image Filtering using latest tech | Not Started | Cancelled | Time constraint |
| | C - 41 | Backend | Training Model for JustGivit Context | Create a model for justgivit | Not Started | Cancelled | Time constraint |

Fig 9, a partial content of WBS shared excel document with the supervisor.

Assumptions

| S/N | Title | Description |
|-----|-------------------------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------|
| 1 | Limitless budgets | This application was sponsored by investors and stakeholders. This application was also run by a company with infinite wealth to develop this project. |
| 2 | Monitor by Administrators | The company hired a team of administrators to monitor the system of this application |
| 3 | Supporting Operating System | This application only supports Window 10 Desktop and Android device exclusively |
| 4 | Supporting Browser | This application only supports Google Chrome to showcase the features of this application |
| 5 | Newbie Users | All users do not know how to use this application |
| 6 | Government Datasets | The datasets provided by the government are legitimate, such as registered charity organization companies |
| 7 | User device software and hardware version | The version follows the recommended Software and Hardware specification to yield the best performance and minimal error. |

Constraints

| S/N | Title | Description |
|-----|--------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 1 | Government Datasets API | There is no API from a registered charity organization. Thus, the downloaded dataset was used instead which does not support real-time updated dataset. {9} |
| 2 | Firebase Services Limit | Firebase's Spark Plan has limited usage on Firebase's Firestore and Firebase's Firestorage read and writes count. Thus, a database system was designed to minimise read and write count. {10} |
| 3 | Flutter Web Support | Flutter Web support is still in the beta development phase, therefore some widget features are not compatible. {11} |
| 4 | Limited Knowledge | Performance optimization is based on our best understanding |
| 5 | Google Cloud Service API | Javascript Maps and Android Map SDK API are limited to 500 requests per second. {12} |
| 6 | Government Datasets | The datasets provided by the government assume legitimate such as registered charity organization companies. {9} |

Project Environment

This section describes the objects used throughout this project.

Internal

Software

| S/N | Software Name | Recommended Version | Purpose |
|-----|-----------------------|------------------------------------------------|-----------------------------------------------------------------------------------------------------------------------------------|
| 1 | Window 10 | Version 1909 (OS build 18363) and above | An operating system environment for work to be carried out. |
| 2 | Andriod Emulator | SDK 29 and above | An emulator to simulate Android with SDK 29 and above for debugging. |
| 3 | Android Studio | 4.1.2 and above | A dependency software that supports Android Emulator and Flutter SDK. |
| 4 | Flutter | Channel beta, 1.25.0-8.2.pre and above | It is a development kit to develop cross-platform capability in a single codebase. Using beta version to support web development. |
| 5 | Firebase | 9.1.2 and above | It is used for web or local hosting of the application. |
| 6 | npm | 6.14.9 and above | A package manager that is a pre-requisite for Firebase hosting. |
| 7 | Google Chrome Browser | Version 87.0.4280.88 (Official Build) (64-bit) | It is used to view the web user interface of the application. |
| 8 | VSCode | 1.52.1 and above | It is an Integrated Development Environment (IDE) that provides a work environment for software programming development. |
| 9 | Dart | 2.12.0 and above | It is a client-optimised programming language for applications that support multiple platforms. |

Hardware

| S/N | Hardware Name | Model | Purpose |
|-----|--------------------------------------------|-------------|-------------------------------------------------------------------------------------------------------------|
| 1 | Laptop | Lenovo Z470 | A portable device to work with the project. |
| 2 | Ethernet port | Cat 7 | Provide Direct Access to the internet |
| 3 | Power Adaptor | 1022-5DU AC | Provide power to run the laptop. |
| 4 | Wifi Docker | D-Link | Provide Wireless Access to the internet |
| 5 | Keyboard | Logitech | It is a QWERTY standard that inputs characters into the computer system. |
| 6 | Mouse | X-Craft Pro | Hand-held point device that displays the pointer on the screen and provides input into the computer system. |
| 7 | High Definition Multimedia Interface Cable | Standard | Provide HD signal for transferring high definition video to monitor to display. |

| | | | |
|----------|---------------|---------------------------------------|----------------------------------------------------------------------------------------------------------|
| 8 | Monitor | SAMSUNG LS24F350FHEXXS 24IN LED | Support up to 1920x1080 resolution to better view standard UI from the application for the web platform. |
| 9 | Mobile Device | Samsung S9 | It is used to test the application usage on the Android platform. |

External

| S/N | Services | Purpose |
|----------|-----------------------|----------------------------------------------------------------------------------------------------|
| 1 | Google Cloud Platform | Provide APIs for google map-related services on both Android and Web platforms. |
| 2 | Firebase | It is used as data storage such as text and images. It is also used to store user authentications. |

Technical Specification

This section will be sharing all technical-related content about this project.

Technical Description

Experiences

The choices are heavily prioritised based on my ability and the complexity of this project.

Flutter

Flutter is a UI software development kit created by Google. It supports application development for different operating systems. The Flutter community has provided countless guidance and useful resources. This has tremendously aided me in the development process. Besides that, Flutter also supports a single-code base that shares code in multiple different operating systems. The language used was DART [21]. It is object-oriented with C-style syntax. It can be compiled into native code or Javascript.

To achieve cross-platform capabilities, there are multiple ways to approach it such as Flutter, React Natives, and Progressive Web Apps.

I choose Flutter is simply due to its beginner friendliness since I have no experience in mobile development. Also, the scale of this project requires a fast development process to complete them; Apart from Flutter's single-code base [22], it also has this feature called 'Hot reload' [23] which allows quick and easy debugging.

However, no software is perfect; there are drawbacks to Flutter too. Its web support is still beta [24] and was not fully tested for production use. For example, the scroller function does not work well with dynamic window size during runtime, an interactive scrollbar feature [25]. On the other hand, it does not have many internal libraries to support native hardware features. Thus, the need to use plenty of external libraries as dependencies which may not be good when there are new updates from Flutter's development team.

Firebase

It is one of the Backend-as-a-Service (BaaS) solutions on the market. It supports sharing of data between different operating systems. It is also a cloud-hosted software that supports real-time changes to the data. It consists of many ready-to-use services such as Firestore, Web Hosting, Authentication, Firestorage. This solution help developer to boost their implementation speed without having to worry about hectic backend configuration or setup [26].

Firebase solution like any other similar cloud computing. It has experienced increased popularity over the last decade [27]. Firebase features cross-platform data [28], and real-time updates [29]. In addition, it is sufficiently free of charge for this project base on Firebase 'Spark' plans [30]. It is also beginners friendly as well based on my experience.

However, the disadvantage of using Firebase is that it supports limited query capabilities [31] but it does not affect the complexity of this project.

Google Cloud Platform (GCP)

In this project, there are 2 APIs from the Google Cloud service provider being used for Android and Web, namely, Maps JavaScript API and Maps SDK for Android [32].

Data Dictionary

Before we dive into the technical information, it is important to define the technical terms to prevent misinterpretation or ambiguity of certain terms.

| S/N | Term | Description |
|------------|-----------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 1 | Public | It refers to any unauthenticated users in the system. These users are only allowed to view publicly available data only. |
| 2 | Administrator | It refers to an authenticated user with access privileges to manage or monitor donor, exchanger, collector, and company users. |
| 3 | Collector | It refers to an authenticated user with access privileges to collect donor's item(s). |
| 4 | Company | It refers to an authenticated user with access privileges to register the item needs. These charity-related companies are approved by the commissioner of charities. They are required to abide by the charities' acts and regulations. |
| 5 | Donor | It refers to an authenticated user with access privileges, their main purpose is to donate items to a charity company. |
| 6 | Exchange | It refers to an authenticated user with access privileges, their main purpose is to exchange items with other exchange users |
| 7 | Normal user | It refers to both Donor or Exchanger to simplify the terms. |
| 8 | Stateful Class | It refers to elements in the UI that change over time due to events triggered by the user or external API |
| 9 | Stateless Class | It refers to elements in the UI that do not change during run-time |
| 10 | Widgets | It refers to an element of the user interface. It also defines the design, interaction, and characteristics of the element. |

Functional Requirement

Cross-Platform

Public

1.1 The public users must be able to register as a company successfully to the system.

1.1.1 The public users must not be able to register an existing account.

1.1.2 The public users must be able to select one of the companies in the given list.

1.1.3 The public users must be able to provide a Singapore format phone or office number.

1.1.4 The public users must be able to provide a valid password.

1.1.4.1 A valid password must contain at least 8 minimum characters.

1.1.4.2 A valid password must contain at least 1 capital letter.

1.1.5 The public users must re-enter the password correctly.

1.1.5.1 The re-enter password must match the password in point 1.1.3.

1.1.6 The public users must be able to upload a profile image optionally.

1.1.7 The public users must be able to check a tickbox to agree to the terms and conditions.

1.2 The public users must be able to register as a donor/exchanger successfully to the system.

1.2.1 The public users must not be able to register an existing account.

1.2.2 The public users must be able to provide a username.

1.2.2.1 The username must between 5 to 16 characters.

1.2.2.2 The username must only contain alphabets characters.

1.2.3 The public users must be able to provide email correctly.

1.2.3.1 The email domain must be within the selectable email domain list namely, gmail.com, yahoo.com, outlook.com, hotmail.com.

1.2.4 The public users must be able to provide a valid password.

1.2.4.1 A valid password must contain at least 8 minimum characters.

1.2.4.2 A valid password must contain at least 1 capital letter.

1.2.5 The public users must re-enter the password correctly.

1.2.5.1 The re-enter password must match the password in point 1.2.3.

1.2.6 The public users must be able to upload a profile image optionally.

1.2.7 The public users must be able to check a tickbox to agree to the terms and conditions.

1.3 The public users must be able to view donate information.

1.3.1 The donate information must contain the item type, item name, quantities.

1.3.2 The public users must be able to select and view the respective item type donate information.

1.4 The public users must be able to view exchange information.

1.4.1 The exchange information must contain a 5-star rating, item name, and username of the exchanger, the image represents the exchange item.

1.4.2 The public users must be able to select and view the respective item type exchange information.

1.4.3 The public users must be able to view selected exchange information for details.

1.4.3.1 The details must contain, the number of exchanges, posted date, unique ID, the exchanger preferences, the exchanger feedback, the exchange's item Q&A.

1.4.3.1.1 The exchanger feedback consists of images of the exchange item, username, unique ID, remarks, posted date, and 5-rated stars.

1.4.3.1.2 The exchange's item Q&A consists of a username, posted date, unique ID, questions, and answers.

1.4.3.1.3 The preferences consist of exchanger's 3 locations, 3 days and time, 3 recommended exchange item.

1.5 The public users must be able to login into the system successfully.

1.5.1 The public users must be able to provide an email.

1.5.2 The public users must be able to provide a password.

1.5.3 The public users must not be able to login if provided 1.5.1 and 1.5.2 is invalid.

1.5.3.1 The credential is invalid when the email does not exist in the system or mismatches credentials.

1.5.4 The public users must be able to reset their passwords with a valid email.

Donor or Exchanger (Normal)

2.1 The normal users must be able to view donate information same as point 1.3.

2.2 The normal users must be able to view exchange information same as point 1.4.

2.3 The normal users must be able to donate items through the system.

2.3.1 The normal users must provide collection location

2.3.1.1 The normal users must be able to provide the avenue of the address.

2.3.1.1.1 The avenue of the address must not be empty.

2.3.1.2 The normal users must be able to provide the unit number of the address.

2.3.1.2.1 The unit number of the address must not be empty.

2.3.1.3 The normal users must be able to provide the postal code of the address.

2.3.1.3.1 The postal code must have exactly 6 characters.

2.3.1.3.2 The postal code must contain digits only.

2.3.2 The normal users must be able to upload a donate item image.

2.3.2.1 The image must not be the default image.

2.3.3 If the normal users had donated and collected the item by the collector.

2.3.3.1 The normal users can provide collector feedbacks.

2.3.3.1.1 The feedbacks must contain 5-rated hearts.

2.3.3.1.2 The feedbacks must be able to provide remarks up to 50 characters.

2.3.4 If the normal users had provided collector feedbacks

2.3.4.1 The normal users must be able to claim rewards 1, 3, 5 points respectively.

2.3.5 The normal users must be able to spend reward points.

2.3.5.1 Each spends the normal user deduct 1 point from the total reward point.

2.3.6 The normal users must be able to check the status of the donation.

2.3.6.1 The donation status must be able to contain 2 different sets of information, the active lists or the achieve lists.

2.3.6.1.1 The "active" lists provide the donor's unique ID, item type, item name, posted date, status is 'collecting'.

2.3.6.1.2 The "achieve" lists provide the previous records of active lists information in point 2.3.6.1.1 except status is 'collected'.

2.4 The normal users must be able to exchange items through the system.

2.4.1 The normal users must set their own preferences.

2.4.1.1 The normal users must be able to set their preferred time range.

- 2.4.1.2** The normal users must be able to select up to 7 preferred days.
- 2.4.1.3** The normal users must be able to set up to 3 meeting location(s).
- 2.4.1.4** The normal users must be able to set up to 3 recommended exchange item(s).
- 2.4.1.5** If no preferred day is selected.
 - 2.4.1.5.1** The normal users must not be able to exchange.
- 2.4.1.6** If no preferred location is selected.
 - 2.4.1.6.1** The normal users must not be able to exchange.
- 2.4.2** The normal users must be able to offer an item to exchange.
 - 2.4.2.1** The normal users must add an offer to exchange.
 - 2.4.2.1.1** The normal users must select their exchange item as an offer.
 - 2.4.2.2** The normal users must be able to provide remarks optionally up to 50 characters.
 - 2.4.2.3** If the normal users did not provide an offer exchange item.
 - 2.4.2.3.1** The normal user will not be able to submit an offer.
- 2.4.3** The normal users must be able to ask or answer questions.
 - 2.4.3.1** If the normal users are viewing their exchange item.
 - 2.4.3.1.1** The normal users must be able to answer questions.
 - 2.4.3.2** If the normal user is viewing another normal user's exchange item.
 - 2.4.3.2.1** The normal users must be able to ask questions.
- 2.4.4** The normal users must be able to chat with the other normal users.
 - 2.4.4.1** The normal users must be able to view all other normal users that chatted with.
 - 2.4.4.2** The normal users must be able to provide a message to other normal users.
- 2.4.5** The normal users must be able to accept exchange items if there exists an offer from other normal users.
- 2.4.6** If an offer is accepted by the normal users.
 - 2.4.6.1** The normal users must be able to provide either QR Code string or QR Code Scanner from other normal users vice versa.
 - 2.4.6.2** The normal users must be able to provide exchanger feedback.
 - 2.4.6.2.1** The normal users must be able to upload an exchange image.
 - 2.4.6.2.2** The normal users must be able to provide a 5-rated heart.
 - 2.4.6.2.3** The normal users must be able to provide remarks up to 50 characters.
- 2.4.7** The normal users must be able to check the status of the exchange.
 - 2.4.7.1** The exchange status must be able to contain 2 different sets of information, the active lists or achieve lists.
 - 2.4.7.1.1** The active lists provide the exchanger's ID, item names, posted date, status 'request' or 'exchanging'.
 - 2.4.7.1.2** The achieve lists provide the previous records of active lists information in point 2.4.7.1.1 except status is 'exchange'.
- 2.5** The normal users must be able to provide application feedback.
 - 2.5.1** The normal users are allowed to submit more than one application feedback.
 - 2.5.2** The normal users must be able to provide 5-rated hearts.
 - 2.5.3** The system will record down the current timestamp once a normal user submitted the application feedback.
 - 2.5.4** The normal users must be able to provide feedback optionally.
 - 2.5.4.1** The feedback must not be more than 50 characters.

Company

- 3.1** If the company user is approved by the administrator.
 - 3.1.1** The company user must be able to view their own needs.
 - 3.1.1.1** The needs must be able to show the item type, item name, quantity value, unique ID, posted date, and status ‘pending’, ‘approved’, ‘collecting’, ‘collected’.
 - 3.1.2** The company must be able to add needs.
 - 3.1.2.1** The company user must be able to select the item type.
 - 3.1.2.2** The company user must be able to select the item name.
 - 3.1.2.3** The company user must be able to select the quantity value between 0 to 100.

Web platform exclusively

Administrator

- 4.1** The administrator must be able to view Singapore map details
 - 4.1.1** The details must be able to provide pin-points
 - 4.1.1.1** Each pin-point contains the location, usernames, item names from normal users.
- 4.2** The administrator must be able to manage company users.
 - 4.2.1** The administrator must be able to view the company information.
 - 4.2.1.1** The company information must contain the company logo, company name, description, address, and contact number.
 - 4.2.2** The administrator must be able to accept or reject the company in the list.
- 4.3** The administrator must be able to manage needs.
 - 4.3.1** The administrator must be able to view the needs information
 - 4.3.1.1** The needs information must contain the item type, item name, company name, and contact number.
 - 4.3.1.2** The administrator must be able to accept or reject the needs in the list.
- 4.4** The administrator must be able to manage donor users.
 - 4.4.1** The administrator must be able to view all donors’ uploaded images.
 - 4.4.2** The administrator must be able to select the donor’s uploaded images.
 - 4.4.3** The administrator must be able to delete selected donor’s uploaded images.
- 4.5** The administrator must be able to manage collector users.
 - 4.5.1** The administrator must be able to add collector users.
 - 4.5.1.1** The administrator must be able to provide the username of the collector.
 - 4.5.1.2** The administrator must be able to provide the full name of the collector.
 - 4.5.1.3** The administrator must be able to upload images represent the collector.
 - 4.5.1.4** The administrator must be able to provide the contact number of the collector.
 - 4.5.1.5** The administrator must be able to provide the address’s avenue of the collector.
 - 4.5.1.6** The administrator must be able to provide the address’s unit number of the collector.
 - 4.5.1.7** The administrator must be able to provide the address’s postal code of the collector.
 - 4.5.2** The administrator must be able to select a collector user.
 - 4.5.3** The administrator must be able to delete the selected collector user.
 - 4.5.3.1** The deleted collector will be permanently removed from the system.
- 4.6** The administrator must be able to manage application feedback.
 - 4.6.1** The administrator must be able to view the feedback information.

- 4.6.1.1** The feedback information consists of a unique ID, 5-rated stars remarks, and published date.
- 4.6.2** The administrator must be able to accept or reject feedback information.
 - 4.6.2.1** Accepted feedback information represents profile image will available in public.
 - 4.6.2.2** Rejected feedback information will be achieved in the system.

Mobile platform exclusively
Collector

- 5.1** If the collector has internet access and a camera-enabled.
 - 5.1.1** The collector must be able to assign available collect item locations.
 - 5.1.1.1** The collect item must be able to view the address provided by donor users.
 - 5.1.1.2** The collect item must be able to view the image representation of the collection item.
 - 5.1.2** The collector must be able to capture the collected item.
 - 5.1.2.1** The collector must be able to provide remark optionally.

Non-Functional Requirement

Web and Android

| S/N | Quality Attribute | Description |
|-----|-------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 1 | Capacity | Up to 500 users can request Google Maps API services per second. |
| 2 | Usability | The users shall be able to use all the system functions after 4 hours of training. After this training, the average number of errors made by experienced users shall not exceed 3 per hour of system use. |
| 3 | Performance | Response time must be less than 100ms per request from the user action. |
| 4 | Scalability | When the Android platform is under maintenance, the web platform can operate as usual vice versa. |
| 5 | Availability | The uptime of the application should be 99.99% of the time per annual |

Use Case Diagram

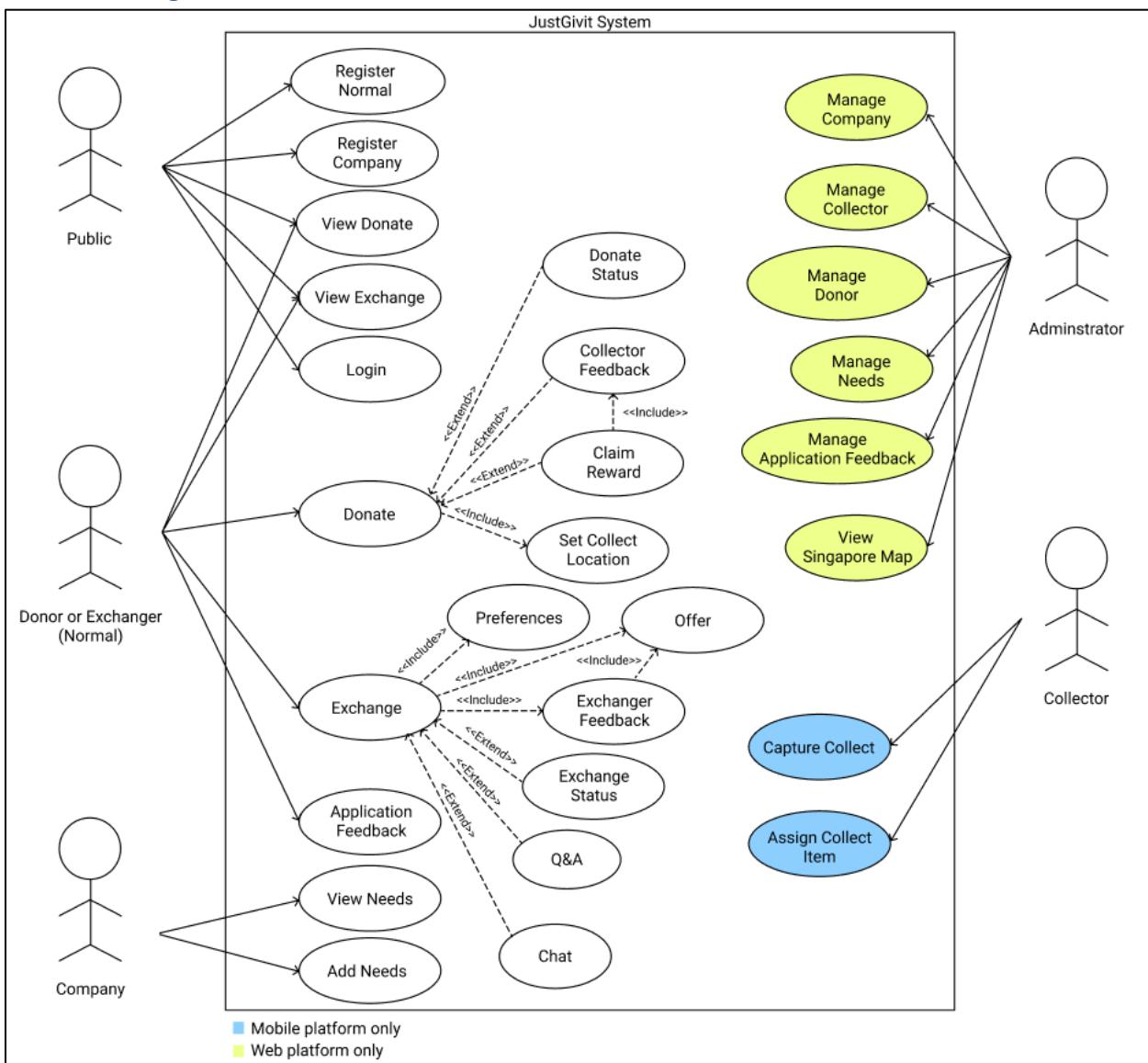


Fig 11, depicts the **partial** (key features) use cases of the 'JustGivit' system.

Use Case Description

| | | | |
|-----------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|---------------------------|---------------|
| Use Case ID: | UC-01 | | |
| Use Case Name: | Register Normal | | |
| Created By: | Sam Jian Shen | Last Updated By: | Sam Jian Shen |
| Date Created: | 09-01-2020 | Date Last Updated: | 03-03-2021 |
| Actor: | Public | | |
| Description: | It is to allow the normal account to be created in the system. To be used as a donor or exchanger. | | |
| Pre-Conditions: | - | | |
| Post-Conditions: | The public user will be authenticated and login as normal users. | | |
| Priority: | Medium | | |
| Frequency of Use: | Medium | | |
| Flow of Events: | <ol style="list-style-type: none"> 1. The public user inputs: USERNAME, EMAIL, PASSWORD, RE-PASSWORD. 2. The public user may choose to upload a profile image. 3. The public user ticks the TERMS AND CONDITIONS checkbox. 4. The public user clicks the 'REGISTER' button. 5. The system verifies the inputs and checkbox 6. If system verification is successful 7. The public user inputs will be saved into the system. 8. The system will display successful screen 9. The public user will be redirected to normal users on the main page. | | |
| Alternative Flows: | <p>AF-S1: If the public user submits with at least one empty input and an unticked box.</p> <ol style="list-style-type: none"> 1. The system will display incomplete messages. <p>AF-S2: If public user submit with at least one invalid input and non-empty input</p> <ol style="list-style-type: none"> 1. The system will display invalid messages. <p>AF-S3: If user verification fails</p> <ol style="list-style-type: none"> 1. Go back to step 1. | | |
| Exceptions: | <p>EX-1: If any AF-SX fails or any steps</p> <ol style="list-style-type: none"> 1. The system will display an error screen with the respective error code. | | |
| Includes: | - | | |
| Special Requirement: | - | | |
| Assumptions: | - | | |
| Notes and Issues: | - | | |

| | | | |
|-----------------------|------------------|---------------------------|---------------|
| Use Case ID: | UC-02 | | |
| Use Case Name: | Register Company | | |
| Created By: | Sam Jian Shen | Last Updated By: | Sam Jian Shen |
| Date Created: | 09-01-2020 | Date Last Updated: | 03-03-2021 |
| Actor: | Public | | |

| | |
|-----------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Description: | It is to allow a company account to be created in the system. This account will need to be approved by the Administrator. |
| Pre-Conditions: | - |
| Post-Conditions: | - |
| Priority: | Medium |
| Frequency of Use: | Medium |
| Flow of Events: | <ol style="list-style-type: none"> 1. The public user inputs: COMPANY NAME, CONTACT NUMBER, PASSWORD, and RE-PASSWORD. 2. The public user ticks the TERMS AND CONDITIONS checkbox. 3. The public user clicked the 'REGISTER' button. 4. The system verifies the inputs and checkbox 5. If verification is successful 6. The public user inputs will be saved into the system. 7. The public user will be redirected to the normal user main page. |
| Alternative Flows: | <p>AF-S1: If the public user submits with at least one empty input and an unticked box.</p> <ol style="list-style-type: none"> 1. The system will display incomplete messages. 2. Go back to step 1 <p>AF-S2: If public user submit with at least one invalid input and non-empty input</p> <ol style="list-style-type: none"> 1. The system will display invalid messages. 2. Go back to step 1 <p>AF-S3: If user verification fails</p> <ol style="list-style-type: none"> 1. Go back to step 1. |
| Exceptions: | <p>EX-1: If any AF-SX fails or any steps</p> <ol style="list-style-type: none"> 1. The system will display an error screen with the respective error code. |
| Includes: | - |
| Special Requirement: | - |
| Assumptions: | - |
| Notes and Issues: | - |

| | | | |
|--------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|---------------------------|---------------|
| Use Case ID: | UC-03 | | |
| Use Case Name: | View Donate | | |
| Created By: | Sam Jian Shen | Last Updated By: | Sam Jian Shen |
| Date Created: | 09-01-2020 | Date Last Updated: | 03-03-2021 |
| Actor: | Public, Normal | | |
| Description: | It is to showcase all the needs item(s). | | |
| Pre-Conditions: | - | | |
| Post-Conditions: | - | | |
| Priority: | High | | |
| Frequency of Use: | High | | |
| Flow of Events: | <ol style="list-style-type: none"> 1. The system will fetch the donate item information list 2. The system will display donate item information list 3. If the donate item information list is not empty 4. The normal users pressed the 'Give' button | | |

| | |
|-----------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| | <p>5. The normal users will be redirected to give screen</p> <p>6. [UC-06]</p> |
| Alternative Flows: | <p>AF-S1: If the user selects an item type</p> <ol style="list-style-type: none"> 1. The donate item information list will show only the selected item type. 2. Go back to step 3 <p>AF-S2: If the donate item information list is empty</p> <ol style="list-style-type: none"> 1. The system will display 'No Data' in the list 2. Go back to step 2 |
| Exceptions: | <p>EX-1: If any AF-SX fails or any steps</p> <ol style="list-style-type: none"> 1. The system will display an error screen with the respective error code. |
| Includes: | - |
| Special Requirement: | - |
| Assumptions: | - |
| Notes and Issues: | - |

| | | | |
|-----------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|---------------------------|---------------|
| Use Case ID: | UC-04 | | |
| Use Case Name: | View Exchange | | |
| Created By: | Sam Jian Shen | Last Updated By: | Sam Jian Shen |
| Date Created: | 09-01-2020 | Date Last Updated: | 03-03-2021 |
| Actor: | Public, Normal | | |
| Description: | It is to showcase all the exchange item(s). | | |
| Pre-Conditions: | - | | |
| Post-Conditions: | - | | |
| Priority: | High | | |
| Frequency of Use: | High | | |
| Flow of Events: | <ol style="list-style-type: none"> 1. The system will fetch the exchange item information list 2. The system will display an exchange item information list 3. If the exchange item information list is not empty 4. The normal users select one of the exchange items. 5. The normal users will be redirected to the exchange item screen 6. [UC-11] | | |
| Alternative Flows: | <p>AF-S1: If the user selects an item type</p> <ol style="list-style-type: none"> 1. The exchange item information list will show only the selected item type. 2. Go back to step 3 <p>AF-S2: If the exchange item information list is empty</p> <ol style="list-style-type: none"> 1. The system will display 'No Data' in the list 2. Go back to step 2 | | |
| Exceptions: | <p>EX-1: If any AF-SX fails or any steps</p> <ol style="list-style-type: none"> 1. The system will display an error screen with the respective error code. | | |
| Includes: | - | | |
| Special Requirement: | - | | |
| Assumptions: | - | | |

| | |
|--------------------------|---|
| Notes and Issues: | - |
|--------------------------|---|

| | | | |
|-----------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|---------------------------|---------------|
| Use Case ID: | UC-05 | | |
| Use Case Name: | Login | | |
| Created By: | Sam Jian Shen | Last Updated By: | Sam Jian Shen |
| Date Created: | 09-01-2020 | Date Last Updated: | 03-03-2021 |
| Actor: | Normal | | |
| Description: | It is to allow access privileges to respective non-public users. | | |
| Pre-Conditions: | - | | |
| Post-Conditions: | The non-public user will be authenticated and be able to access privileged data respectively. | | |
| Priority: | High | | |
| Frequency of Use: | High | | |
| Flow of Events: | <ol style="list-style-type: none"> 1. The normal user inputs: EMAIL and PASSWORD. 2. The normal user clicks the 'Login' button. 3. The system verifies the inputs and checkbox 4. If verification is successful 5. The normal user will be redirected to the respective assigned role (normal, company, administrator, collector) main page. | | |
| Alternative Flows: | <p>AF-S1: If user verification fails</p> <ol style="list-style-type: none"> 2. The system will display 'Fail to login'. 3. Go back to step 1. | | |
| Exceptions: | <p>EX-1: If any AF-SX fails or any steps</p> <ol style="list-style-type: none"> 1. The system will display an error screen with the respective error code. | | |
| Includes: | - | | |
| Special Requirement: | - | | |
| Assumptions: | - | | |
| Notes and Issues: | - | | |

| | | | |
|--------------------------|------------------------------------------------------------------------------------------------------------------------------------------------|---------------------------|---------------|
| Use Case ID: | UC-06 | | |
| Use Case Name: | Donate | | |
| Created By: | Sam Jian Shen | Last Updated By: | Sam Jian Shen |
| Date Created: | 09-01-2020 | Date Last Updated: | 03-03-2021 |
| Actor: | Normal | | |
| Description: | The user can donate an item and register into the system to be collected by the collector. | | |
| Pre-Conditions: | <p>[UC-05] The user must log in to the system.</p> <p>[UC-10] The user must set their collect location successfully.</p> | | |
| Post-Conditions: | The users donated the item successfully. | | |
| Priority: | High | | |
| Frequency of Use: | High | | |
| Flow of Events: | <ol style="list-style-type: none"> 1. The normal user uploads an image. 2. The normal user clicks the 'Submit' button. | | |

| | |
|-----------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| | <p>3. The system updates the donate lists.</p> <p>4. The system updates the collect item lists.</p> <p>5. The system displays the 'Donate Successfully' message.</p> |
| Alternative Flows: | <p>AF-S1: If the normal user did not upload an image</p> <ol style="list-style-type: none"> 1. The system will display incomplete messages. 2. Go back to step 1 |
| Exceptions: | <p>EX-1: If any AF-SX fails or any steps</p> <ol style="list-style-type: none"> 1. The system will display an error screen with the respective error code. |
| Includes: | UC-10 |
| Special Requirement: | - |
| Assumptions: | - |
| Notes and Issues: | - |

| | | | |
|-----------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|---------------------------|---------------|
| Use Case ID: | UC-07 | | |
| Use Case Name: | Donate Status | | |
| Created By: | Sam Jian Shen | Last Updated By: | Sam Jian Shen |
| Date Created: | 09-01-2020 | Date Last Updated: | 03-03-2021 |
| Actor: | Normal | | |
| Description: | Informative data to keep track of donations or to view donated information. | | |
| Pre-Conditions: | [UC-05] The user must log in to the system. | | |
| Post-Conditions: | - | | |
| Priority: | Medium | | |
| Frequency of Use: | Medium | | |
| Flow of Events: | <ol style="list-style-type: none"> 1. The system will fetch the donate item status information list 2. The system will display the 'Active' donate item status information list. 3. The normal users press the 'Claim' button. 4. The normal users will be redirected to the collector feedback screen. 5. [UC-08] | | |
| Alternative Flows: | <p>AF-S1: If the user clicks the 'Achieve' section.</p> <ol style="list-style-type: none"> 1. The system will display the 'Achieve' donate item status information list. 2. If the user clicks the 'Active' section 3. Go back to step 2. | | |
| Exceptions: | <p>EX-1: If any AF-SX fails or any steps</p> <ol style="list-style-type: none"> 1. The system will display an error screen with the respective error code. | | |
| Includes: | - | | |
| Special Requirement: | - | | |
| Assumptions: | - | | |
| Notes and Issues: | - | | |

| | | | |
|-----------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|---------------------------|---------------|
| Use Case ID: | UC-08 | | |
| Use Case Name: | Collector Feedback | | |
| Created By: | Sam Jian Shen | Last Updated By: | Sam Jian Shen |
| Date Created: | 09-01-2020 | Date Last Updated: | 03-03-2021 |
| Actor: | Normal | | |
| Description: | This provides useful data to improve user experience with the collector or the collector system. | | |
| Pre-Conditions: | [UC-05] The user must log in to the system. [UC-06] The normal users must donate an item. | | |
| Post-Conditions: | The normal users have successfully submitted the collector feedback. | | |
| Priority: | High | | |
| Frequency of Use: | High | | |
| Flow of Events: | <ol style="list-style-type: none"> 1. The normal user can view the donated item name and unique ID 2. The normal user rated the 5-rated heart. 3. The normal user may place remarks. 4. The normal user pressed the 'Claimed Reward' button 5. The system will process the inputs given. 6. The system will display the 'Successfully Feedback' message. 7. The system will be redirected to the claim reward screen 8. [UC-09] | | |
| Alternative Flows: | - | | |
| Exceptions: | EX-1: If any AF-SX fails or any steps <ol style="list-style-type: none"> 1. The system will display an error screen with the respective error code. | | |
| Includes: | - | | |
| Special Requirement: | - | | |
| Assumptions: | - | | |
| Notes and Issues: | - | | |

| | | | |
|--------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|---------------------------|---------------|
| Use Case ID: | UC-09 | | |
| Use Case Name: | Claim Reward | | |
| Created By: | Sam Jian Shen | Last Updated By: | Sam Jian Shen |
| Date Created: | 09-01-2020 | Date Last Updated: | 03-03-2021 |
| Actor: | Normal | | |
| Description: | This provides an incentive for the normal users who stand a chance to win a prize from the points gained from the reward. | | |
| Pre-Conditions: | [UC-05] The user must log in to the system. [UC-08] The user must provide the collector feedback. | | |
| Post-Conditions: | The normal users have successfully claimed the reward. | | |
| Priority: | Low | | |
| Frequency of Use: | Medium | | |
| Flow of Events: | <ol style="list-style-type: none"> 1. The system will randomly generate rewards point 1, 3, 5 points respectively 2. The system will display the rewards point and image represent the reward to the normal users. | | |

| | |
|-----------------------------|--------------------------------------------------------------------------------------------------------------------------------|
| | <p>3. The normal user clicked 'Return'</p> <p>4. The normal users will be redirected to the donate status screen.</p> |
| Alternative Flows: | - |
| Exceptions: | <p>EX-1: If any AF-SX fails or any steps</p> <p>1. The system will display an error screen with the respective error code.</p> |
| Includes: | UC-08 |
| Special Requirement: | - |
| Assumptions: | - |
| Notes and Issues: | - |

| | | | |
|---------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|---------------------------|---------------|
| Use Case ID: | UC-10 | | |
| Use Case Name: | Set Collect Location | | |
| Created By: | Sam Jian Shen | Last Updated By: | Sam Jian Shen |
| Date Created: | 09-01-2020 | Date Last Updated: | 03-03-2021 |
| Actor: | Normal | | |
| Description: | This is to provide a local address for the collector to collect the donated item. | | |
| Pre-Conditions: | [UC-05] The user must log in to the system. | | |
| Post-Conditions: | The user has successfully provided a valid location. | | |
| Priority: | Medium | | |
| Frequency of Use: | High | | |
| Flow of Events: | <p>1. The system checks if the normal user has set at least one collect location.</p> <p>2. The normal users existing set collect location will be fetched from the system.</p> <p>3. The system displays the existing set location.</p> <p>4. The normal pressed 'Submit' button</p> <p>5. The system will verify the inputs</p> <p>6. The system will display 'Successfully Set Location'</p> | | |
| Alternative Flows: | <p>AF-S1: If the normal user has not set at least one collect location.</p> <p>1. The system display blank location information (AVENUE ADDRESS, UNIT NUMBER, POSTAL CODE).</p> <p>2. Go back to step 4</p> <p>AF-S1: If the normal user submits with at least one empty input.</p> <p>3. The system will display incomplete messages.</p> <p>4. Go back to step 4</p> <p>AF-S2: If the normal user submits with at least one invalid input and non-empty input</p> <p>3. The system will display invalid messages.</p> <p>4. Go back to step 4</p> <p>AF-S3: If system verification failed</p> <p>1. Go back to step 1.</p> | | |
| Exceptions: | <p>EX-1: If any AF-SX fails or any steps</p> <p>1. The system will display an error screen with the respective error code.</p> | | |

| | |
|-----------------------------|---|
| Includes: | - |
| Special Requirement: | - |
| Assumptions: | - |
| Notes and Issues: | - |

| | | | |
|---------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|---------------------------|---------------|
| Use Case ID: | UC-11 | | |
| Use Case Name: | Exchange | | |
| Created By: | Sam Jian Shen | Last Updated By: | Sam Jian Shen |
| Date Created: | 09-01-2020 | Date Last Updated: | 03-03-2021 |
| Actor: | Normal | | |
| Description: | The normal user can exchange items and register into the system. | | |
| Pre-Conditions: | [UC-05] The user must log in into the system. [UC-12] The normal users must have preferences set successfully. [UC-13] The normal users must have offered by another normal user. | | |
| Post-Conditions: | The user exchanges the item successfully. | | |
| Priority: | High | | |
| Frequency of Use: | High | | |
| Flow of Events: | <ol style="list-style-type: none"> 1. The normal users require to QR scan or key in the other QR Code value into the system vice versa 2. The normal users pressed the 'Submit' button. 3. If the system verifies successfully 4. The system updates both exchanger information. 5. The system will update exchange status lists. 6. If the system update in step 8 and 9 is successful 7. The system will display the 'Successfully Exchanged' message. | | |
| Alternative Flows: | <p>AF-S1: If the normal user pressed the 'Chat' button.</p> <ol style="list-style-type: none"> 1. The user will be redirected to the chat screen 2. [UC-17] <p>AF-S2: If the normal user pressed the 'Ask' button.</p> <ol style="list-style-type: none"> 1. The user will be redirected to the QA screen 2. [UC-16] <p>AF-S3: If the normal user pressed the 'Exchange Status' button.</p> <ol style="list-style-type: none"> 1. The user will be redirected to the exchange status screen 2. [UC-15] <p>AF-S4: If the normal user submits with an empty QR code.</p> <ol style="list-style-type: none"> 1. The system will display incomplete messages. 2. Go back to step 1 <p>AF-S5: If the normal user submit with wrong format QR code</p> <ol style="list-style-type: none"> 1. The system will display the 'Wrong Format' message. 2. Go back to step 1 <p>AF-S6: If system verify fails</p> <ol style="list-style-type: none"> 1. The system will display invalid messages. 2. Go back to step 1 | | |
| Exceptions: | EX-1: If any AF-SX fails or any steps | | |

| | |
|-----------------------------|----------------------------------------------------------------------------|
| | 1. The system will display an error screen with the respective error code. |
| Includes: | UC-12, UC-13, UC-14 |
| Special Requirement: | - |
| Assumptions: | - |
| Notes and Issues: | - |

| | | | |
|---------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|---------------------------|---------------|
| Use Case ID: | UC-12 | | |
| Use Case Name: | Preferences | | |
| Created By: | Sam Jian Shen | Last Updated By: | Sam Jian Shen |
| Date Created: | 09-01-2020 | Date Last Updated: | 03-03-2021 |
| Actor: | Normal | | |
| Description: | The normal user must provide their preferences to allow other normal users to exchange. | | |
| Pre-Conditions: | [UC-05] The user must log in to the system. | | |
| Post-Conditions: | The normal users have set up their preferences successfully. | | |
| Priority: | Medium | | |
| Frequency of Use: | Low | | |
| Flow of Events: | <ol style="list-style-type: none"> 1. The system will load the existing preferences information. 2. The system will display the preferences information. 3. The system will get the current normal user location. 4. The system will preset the preferred location base on the user location. 5. The normal users will set the inputs for PREFER TIME, PREFER DAY, PREFER LOCATION, PREFER EXCHANGE ITEM. 6. The normal users will press the 'Apply' button 7. The system will verify the inputs 8. The system will update the preferences data of the normal users successfully 9. The system displays the 'Success Update Preferences' message | | |
| Alternative Flows: | <p>AF-S1: If the system does not have existing preferences information.</p> <ol style="list-style-type: none"> 1. The system will load the default preferences information. 2. Go back to step 2. <p>AF-S2: If inputs for PREFER LOCATION or PREFER EXCHANGE ITEM is empty.</p> <ol style="list-style-type: none"> 1. The system will display incomplete messages. 2. Go back to step 2. <p>AF-S3: If verification of inputs fail</p> <ol style="list-style-type: none"> 1. Go back to step 2 | | |
| Exceptions: | <p>EX-1: If any AF-SX fails or any steps.</p> <ol style="list-style-type: none"> 1. The system will display an error screen with the respective error code. <p>EX-2: If the device does not have internet access</p> <ol style="list-style-type: none"> 1. The system will display an error screen with a 'No Internet' message. <p>EX-3: If the device disables permission for current location permission.</p> | | |

| | |
|-----------------------------|---------------------------------------------------------------------------------------|
| | 1. The system will display an error screen with a 'Please Enable Permission' message. |
| Includes: | - |
| Special Requirement: | - |
| Assumptions: | - |
| Notes and Issues: | - |

| | | | |
|---------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|---------------------------|---------------|
| Use Case ID: | UC-13 | | |
| Use Case Name: | Offer | | |
| Created By: | Sam Jian Shen | Last Updated By: | Sam Jian Shen |
| Date Created: | 09-01-2020 | Date Last Updated: | 03-03-2021 |
| Actor: | Normal | | |
| Description: | The normal users can offer exchange items to be approved by other normal users. | | |
| Pre-Conditions: | [UC-05] The user must log in to the system. | | |
| Post-Conditions: | The normal users have successfully and approved by the other normal users | | |
| Priority: | High | | |
| Frequency of Use: | Medium | | |
| Flow of Events: | <ol style="list-style-type: none"> 1. The system will fetch the exchange information from the exchanger (other normal) user. 2. The system will display exchange information and meeting location 3. The normal user will press the 'Add' button 4. The system will redirect the normal user to select the offer screen 5. The system will fetch the normal user's exchange items. 6. The system will display normal user's exchange items. 7. The normal user will select one of the exchange items. 8. The normal user press the 'Add' button 9. The system will fetch selected exchange information. 10. The system will redirect users back to the previous screen 11. The system will display both normal users' exchange information and meeting location. 12. The normal users may input: REMARKS 13. The normal will user press the 'Send Request' button 14. The system will verify the inputs 15. The system will update the exchange information status lists. 16. The other normal users will press the 'Approved' button 17. Both normal users will press the 'Exchange' button 18. [UC-11] | | |
| Alternative Flows: | <p>AF-S1: If verify the inputs fail</p> <ol style="list-style-type: none"> 1. The system will display the 'Please add an exchange item' message. 2. Go back to step 13 | | |
| Exceptions: | <p>EX-1: If any AF-SX fails or any steps</p> <ol style="list-style-type: none"> 1. The system will display an error screen with the respective error code. | | |
| Includes: | - | | |

| | |
|-----------------------------|---|
| Special Requirement: | - |
| Assumptions: | - |
| Notes and Issues: | - |

| | | | |
|-----------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|---------------------------|---------------|
| Use Case ID: | UC-14 | | |
| Use Case Name: | Exchange Feedback | | |
| Created By: | Sam Jian Shen | Last Updated By: | Sam Jian Shen |
| Date Created: | 09-01-2020 | Date Last Updated: | 03-03-2021 |
| Actor: | Normal | | |
| Description: | This is to provide feedback information to the exchange such that other normal users can review. | | |
| Pre-Conditions: | [UC-05] The user must log in to the system. [UC-013] The normal users must approve an offer. | | |
| Post-Conditions: | The users submitted exchange feedback successfully. | | |
| Priority: | Medium | | |
| Frequency of Use: | Medium | | |
| Flow of Events: | <ol style="list-style-type: none"> 1. The normal user can view the exchanged item name and unique ID 2. The normal user may optionally upload an image. 3. The normal user rated the 5-rated heart. 4. The normal user may place remarks. 5. The normal user pressed the 'Submit' button 6. The system will process the inputs given. 7. The system will display the 'Successfully Feedback' message. 8. The system will be redirected to the exchange status screen 9. [UC-11] | | |
| Alternative Flows: | - | | |
| Exceptions: | EX-1: If any AF-SX fails or any steps <ol style="list-style-type: none"> 1. The system will display an error screen with the respective error code. | | |
| Includes: | UC-13 | | |
| Special Requirement: | - | | |
| Assumptions: | - | | |
| Notes and Issues: | - | | |

| | | | |
|-------------------------|-----------------------------------------------------------------------------|---------------------------|---------------|
| Use Case ID: | UC-15 | | |
| Use Case Name: | Exchange Status | | |
| Created By: | Sam Jian Shen | Last Updated By: | Sam Jian Shen |
| Date Created: | 09-01-2020 | Date Last Updated: | 03-03-2021 |
| Actor: | Normal | | |
| Description: | Informative data to keep track of exchange or to view exchange information. | | |
| Pre-Conditions: | [UC-05] The user must log in into the system. | | |
| Post-Conditions: | - | | |
| Priority: | Medium | | |

| | |
|-----------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Frequency of Use: | Medium |
| Flow of Events: | <ol style="list-style-type: none"> 1. The system will fetch the exchange item status information list 2. The system will display the 'Active' exchange item status information list 3. The normal users pressed the 'Exchange' button 4. The normal users will be redirected to the collector verify screen 5. [UC-08] |
| Alternative Flows: | <p>AF-S1: If the normal user clicked the 'Achieve' section.</p> <ol style="list-style-type: none"> 1. The system will display the 'Achieve' exchange item status information list. 2. If the normal user click the 'Active' section 3. Go back to step 2. |
| Exceptions: | <p>EX-1: If any AF-SX fails or any steps</p> <ol style="list-style-type: none"> 1. The system will display an error screen with the respective error code. |
| Includes: | - |
| Special Requirement: | - |
| Assumptions: | - |
| Notes and Issues: | - |

| | | | |
|---------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|---------------------------|---------------|
| Use Case ID: | UC-16 | | |
| Use Case Name: | Q&A | | |
| Created By: | Sam Jian Shen | Last Updated By: | Sam Jian Shen |
| Date Created: | 09-01-2020 | Date Last Updated: | 03-03-2021 |
| Actor: | Normal | | |
| Description: | This provides information about the exchange item to the normal users. | | |
| Pre-Conditions: | [UC-05] The user must log in to the system to ask or answer questions. | | |
| Post-Conditions: | The normal users have successfully asked or answered the questions. | | |
| Priority: | Low | | |
| Frequency of Use: | Low | | |
| Flow of Events: | <ol style="list-style-type: none"> 1. The system will fetch the exchanger username and profile image. 2. The system will display the exchanger username and profile image. 3. The normal user will input: ENQUIRY 4. The normal user will press the 'Send' button 5. The system will update the exchange item's Q&A lists 6. The system will display 'Successfully Send' message 7. The system will redirect the normal user back to the exchange item screen. | | |
| Alternative Flows: | <p>AF-S1: If the exchange item belongs to the normal users</p> <ol style="list-style-type: none"> 1. The ENQUIRY input will create a QUESTION in the Q&A lists 2. Go back to step 4 <p>AF-S2: If the exchange item belongs to the other normal users</p> <ol style="list-style-type: none"> 1. The ENQUIRY input updates an ANSWER in the Q&A lists 2. Go back to step 4 | | |
| Exceptions: | EX-1: If any AF-SX fails or any steps | | |

| | |
|-----------------------------|----------------------------------------------------------------------------|
| | 1. The system will display an error screen with the respective error code. |
| Includes: | - |
| Special Requirement: | - |
| Assumptions: | - |
| Notes and Issues: | - |

| | | | |
|-----------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|---------------------------|---------------|
| Use Case ID: | UC-17 | | |
| Use Case Name: | Chat | | |
| Created By: | Sam Jian Shen | Last Updated By: | Sam Jian Shen |
| Date Created: | 09-01-2020 | Date Last Updated: | 03-03-2021 |
| Actor: | Normal | | |
| Description: | This allows negotiation, assurance, confirmation between normal users. | | |
| Pre-Conditions: | [UC-05] The user must log in to the system. | | |
| Post-Conditions: | The chat messenger will be updated. | | |
| Priority: | Low | | |
| Frequency of Use: | Medium | | |
| Flow of Events: | <ol style="list-style-type: none"> 1. The system will fetch a chat list related to the normal users. 2. The system will display the chat list 3. If the normal user will press select one of the chat lists. 4. The system will fetch the selected chat's message list: USERNAME, MESSAGE TIME list, MESSAGE INFORMATION list. 5. The system will display the selected chat's message list and message box 6. The normal user will input: NEW MESSAGE 7. The normal user will press the 'Send' button 8. The system will update the selected chat's message list. | | |
| Alternative Flows: | <p>AF-S1: If the chat list is empty</p> <ol style="list-style-type: none"> 1. The system will display 'No Data' in the chat list 2. Go back to step 2 <p>AF-S2: If the message list is empty</p> <ol style="list-style-type: none"> 1. The system will display a message box only 2. Go back to step 5 <p>AF-S3: The normal user pressed 'Send' when the message box is empty</p> <ol style="list-style-type: none"> 1. Go back to step 6 <p>AF-S4: The normal user select another chat list</p> <ol style="list-style-type: none"> 1. Go back to step 4 | | |
| Exceptions: | EX-1: If any AF-SX fails or any steps The system will display an error screen with the respective error code. | | |
| Includes: | - | | |
| Special Requirement: | - | | |
| Assumptions: | - | | |
| Notes and Issues: | - | | |

| | |
|---------------------|-------|
| Use Case ID: | UC-18 |
|---------------------|-------|

| | | | |
|-----------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|---------------------------|---------------|
| Use Case Name: | Application Feedback | | |
| Created By: | Sam Jian Shen | Last Updated By: | Sam Jian Shen |
| Date Created: | 09-01-2020 | Date Last Updated: | 03-03-2021 |
| Actor: | Normal | | |
| Description: | This is to provide feedback about using the 'JustGivit' system experience. This allows improvement to the system. | | |
| Pre-Conditions: | [UC-05] The user must log in into the system. | | |
| Post-Conditions: | The normal users have submitted the application feedback successfully. | | |
| Priority: | Low | | |
| Frequency of Use: | Low | | |
| Flow of Events: | <ol style="list-style-type: none"> 1. The normal user rated 5-rated hearts 2. The normal user will press the 'Submit' button. 3. The system will update the application feedback list. 4. The system will display 'Success Submitted Application Feedback' 5. The system will redirect the normal user to the main screen. | | |
| Alternative Flows: | <p>AF-S1: The normal user inputs: FEEDBACK</p> <ol style="list-style-type: none"> 1. Go back to step 2 | | |
| Exceptions: | <p>EX-1: If any AF-SX fails or any steps</p> <ol style="list-style-type: none"> 1. The system will display an error screen with the respective error code. | | |
| Includes: | - | | |
| Special Requirement: | - | | |
| Assumptions: | - | | |
| Notes and Issues: | - | | |

| | | | |
|-----------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|---------------------------|---------------|
| Use Case ID: | UC-19 | | |
| Use Case Name: | View Needs | | |
| Created By: | Sam Jian Shen | Last Updated By: | Sam Jian Shen |
| Date Created: | 09-01-2020 | Date Last Updated: | 03-03-2021 |
| Actor: | Company | | |
| Description: | This allows the company to keep track of their needs. | | |
| Pre-Conditions: | [UC-05] The user must log in to the system. | | |
| Post-Conditions: | - | | |
| Priority: | Medium | | |
| Frequency of Use: | Medium | | |
| Flow of Events: | <ol style="list-style-type: none"> 1. The system will fetch company need lists 2. The system will display company need lists | | |
| Alternative Flows: | <p>AF-S1: If the company need lists are empty</p> <ol style="list-style-type: none"> 1. The system will display 'No Data' in company need lists 2. Go back to step 2 | | |
| | <p>EX-1: If any AF-SX fails or any steps</p> <ol style="list-style-type: none"> 1. The system will display an error screen with the respective error code. | | |
| Includes: | - | | |
| Special Requirement: | - | | |

| | |
|--------------------------|---|
| Assumptions: | - |
| Notes and Issues: | - |

| | | | |
|-----------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|---------------------------|---------------|
| Use Case ID: | UC-20 | | |
| Use Case Name: | Add Needs | | |
| Created By: | Sam Jian Shen | Last Updated By: | Sam Jian Shen |
| Date Created: | 09-01-2020 | Date Last Updated: | 03-03-2021 |
| Actor: | Company | | |
| Description: | This allows the company to register needs into the system upon approval. | | |
| Pre-Conditions: | [UC-05] The user must log in to the system. | | |
| Post-Conditions: | The company user has successfully registered the needs into the system pending approval from the Administrator. | | |
| Priority: | Medium | | |
| Frequency of Use: | Medium | | |
| Flow of Events: | <ol style="list-style-type: none"> 1. The company user select one of the item types 2. The company user input: ITEM NAME 3. The company user set quantity value 4. The company user press the 'Add Item' button 5. The system will verify the inputs from the company user 6. The system will prompt the company user for confirmation with 2 options 'Yes' or 'No' 7. If the company user clicks 'Yes' 8. The system will update the company need lists 9. The system will display 'Success Update Need' and updated need lists. | | |
| Alternative Flows: | <p>AF-S1: If the company user click 'No' at the prompt</p> <ol style="list-style-type: none"> 1. Go back to step 4 <p>AF-S2: If company user input at least one empty</p> <ol style="list-style-type: none"> 1. The system will display an incomplete message. 2. Go back to step 1 <p>AF-S3: If company user input at least one invalid</p> <ol style="list-style-type: none"> 1. The system will display an invalid message 2. Go back to step 1 | | |
| Exceptions: | <p>EX-1: If any AF-SX fails or any steps</p> <ol style="list-style-type: none"> 1. The system will display an error screen with the respective error code. | | |
| Includes: | - | | |
| Special Requirement: | - | | |
| Assumptions: | - | | |
| Notes and Issues: | - | | |

| | | | |
|-----------------------|----------------|---------------------------|---------------|
| Use Case ID: | UC-21 | | |
| Use Case Name: | Manage Company | | |
| Created By: | Sam Jian Shen | Last Updated By: | Sam Jian Shen |
| Date Created: | 09-01-2020 | Date Last Updated: | 03-03-2021 |

| | |
|-----------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Actor: | Administrator |
| Description: | This allows a valid company to be accepted into the system. |
| Pre-Conditions: | [UC-05] The user must log in to the system via Web. |
| Post-Conditions: | The unapproved company lists will be updated successfully. |
| Priority: | Low |
| Frequency of Use: | Low |
| Flow of Events: | <ol style="list-style-type: none"> 1. The system will fetch the company approval list 2. The system will display the company approval list 3. The administrator pressed 'Accept' on the selected company 4. The system will remove the selected company from the company approval list. 5. The system will add the selected company to the company list. 6. The system will display the 'Success' message and display the updated company approval list. |
| Alternative Flows: | <p>AF-S1: If the administrator press 'Reject' on the selected company</p> <ol style="list-style-type: none"> 1. The system will remove the selected company from the company approval list. 2. The system will display the updated company approval list. 3. Go back to step 2 |
| Exceptions: | <p>EX-1: If any AF-SX fails or any steps</p> <ol style="list-style-type: none"> 1. The system will display an error screen with the respective error code. |
| Includes: | - |
| Special Requirement: | - |
| Assumptions: | - |
| Notes and Issues: | - |

| | | | |
|--------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|---------------------------|---------------|
| Use Case ID: | UC-22 | | |
| Use Case Name: | Manage Collector | | |
| Created By: | Sam Jian Shen | Last Updated By: | Sam Jian Shen |
| Date Created: | 09-01-2020 | Date Last Updated: | 03-03-2021 |
| Actor: | Administrator | | |
| Description: | This allows a collector account to be created and be used by them. | | |
| Pre-Conditions: | [UC-05] The user must log in to the system via Web. | | |
| Post-Conditions: | The collector lists will be updated successfully. | | |
| Priority: | Low | | |
| Frequency of Use: | Low | | |
| Flow of Events: | <ol style="list-style-type: none"> 1. The system will fetch collector lists 2. The system will display collector lists 3. The administrator will inputs: USERNAME, FULLNAME, CONTACT NUM, ADDRESS AVENUE, UNIT NUMBER, POSTAL CODE 4. The administrator upload an image 5. If the administrator pressed 'Add' 6. The system will transfer the input and update the collector lists 7. The system displays the updated collector lists successfully 8. The administrator selects one of the collector lists | | |

| | |
|-----------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| | <p>9. If the administrator presses the 'Delete' button</p> <p>10. The system will remove the selected collector from the collector list.</p> <p>11. The system displays the updated collector lists successfully.</p> |
| Alternative Flows: | <p>AF-S1: If the inputs at least one is empty or using the default image</p> <ol style="list-style-type: none"> 1. The system will display an incomplete message. 2. Go back to step 2 <p>AF-S2: If the administrator presses the 'Delete' button without a selected collector.</p> <ol style="list-style-type: none"> 1. Go back to step 7 |
| Exceptions: | <p>EX-1: If any AF-SX fails or any steps</p> <ol style="list-style-type: none"> 1. The system will display an error screen with the respective error code. |
| Includes: | - |
| Special Requirement: | - |
| Assumptions: | - |
| Notes and Issues: | - |

| | | | |
|-----------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|---------------------------|---------------|
| Use Case ID: | UC-23 | | |
| Use Case Name: | Manage Donor | | |
| Created By: | Sam Jian Shen | Last Updated By: | Sam Jian Shen |
| Date Created: | 09-01-2020 | Date Last Updated: | 03-03-2021 |
| Actor: | Administrator | | |
| Description: | This allows the administrator to view and do basic manual filter tasks for donor uploaded images to minimize misuse. | | |
| Pre-Conditions: | [UC-05] The user must log in to the system via Web. | | |
| Post-Conditions: | The donor images list will be updated successfully. | | |
| Priority: | Low | | |
| Frequency of Use: | Low | | |
| Flow of Events: | <ol style="list-style-type: none"> 1. The system will fetch all normal users donated images list. 2. The system will display all normal users donated images list. 3. The administrator selects one of the donated images 4. The administrator press the 'Delete' button 5. The system will remove the selected donated images from the list 6. The system will display the updated donated images list | | |
| Alternative Flows: | <p>AF-S1: If the administrator presses the 'Delete' button without selected the donated image.</p> <ol style="list-style-type: none"> 1. Go back to step 2 | | |
| Exceptions: | <p>EX-1: If any AF-SX fails or any steps</p> <ol style="list-style-type: none"> 1. The system will display an error screen with the respective error code. | | |
| Includes: | - | | |
| Special Requirement: | - | | |
| Assumptions: | - | | |
| Notes and Issues: | - | | |

| | | | |
|-----------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|---------------------------|---------------|
| Use Case ID: | UC-24 | | |
| Use Case Name: | Manage Needs | | |
| Created By: | Sam Jian Shen | Last Updated By: | Sam Jian Shen |
| Date Created: | 09-01-2020 | Date Last Updated: | 03-03-2021 |
| Actor: | Administrator | | |
| Description: | This allows intermediate control of needs lists to minimize misuse. | | |
| Pre-Conditions: | [UC-05] The user must log in to the system via Web. | | |
| Post-Conditions: | The needs list will be updated successfully. | | |
| Priority: | Low | | |
| Frequency of Use: | Medium | | |
| Flow of Events: | <ol style="list-style-type: none"> 1. The system will fetch needs approval list 2. The system will display needs approval list 3. The administrator pressed 'Accept' on selected needs 4. The system will remove selected needs from the needs approval list. 5. The system will add selected needs to the needs list. 6. The system will display the 'Success' message and display an updated needs approval list. | | |
| Alternative Flows: | <p>AF-S1: If the administrator press 'Reject' on selected needs</p> <ol style="list-style-type: none"> 1. The system will remove selected needs from the needs approval list. 2. The system will display an updated needs approval list. 3. Go back to step 2 | | |
| Exceptions: | <p>EX-1: If any AF-SX fails or any steps</p> <ol style="list-style-type: none"> 1. The system will display an error screen with the respective error code. | | |
| Includes: | - | | |
| Special Requirement: | - | | |
| Assumptions: | - | | |
| Notes and Issues: | - | | |

| | | | |
|--------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|---------------------------|---------------|
| Use Case ID: | UC-25 | | |
| Use Case Name: | Manage Application Feedback | | |
| Created By: | Sam Jian Shen | Last Updated By: | Sam Jian Shen |
| Date Created: | 09-01-2020 | Date Last Updated: | 03-03-2021 |
| Actor: | Administrator | | |
| Description: | This is to publish the positive outcome of this system to the public. This also helps to gain public trust in this system. | | |
| Pre-Conditions: | [UC-05] The user must log in to the system via Web. | | |
| Post-Conditions: | The application feedback list will be updated successfully. | | |
| Priority: | Low | | |
| Frequency of Use: | Medium | | |
| Flow of Events: | <ol style="list-style-type: none"> 1. The system will fetch a feedback application list 2. The system will display a feedback application list 3. The administrator pressed 'Accept' on selected needs | | |

| | |
|-----------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| | <ol style="list-style-type: none"> 4. The system will copy the selected feedback to archive a feedback application list. 5. The system will remove selected feedback from the feedback application list. 6. The system will add selected needs to the public feedback application list. 7. The system will display the 'Success' message and display an updated needs feedback application list. |
| Alternative Flows: | <p>AF-S1: If the administrator press 'Reject' on selected feedback</p> <ol style="list-style-type: none"> 1. The system will copy the selected feedback to achieve a feedback application list. 2. The system will remove the selected feedback application from the feedback application list. 3. The system will display an updated feedback application list. 4. Go back to step 2 |
| Exceptions: | <p>EX-1: If any AF-SX fails or any steps</p> <ol style="list-style-type: none"> 1. The system will display an error screen with the respective error code. |
| Includes: | - |
| Special Requirement: | - |
| Assumptions: | - |
| Notes and Issues: | - |

| | | | |
|-----------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|---------------------------|---------------|
| Use Case ID: | UC-26 | | |
| Use Case Name: | View Singapore Map | | |
| Created By: | Sam Jian Shen | Last Updated By: | Sam Jian Shen |
| Date Created: | 09-01-2020 | Date Last Updated: | 03-03-2021 |
| Actor: | Administrator | | |
| Description: | This is to analyse the overall normal users' non-sensitive data of the system | | |
| Pre-Conditions: | [UC-05] The user must log in into the system via Web. | | |
| Post-Conditions: | - | | |
| Priority: | Low | | |
| Frequency of Use: | Medium | | |
| Flow of Events: | <ol style="list-style-type: none"> 1. The system will fetch a pin-point list 2. The system will call Google API to generate the Singapore Map 3. The system will display the pin-point icons onto the Singapore Map 4. The administrator hover over the pin-point icons 5. The system will display popup information about the hovered pin-point. | | |
| Alternative Flows: | - | | |
| Exceptions: | <p>EX-1: If any AF-SX fails or any steps</p> <ol style="list-style-type: none"> 1. The system will display an error screen with the respective error code. | | |
| Includes: | - | | |
| Special Requirement: | - | | |
| Assumptions: | - | | |

| | |
|--------------------------|---|
| Notes and Issues: | - |
|--------------------------|---|

| | | | |
|-----------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|---------------------------|---------------|
| Use Case ID: | UC-27 | | |
| Use Case Name: | Capture Collect | | |
| Created By: | Sam Jian Shen | Last Updated By: | Sam Jian Shen |
| Date Created: | 09-01-2020 | Date Last Updated: | 03-03-2021 |
| Actor: | Collector | | |
| Description: | This is to prove that item has been collected. | | |
| Pre-Conditions: | [UC-05] The user must log in to the system via an Android device. | | |
| Post-Conditions: | The collector has captured the image successfully. | | |
| Priority: | Medium | | |
| Frequency of Use: | High | | |
| Flow of Events: | <ol style="list-style-type: none"> 1. The system will check for Camera permission 2. If Camera permission is enabled 3. The system will render the camera view to the display screen 4. The collector pressed the 'Capture' button 5. The system will be saved the image into donated capture lists with the unique ID from the donated item. 6. The system will display the 'Successfully Capture' message. | | |
| Alternative Flows: | - | | |
| Exceptions: | EX-1: If any AF-SX fails or any steps <ol style="list-style-type: none"> 1. The system will display an error screen with the respective error code. EX-2: If the android device does not have internet access <ol style="list-style-type: none"> 1. The system will display an error screen with a 'No Internet' message. EX-3: If the android device disables permission for Camera or Folder/File permission. <ol style="list-style-type: none"> 1. The system will display an error screen with a 'Please Enabled Permission' message. | | |
| Includes: | - | | |
| Special Requirement: | - | | |
| Assumptions: | - | | |
| Notes and Issues: | - | | |

| | | | |
|-------------------------|----------------------------------------------------------------------------------------------|---------------------------|---------------|
| Use Case ID: | UC-28 | | |
| Use Case Name: | Assign Collect Item | | |
| Created By: | Sam Jian Shen | Last Updated By: | Sam Jian Shen |
| Date Created: | 09-01-2020 | Date Last Updated: | 03-03-2021 |
| Actor: | Collector | | |
| Description: | This is to assign which item to collect and provide routing information to collect the item. | | |
| Pre-Conditions: | [UC-05] The user must log in to the system via an Android device. | | |
| Post-Conditions: | The Singapore Map will update the routing successfully. | | |

| | |
|-----------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| | The address of the destination will be updated successfully. |
| Priority: | Medium |
| Frequency of Use: | High |
| Flow of Events: | <ol style="list-style-type: none"> 1. The system will fetch the collect list 2. The system will call Google API to generate the Singapore Map 3. The system will get the current collector ‘geopoint’ location 4. The system will display pinpoint icons (blue) onto the Singapore Map with a collect list 5. The system will display pinpoint icons (red) from the current user. 6. The collector will select one of the pin-point icons (blue) 7. The system will calculate the best routing path. 8. The system will get collect item addresses from the collection list. 9. The system will display the collect item address. 10. The system will display the best routing path from the source point to the destination point. |
| Alternative Flows: | <p>AF-S1: If the user re-select another pin-point icons (blue)</p> <ol style="list-style-type: none"> 1. Go to step 7 |
| Exceptions: | <p>EX-1: If any AF-SX fails or any steps</p> <ol style="list-style-type: none"> 1. The system will display an error screen with the respective error code. <p>EX-2: If the android device does not have internet access</p> <ol style="list-style-type: none"> 2. The system will display an error screen with a ‘No Internet’ message. <p>EX-3: If the android device disables permission for Camera or Folder/File permission.</p> <ol style="list-style-type: none"> 1. The system will display an error screen with a ‘Please Enabled Permission’ message. |
| Includes: | - |
| Special Requirement: | - |
| Assumptions: | - |
| Notes and Issues: | - |

Design Specification

System Design

System-Level

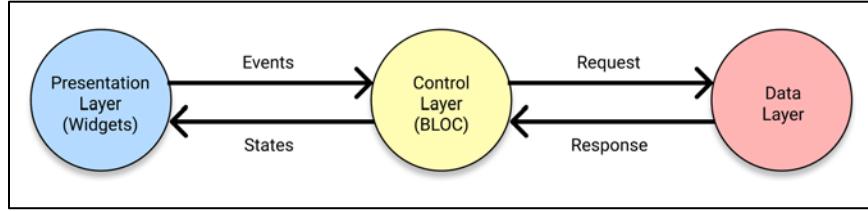


Fig 12, the BLOC Software architecture design

In Flutter world, the problem arises when there exists this API function called “`setStates()`” [33] that rebuilds your entire screen when there is a change to the logic. For example, if the screens consist of an expandable list with a bunch of animations, composed elements, and information from external API at the same time, rebuild all of them would be a performance killer and not feasible to continue development at this stage. Thus, ensuring scalability while maintaining relative performance is important for our project. This motivates me to use another approach for state management architecture called Business Logic Component (BLOC) [34]. I choose this due to it being resourceful and easier for beginners to start with. As shown in Fig 12, it consists of 3 layers namely, presentation, control, and data with intermediate operations.

With this approach, apart from better readability by separate logic and UI code it also allows rendering of selective elements instead of the entire screen's base on user actions which greatly improve the performance when scaling up.

Presentation Layer

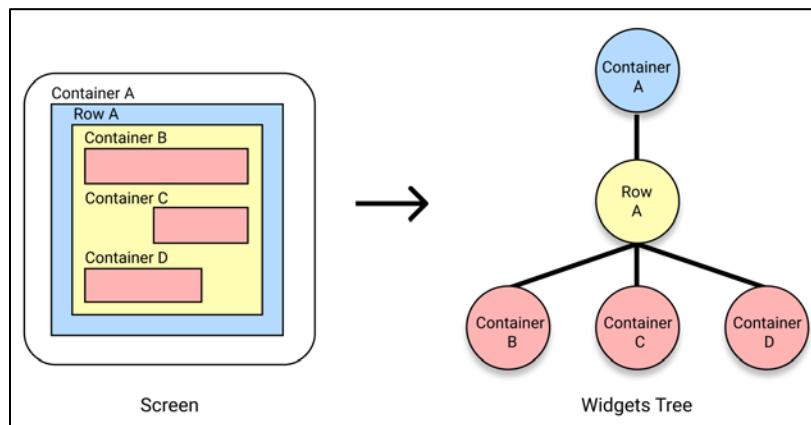


Fig 13, a relation between screen and widgets tree

This layer represents the user interface, Flutter's framework compose of widgets [35] in a tree-like structure with root and leaf nodes represent as widgets as illustrated in Fig 13, the widget itself can be classified as a button, a view list, or the layouts, etc, each widget consists of different parameters. For example, a container can have different widths, heights, relative positions. The tree structure is dependent on how the developer would want to design them.

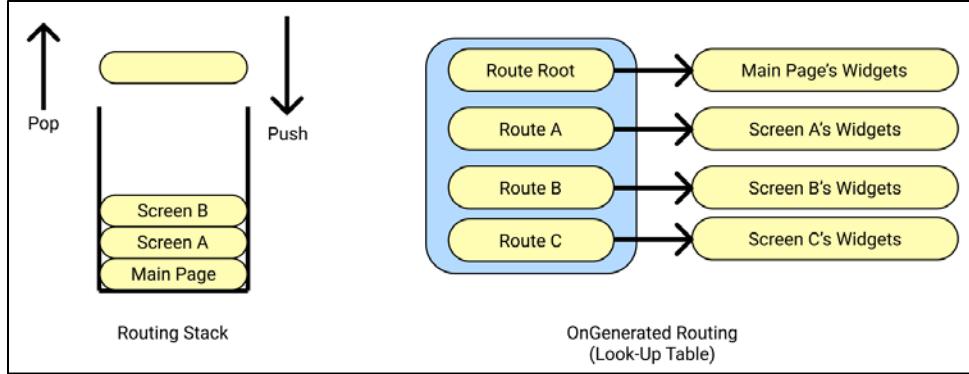


Fig 14, routing mechanism

To navigate different screens in an organized manner. 2 different routing instances keep track of the screens accessed by the users as shown in Fig 14. First of all, the routing stack is responsible to keep all the routing history in a stack. Newly accessed pages are pushed into the top of the stack while going back to the previously accessed screen will pop-top of the stack. The stack must initiate with a root stack which is by default backslash '/'. Secondly, Flutter's OnGenerated routing [36] is responsible to identify which route to navigate and which widgets tree to be accessed, it is a look-up table with unique routing names to access the selected screen respectively. For example, accessing the "about us" screen, the routing name can be "/aboutus", this string is similar to a web URL (domain name) where it is used to substitute the IP address for more readability.

Control Layer

This layer consists of several mechanisms. Firstly, the intermediate operation between the control and presentation layer is Events and States. Secondly, the intermediate operation between the control and data layer is Request and Response. Lastly, the BLOC internal operation and its features.

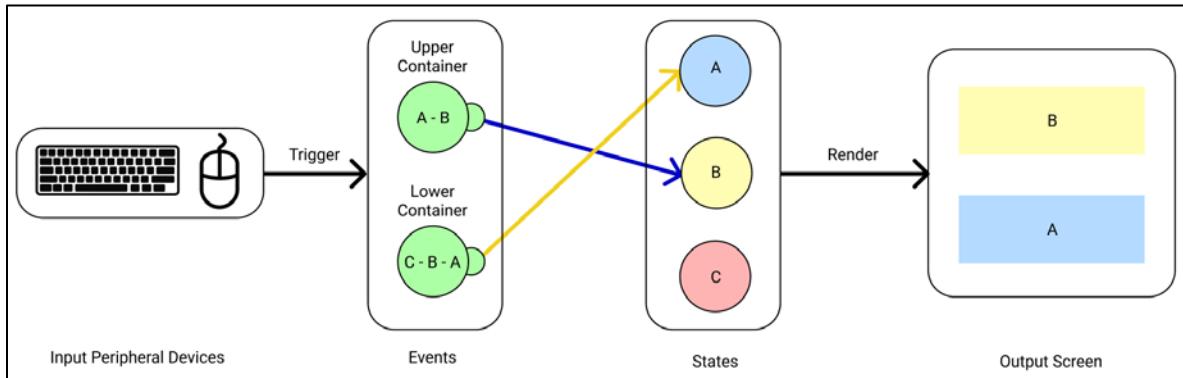


Fig 15, event-driven concept

The BLOC architecture was event-driven where it listens for events triggered by users' input such as mouse or keyboard pressed on the screen [37]. For example, when the user login and press submit button, this will trigger a login event. An event can consist of a sequence of states as shown in Fig 15, at any one point the event only points to 1 state at a time. For example, the upper container consists of 2 states, state A then follow by state B. The system will render the widgets based on the sequence of states. In practice, the states should represent different sets of widgets that serve different purposes.

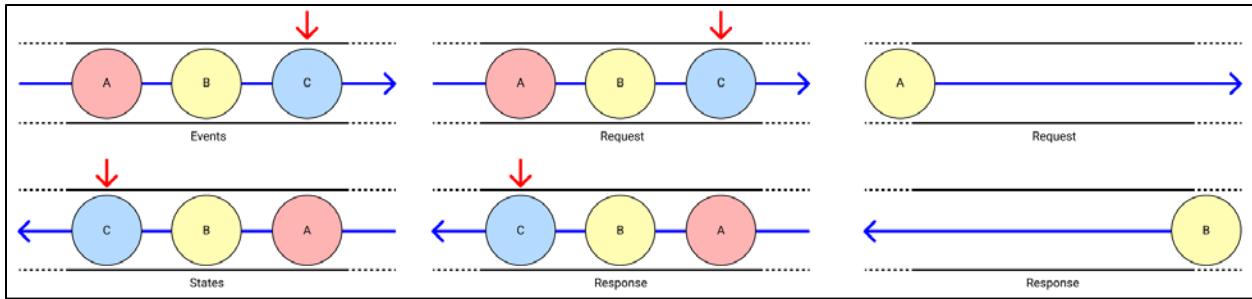


Fig 16, data flow in intermediate operations

To ensure data flow in an asynchronous manner especially when external API takes inconsistent time also known as Future to return the requested data by waiting. This motivated me to use the stream mechanism, a sequence of asynchronous events where it sequentially processes the data. For events and states is it compulsory to use the stream as this is part of the BLOC mechanism. Apart from that, the request and response of data can be either in the stream or fetch and retrieve format.

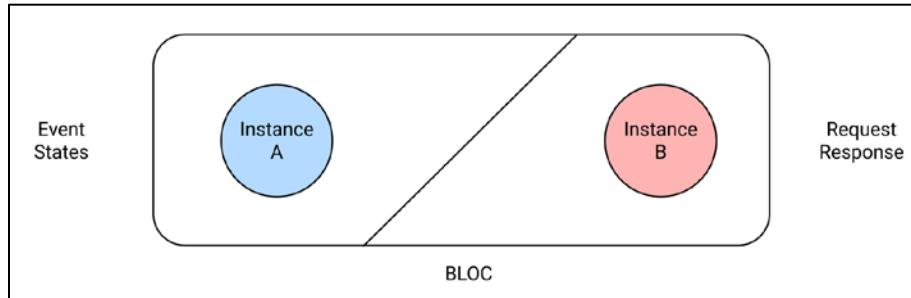


Fig 17, internal of BLOC instance

This is the main part that illustrates the separation between logic and UI as shown in Fig 17. Business Logics are live in here. For each of the BLOC instances, it holds 2 separate sub instances that “indirectly communicate” one another. For example, when the user presses the “Enter” button, the input data was parse into the event and place into a logic function’s parameter for further processing and may ‘await’ for data to return, meanwhile the states was updated to ‘loading’ state, the user will be shown a loading screen until data has retrieved, the states will be updated to ‘done’ state with the retrieve data, eventually display the expected output with the retrieve data back to the user.

BLOC (flutter_bloc) package [38] comes with plenty of features, below are the key features that were used for this project.

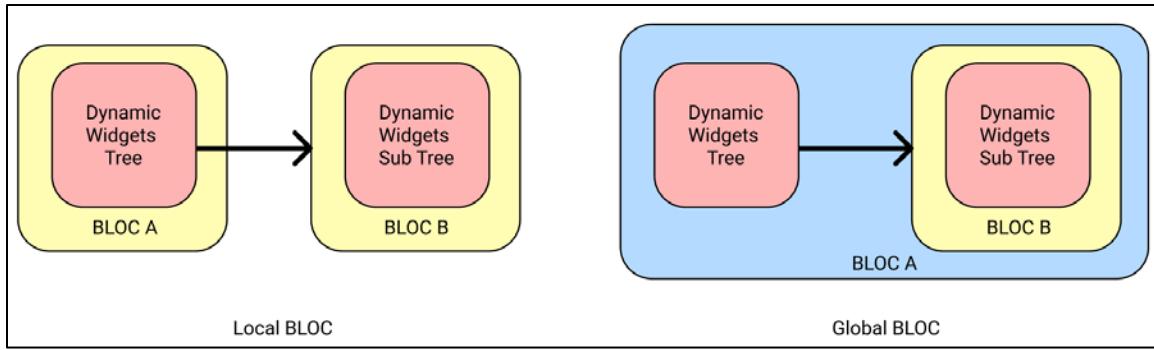


Fig 18, local and global feature of BLOC

The local BLOC is used when the BLOC's states are applied on a single screen. This is useful when we want to perform a specific task in different widgets. For example, we would not want to mix an administrator's BLOC with normal's BLOC as the normal user does not have access rights to the administrator.

To propagate the states to another screen, the BLOC's states must provide its value to its child screen this is known as global BLOC. Typically, the global BLOC is declared in `MaterialApp()` which automatically inherits the state to the entire application without the need to declare parsing the value. For example, in our project, the status of the user's authentication such as their role must be persistent throughout the application.

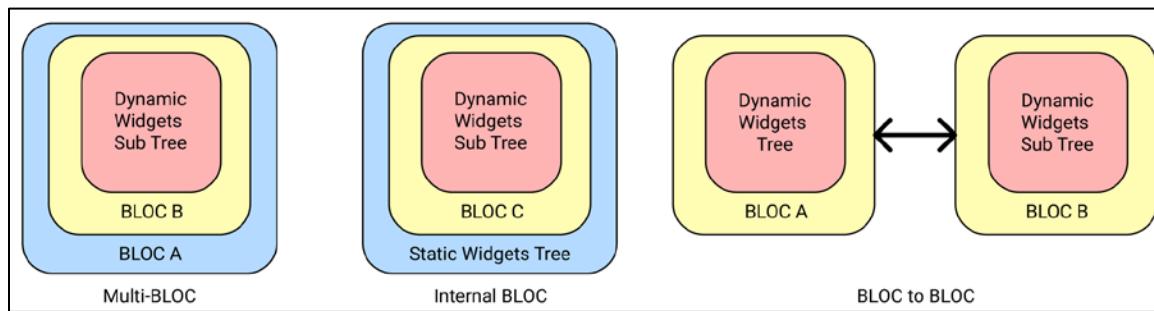


Fig 19, Multi BLOC, Internal BLOC, BLOC to BLOC Communication features

The Multi-BLOC helps provide multiple different logics on a single page. For example, a screen may contain the "login" button and "add" donate item button. These two logics are independent of one another thus 2 different BLOC are used.

The BLOC can also be declared within the widgets called internal BLOC, as shown where the parent of BLOC widgets are static widgets and the child of BLOC widgets are dynamic widgets. This means that static widgets are stateless widgets while dynamic widgets are stateful widgets. For example, the stateless widget can be a background of the screen that does not change its properties or behavior, stateful widgets can be a real-time list that reflects the changes through updating it.

The BLOC can also be dependent on the other BLOC, this is known as BLOC to BLOC communication. For example, users have to key in all the inputs correctly to press the "Submit" button.

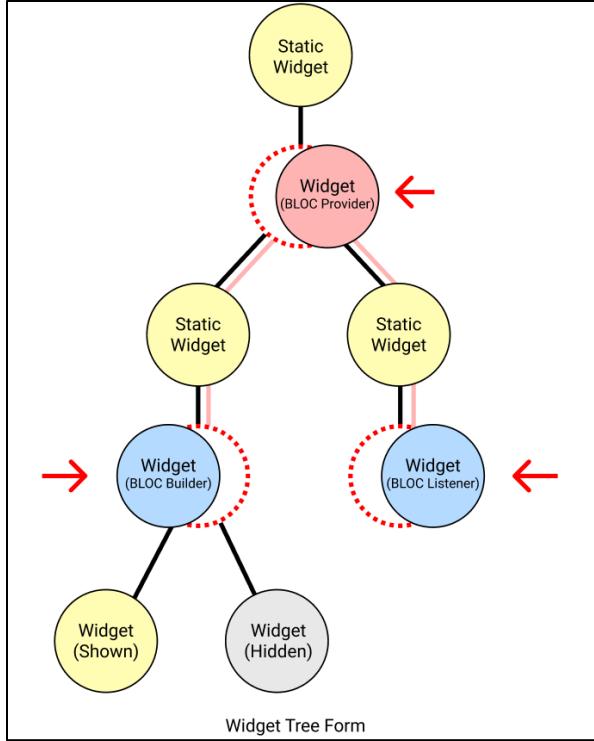


Fig 20, the BLOC declaration on front-end

The above Fig 20 shows the pattern of how a typical BLOC is being used. Firstly, to use a BLOC it has to declare a BLOC's provider such that the child widgets can use it. Secondly, to build the stateful widgets, the BLOC builder is used to "control" the states which widgets to be visible to the user. Optionally, the BLOC can have a listener as shown in Fig 20, since the BLOC builder and BLOC listener is the parent of the same BLOC provider it can make changes based on their states. For example, the listener can show a notify message, this message can produce differently based on which button the user has pressed.

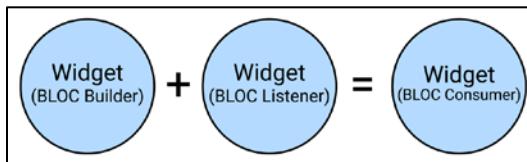


Fig 21, BLOC consumer feature

For better readability and better productivity, sometimes both BLOC listener and BLOC builder are commonly used together. Thus BLOC consumer is introduced.

Data Layer

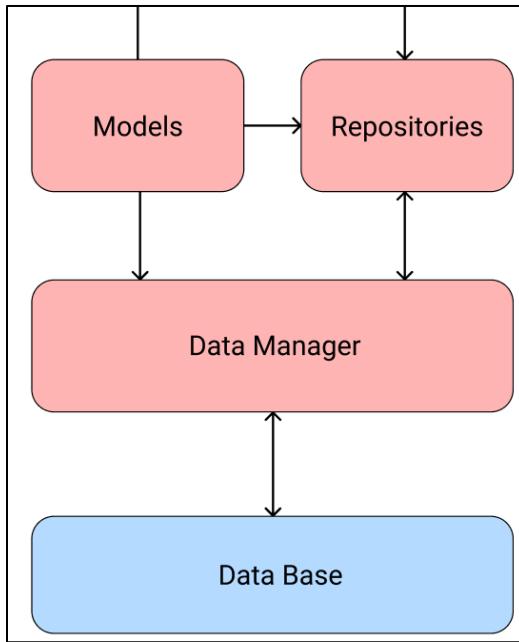


Fig 22, data layer components

In this layer, we have multiple components:

1. The data models use abstraction to represent and better understand the nature of data flow within this application. For example, donor entities must always have a string type username.
2. The repositories are responsible for fetching or sending data or can be used for cache data. For example, fetching a donated item list will call repositories to fetch it.

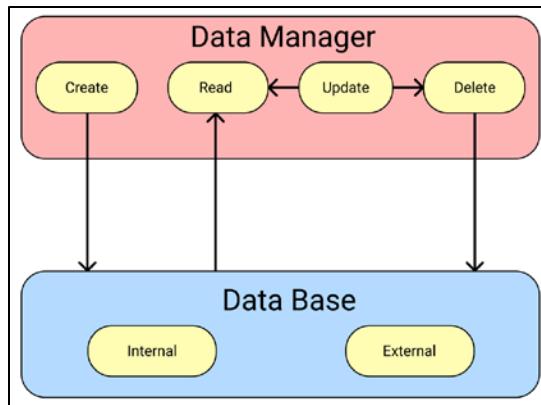


Fig 23, CRUD operations

3. The data manager is a domain base CRUD operations [39] from the database which is our Firestore and Firestorage. For example, fetching Firestore related data are classified into one 'text' manager.
4. The database can be an external API such as firebase or assets from workspace directories.

Software-Level

In this section, we will be looking at some of the key software patterns used in our project as references from G. Erich's book [40].

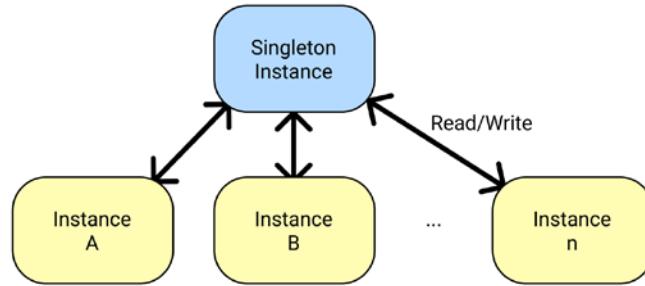


Fig 24, the Singleton Pattern

We use a singleton pattern to allow single instances to persist throughout the life cycle of the application such as the sessions mechanism. For example, sessions' instances cached user inputs after exiting the application.

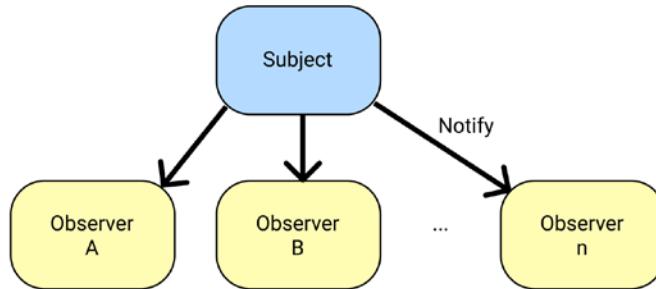


Fig 25, the Observer Pattern

We use observer patterns as a way to keep monitor certain changes to the data or behavior of the subject. For example, BLOC's listener.

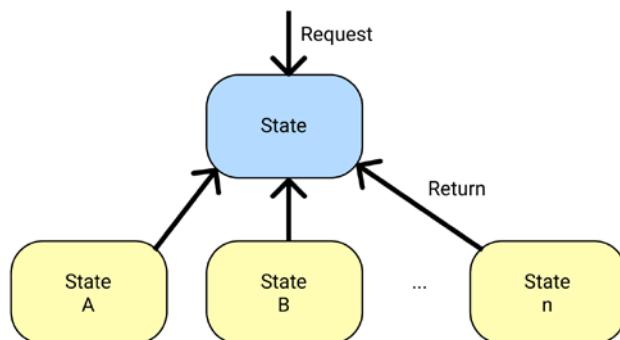


Fig 26, the State Pattern

State pattern can be seen from the BLOC mechanism, this pattern helps us to render different user interface elements to the user such that it is responsive and easier to keep track of faults or bugs as well.

Database Design

For our database, we are using Google Firebase's cloud services [41] such as Authentication, Firestore, and Firestorage. Each database has a different data storage structure.

Firestore

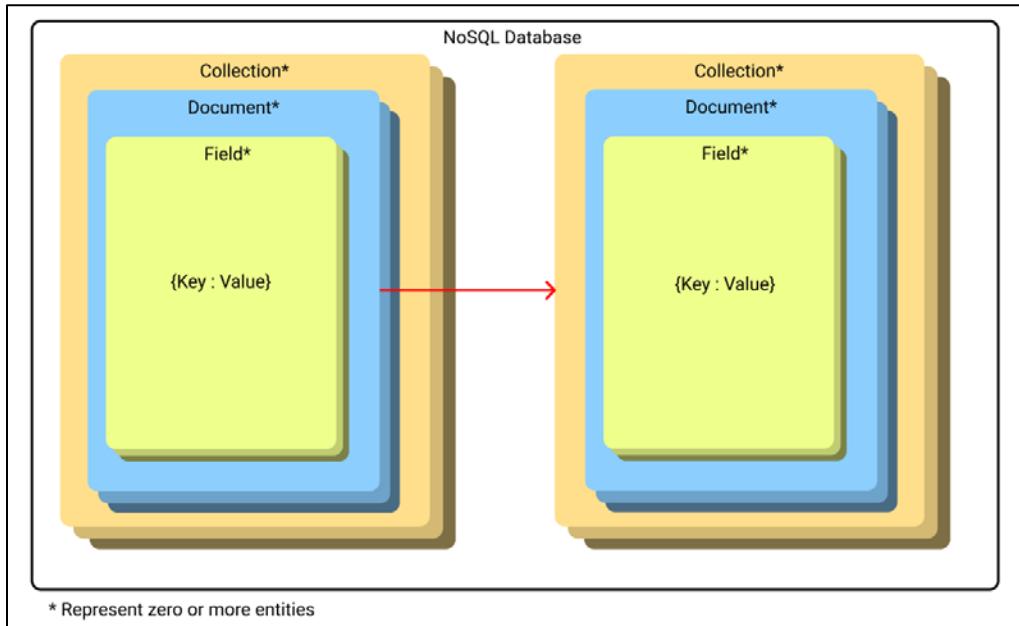


Fig 27, NoSQL Database Concepts

It takes the form of a cloud-based NoSQL database server that stores and syncing data in real-time listeners across multiple client applications. It stores data as documents that are logically classified into collections, the Firestore document offers support for multiple file types such from string to nested objects. Each of the documents can consist of zero or more fields in the key-value pairing. The keys represent the identification for the content which are the values. It also supports indexed queries which means that users can sort, combine, filter functions in a single query. Documents and collections use ACID (Atomicity, Consistency, Isolation, and Durability) transaction, this helps to maintain consistency in the database [42].

In our context, we are using the Firestore to store various users' static data and dynamic data for real-time updates. Choosing NoSQL is less costly as compared to RelationalSQL as it is flexible for redesign and easily scalable as this project is expected to further restructure and improve in the future implementation.

Firestorage

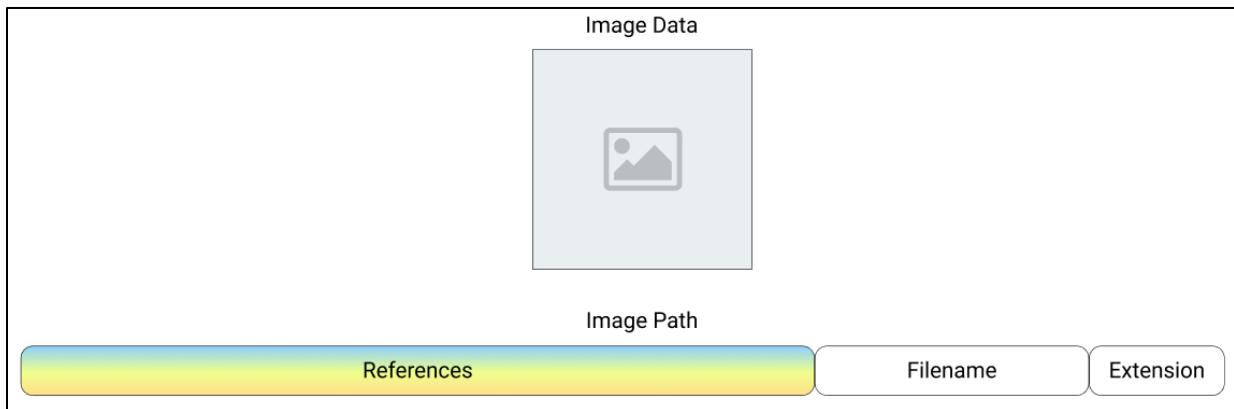


Fig 28, Firestorage mechanism

Firestorage is simply a directory that usually use to store audio, video, or images. In our context, we will use storage to store various user images only. It requires 2 components as follow:

- **Image Data**
 - It stores RGB image data in JPG format with varies in terms of dimension.
- **Image Path**
 - It acts as a pointer to retrieve the images in a string data type. It consists of the references, filename, and the extension which is in JPG format

Authentication

It is used to authenticate with their email addresses and passwords. It also provides methods to create and manage users that use their email addresses and passwords to sign in. It also handles sending password reset emails and so forth. It also supports other form authentication such as Google, Apple, Facebook, Twitter, and Github to login. In our context, we just use basic email and password-based authentication.

UI Design

Consideration

Users Perspective Goals

Each user has a different focus when comes to designing the UI.

- Normal User
 - Simple to use: Minimize the number of actions to achieve the goal.
 - Smooth unique design: Most interface shape edges are curved.
- Company and Administrator User
 - Easy access: All possible functions should be available on a Single page as much as possible without clustering.
- Collector User
 - Frequency use functions: All highly use functions are centric around the user's hand since it is only available on mobile.

Shneiderman's 8-Golden Rules

These rules ensure the design satisfies, productive and frustration-free UI [43]. Below are some of the highlights that support these rules.

1. Consistency



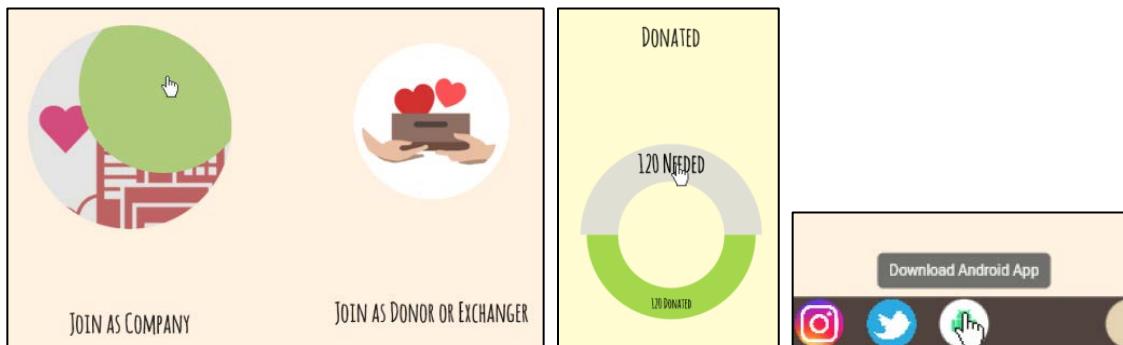
- Most of the screens consist of a header and footer and the color of the screen is used within a theme of colors.

2. Shortcut



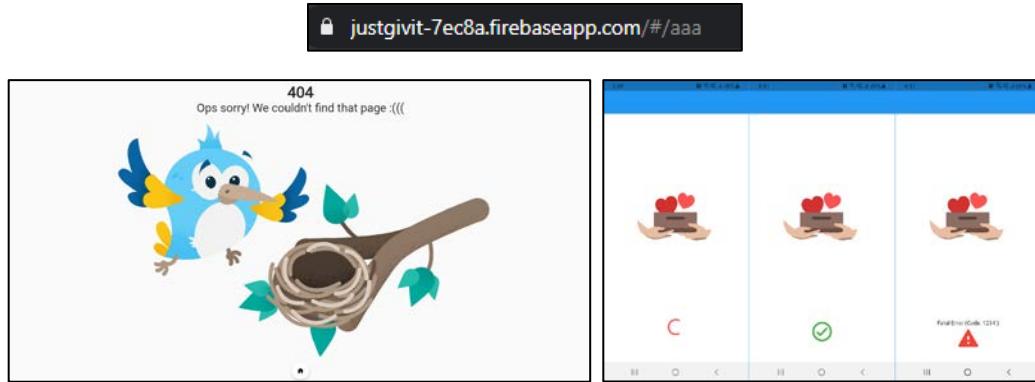
- These are the some of commonly used features such as going back to the main page, accessing the status of the exchange, or donate screen. These are placed in a header for easy access as well

3. Information Feedback



- This to ensure the user is in control at all times. For example, hovering will change in element size, color, shape, and may fade in informative information which describes the element.

4. Dialogue



- For example, if a user accesses an unavailable screen will return a 404-error page {13}, this tells the user this page is non-existent. Besides, during processing or after processing after clicking certain functions, the user may receive a different state of UI to tell the user what the state of the processing is. This allows users to aware of what the system is processing.

5. Error Handling

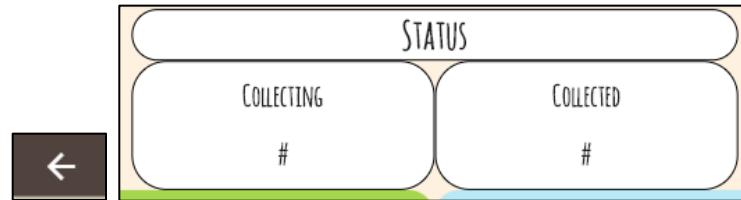
- To prevent error is to control what users can input into the system. For every single error, the user should receive an error page/message indicating the error. Ideally, each error describes the next course of action to resolve this error. Usually, the system is designed such that this occurrence is minimized.

6. Permit Reversal of Action



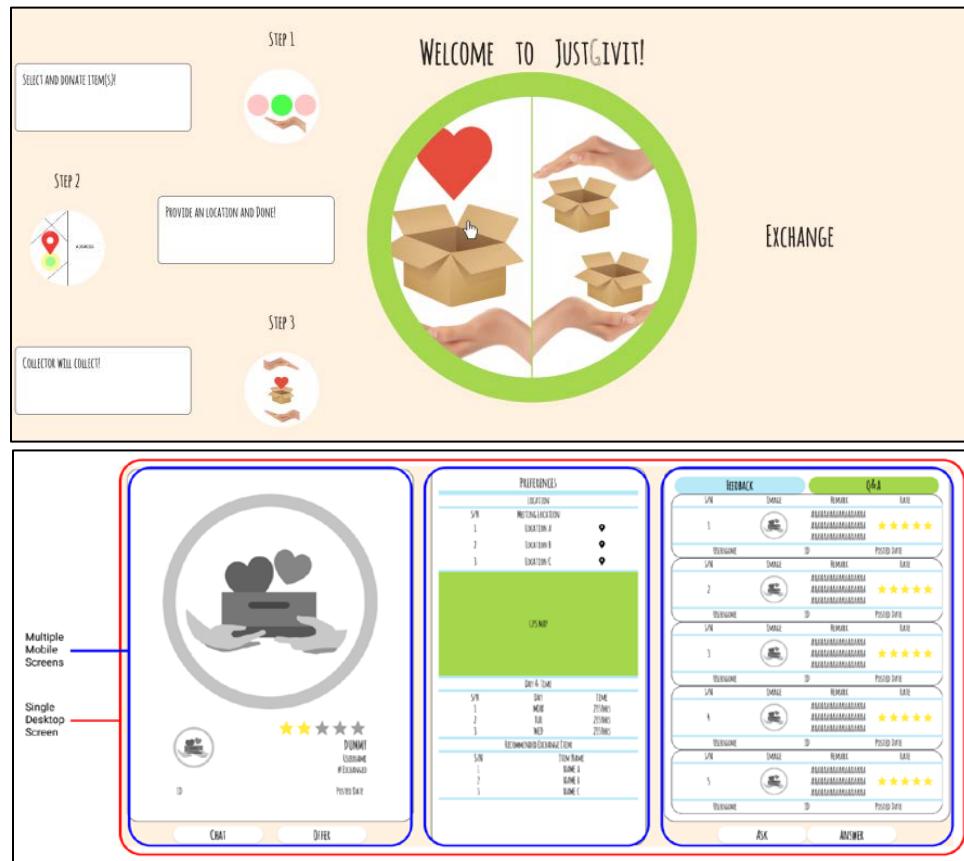
- Some important data is not reversal once submitted. Thus, introduce a message box to double confirm the users.

7. Support internal locus of control



- Users can navigate around the page as well as going back to previous pages with the 'back' button on the top left corner of the screen. Apart from that, there is status to keep track of each process of the exchange or donate items.

8. Reduce short-term memory load



- The rule here is by maintaining a symmetrical view, 3 to 5 element per cluster to ensure information are kept minimal to the users and text usually supported with images or act as icons to further reduce one's memory load. This also heavily depends on which platform is used as the size of the screen limit what we can see. Typically, mobile platform

Tools Used



Fig 29, Logo of Figma

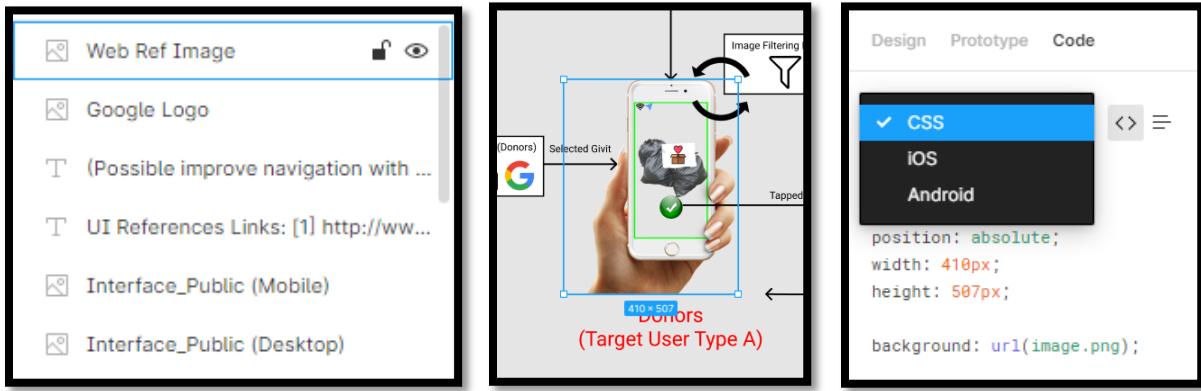


Fig 30, (left) layer features, (middle) the dimension of the selected content in pixel units, and (right) the styling codes that support various languages (CSS, IOS, Android).

In this project, Figma is selected for the digital design and prototyping tool [44]. It is a UI and UX design application that can be used to design such as LO-FI or HI-FI. Apart from that, this tool provides some styling code for Web/Mobile design such as CSS, it also provides the content dimensions as shown in Fig 30 and it has layer features that allow easy re-design or modification. Thus, it allows an easier implementation process.

Initial Concept

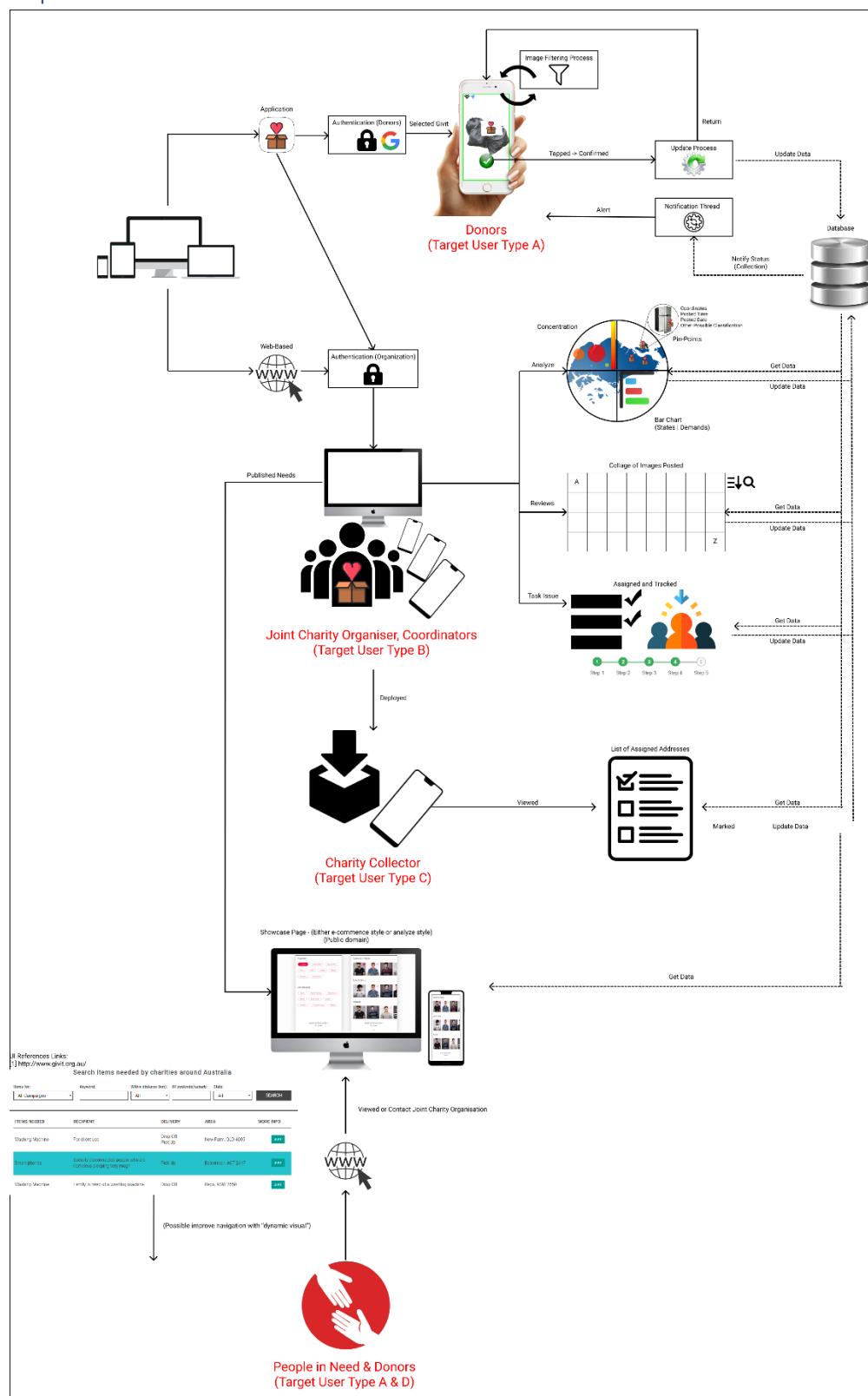


Fig 31, the Initial Concept design of the project

LO-FI

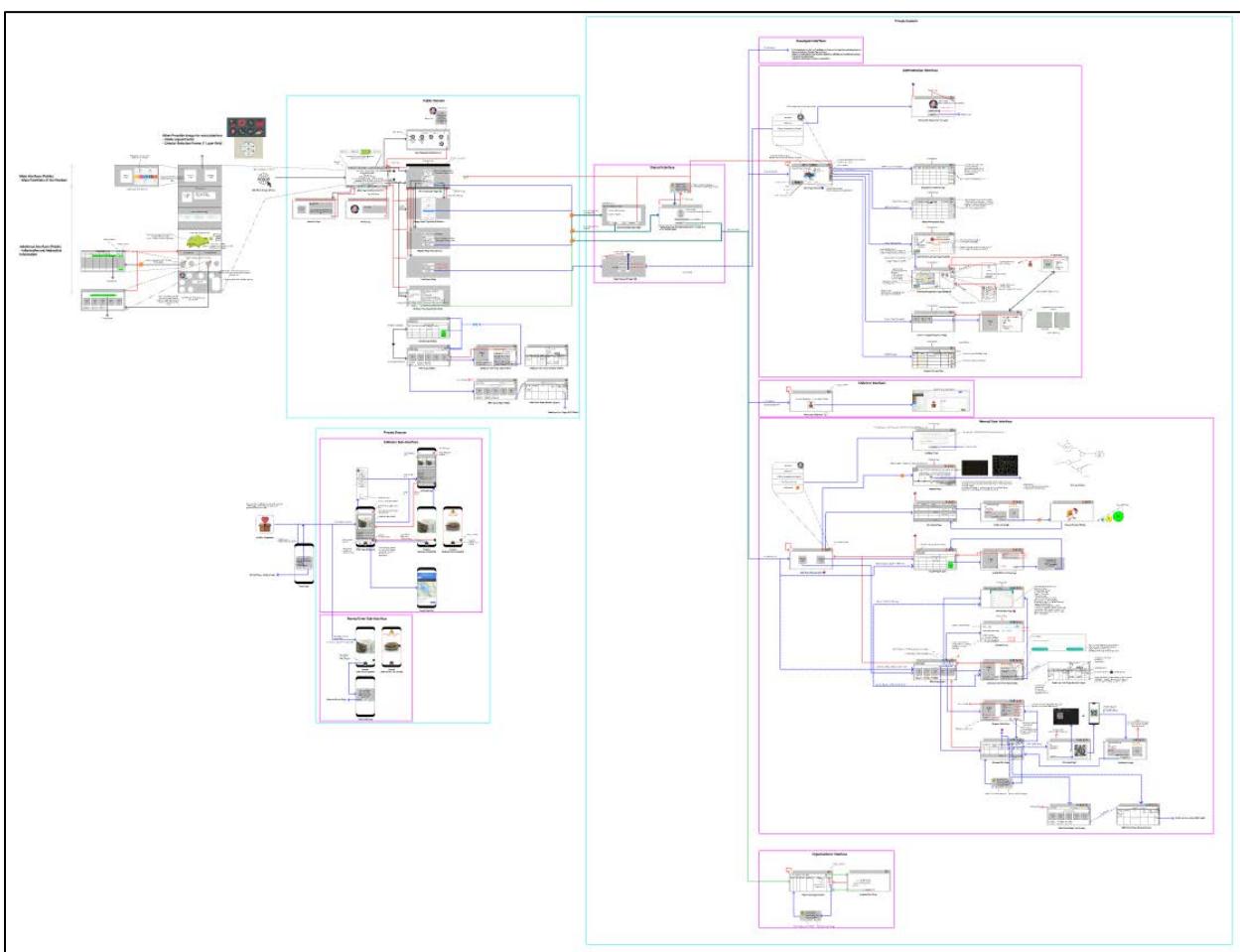


Fig 32, the LOFI design of the project. {14}

Hi-Fi
Android Platform

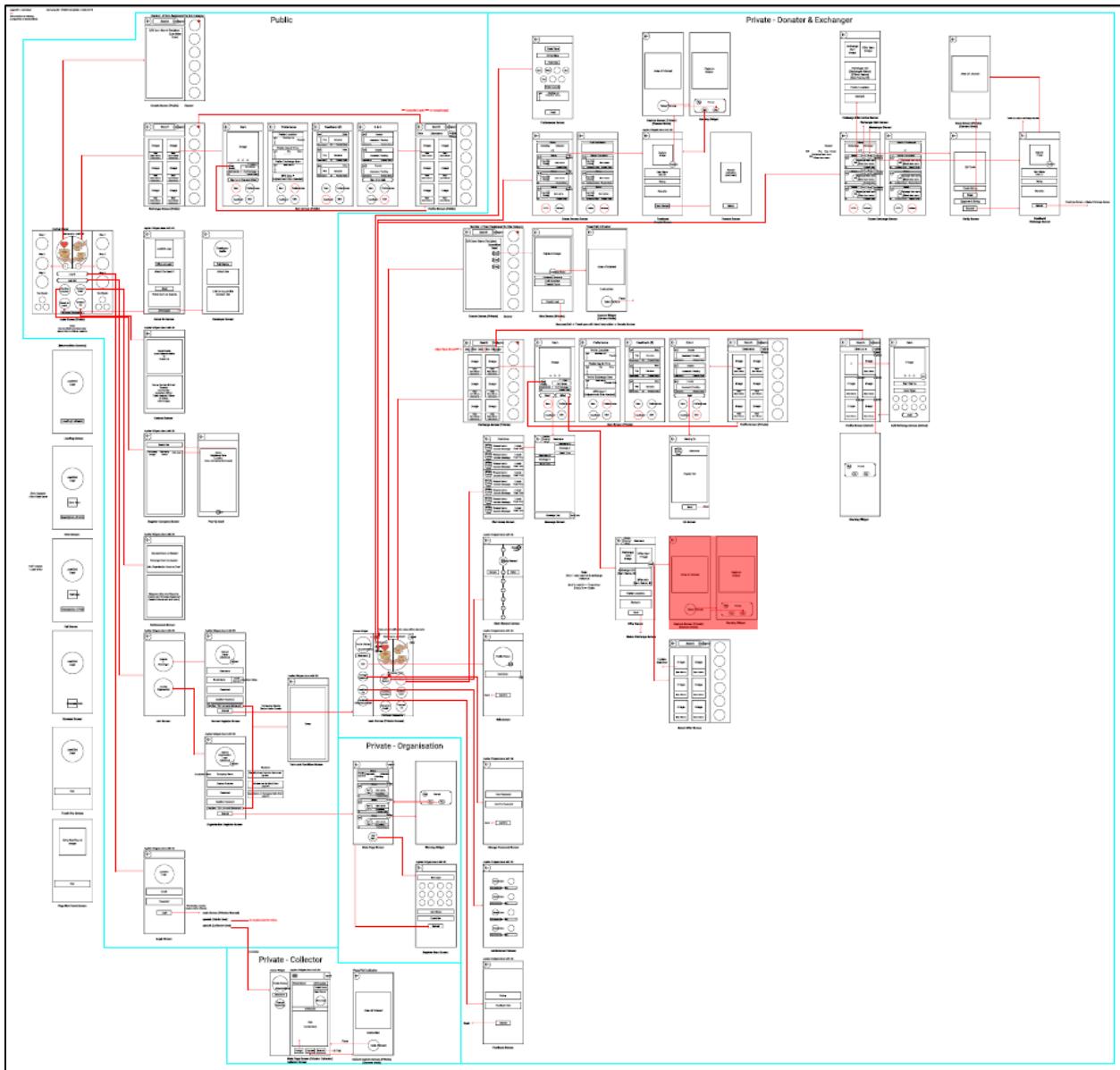


Fig 33, the HIFI design of the project in Android Platform {15}

Web Platform

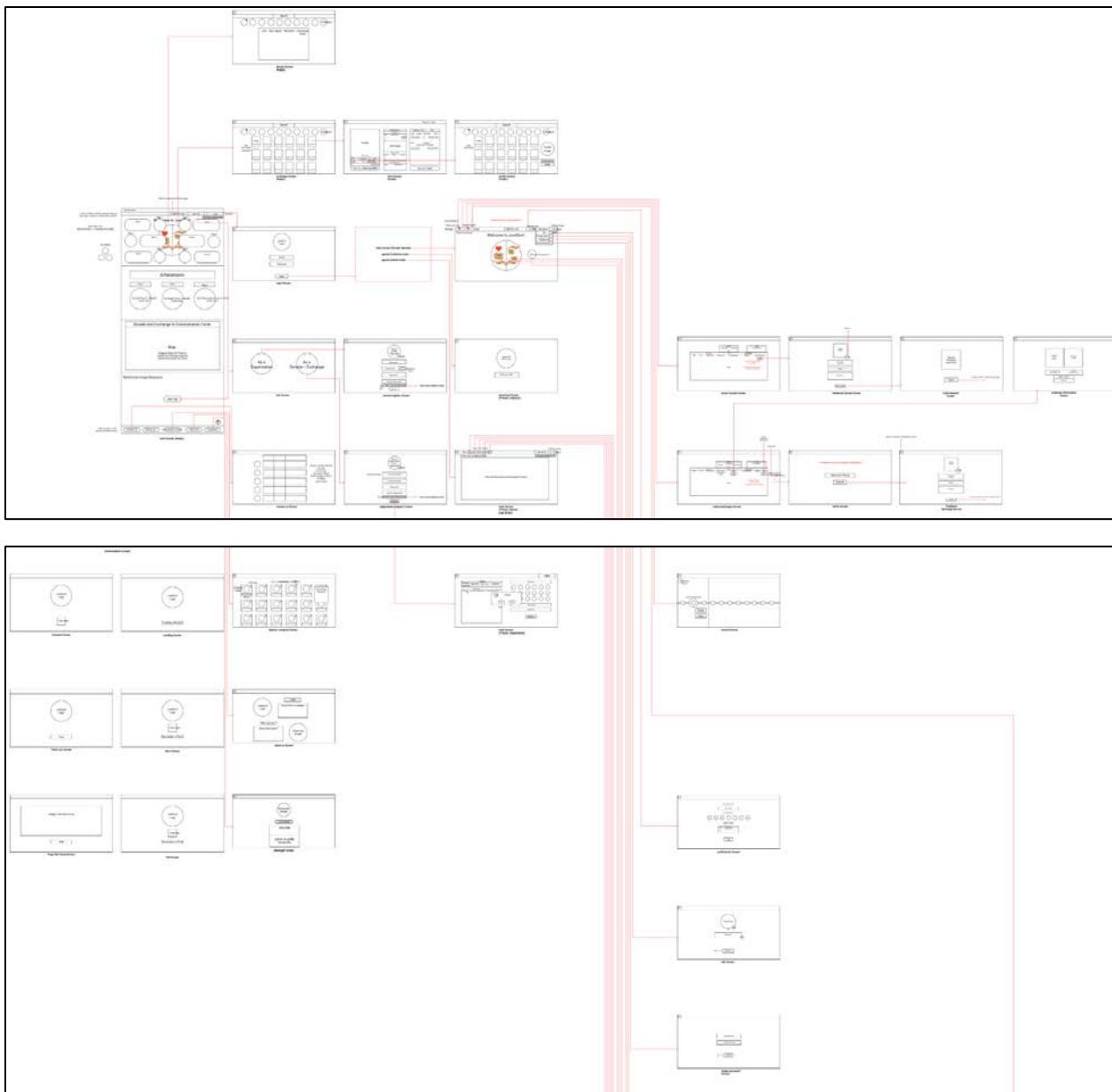


Fig 34.1, the HIFI design of the project in Web Platform (First Half) {15}

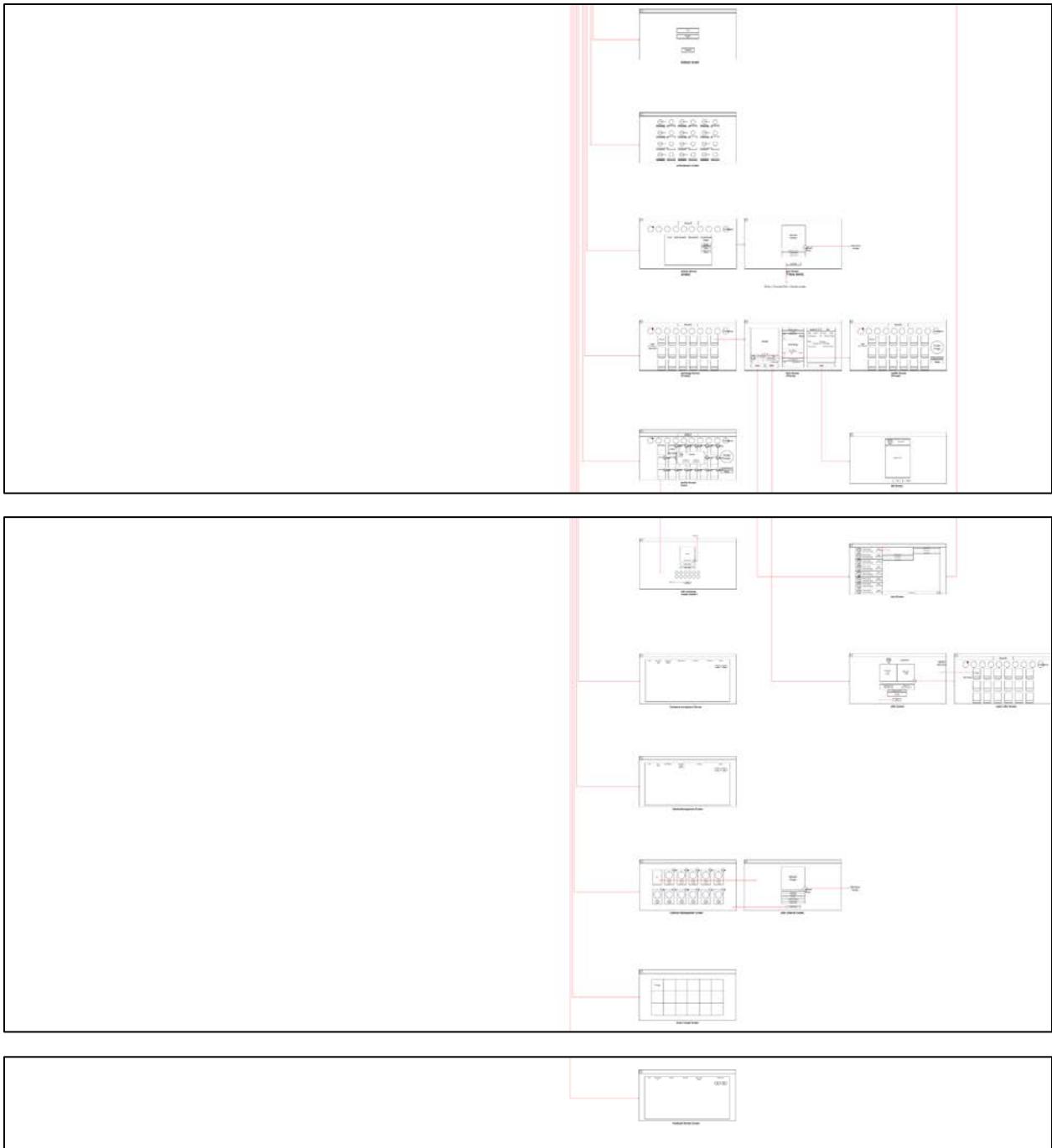


Fig 34.2, the HiFi design of the project in Web Platform (Second Half) {15}

Implementation Specification

Foundation Phase Implementation

The Development Screen shows a list of items under 'Session Mechanism with BLOC'. The items include: Session Key "Token" is : Default, Get, Auth Status: Not Signed In, APK, Error Page, Loading Page, Success Page, Fail Page, Export, Login (Non-Gmail), Register (Non-Gmail), Logout (Non-Gmail), Stream Mechanism with BLOC, QR Code, QR Scan, and QR Scan with BLOC. Below this is a section for 'CRUD Fire Store + (Real-Time Update)' and 'CRUD Fire Storage + (Real-Time Update)'. At the bottom is a 'Session Mechanism' section.

The QR Code Screen displays a QR code with a small logo in the center. Below the QR code is the text 'Random Generate QR Code' and the SHA256 hash of the generated code: c6d4403ef5fb6aa0782a57ff1f128a9fb027ed91b51fb8fc8421784e85604bc.

The Fire Store Screen shows a 'Create / Delete / Update / Read -> Stream -> Message System / Search System' interface. It includes buttons for Manual, Auto, Stop, and Reset DB. A 'Firestore Database' section shows an 'Initial' state. Below it is a table for Actions and Inputs, with Create, Delete, and Update sections. A 'Message System' section contains a JSON message log. At the bottom are 'Key Message' and 'Search DB System' fields.

The Fire Storage Screen shows a 'Note: Real-Time Not Supported. Update equivalent to Upload (URL is unique overwrite not allowed)' message. It features 'Upload Image' and 'Manual Update DB' buttons. Below is a grid of three images of a person in an orange vest. A 'Select a image above first' message is displayed over the images. At the bottom are 'Delete Image', 'Update Image', 'List File', and 'List URL' buttons.

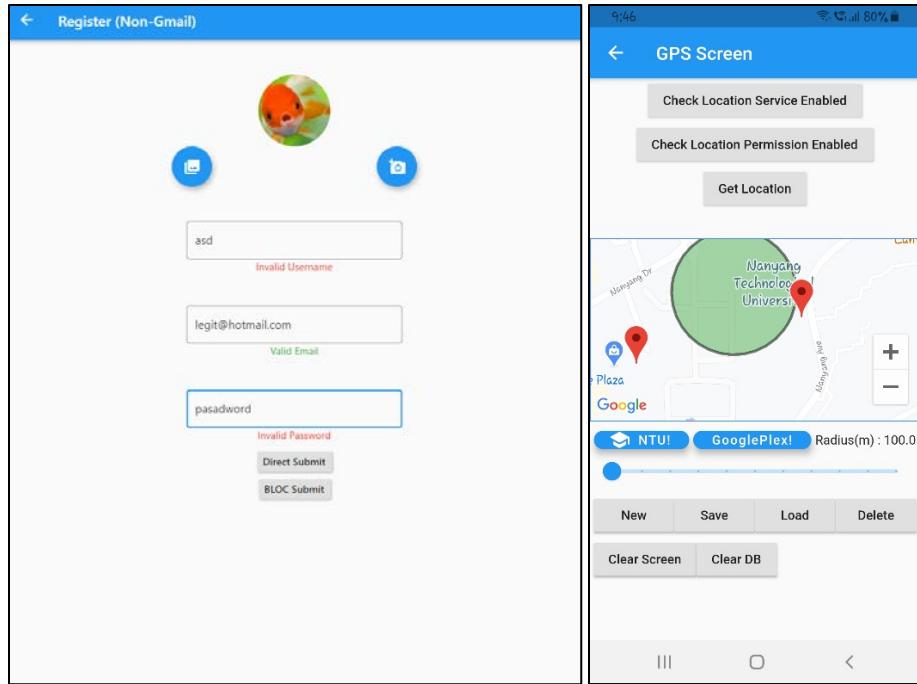


Fig 35, the various screen of foundation development screen {16,17,18}

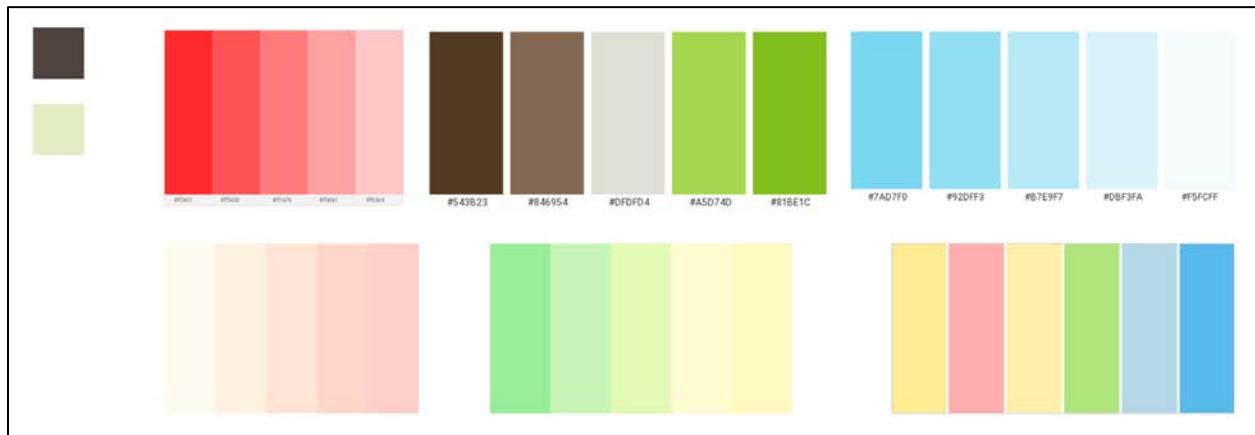


Fig 36, Color theme selection work with Figma

Before beginning implementation, critical functions are to identify and tested in modular form to proof of functionality, once complete most of this functional code can be reuse, integrate, and restructure to fit into the context of this project. These are also tested on both Android and Web platforms to ensure both perform as intended.

System Design Implementation

System-Level

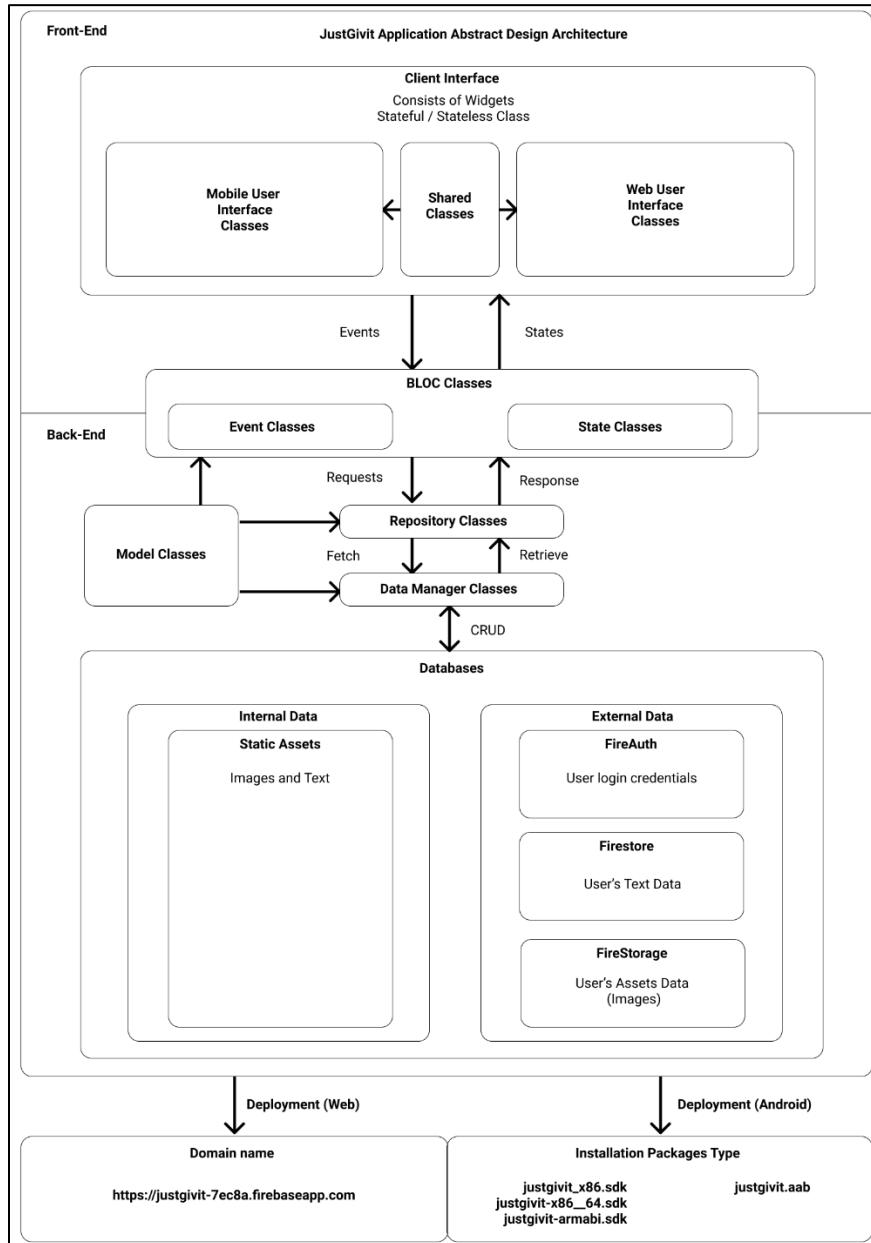


Fig 37, JustGivit system design

For the front-end of the project, the classes would be separated base on the domain platform they are accessing, this is due to differences in the user behavior and screen orientation and size consideration which result in totally different layouts structure while maintaining similar content. The shared classes consist of the same content on both platforms.

Next, the deployment for the web platform would be using Firebase Hosting which allows access from worldwide with internet access, the deployment for the android platform would be an android install application file in APK format. The rest of the content has been mentioned earlier in the report.

Software-Level

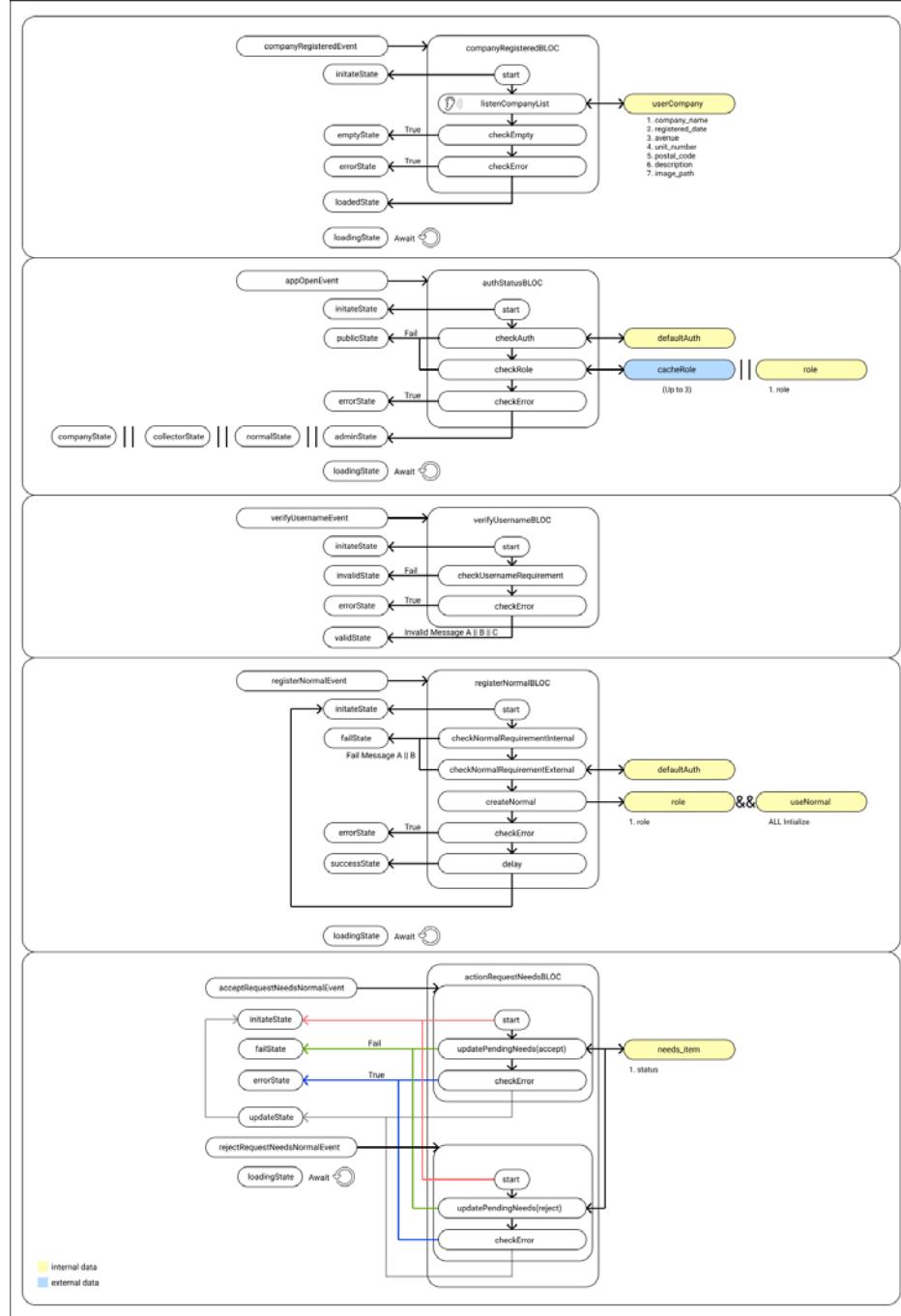


Fig 38, showcase a key sample of JustGivit's BLOC and its interaction between front-end and back-end

As shown in Fig 38, we can see how BLOC was implemented in our project, each BLOC consists of unique sets of logics classes where each of the classes is composed of different classes as well such that separate of concern is achieved. The triggered event(s) communicate with the BLOC itself and return the state which reflects according to the process stage of the logic. The right-hand side of the BLOC depicts the data needed for this BLOC. The data can be retrieved internally or externally depending on the logic.

Database Design Implementation

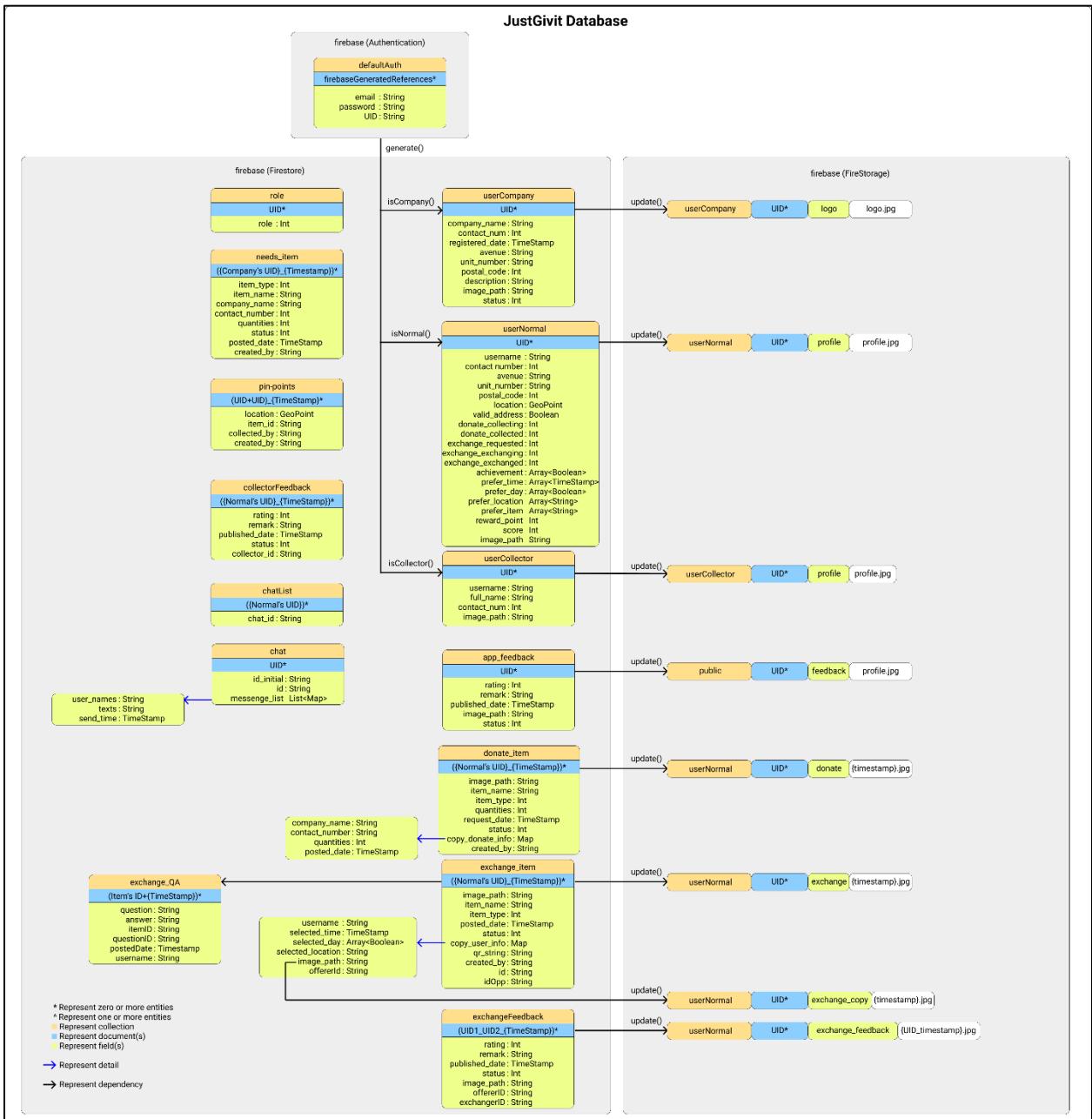


Fig 39, JustGivit database design

The design of this project database ensures it is minimal depth documents. This helps reduce the time taken for query retrieval as the goal here is real-time updates. The document is designed such that they mimic closely the model data allow CRUD operation to be used directly without having to normalized the data.

Directory Organisation

This section will share about the workspace organisation.

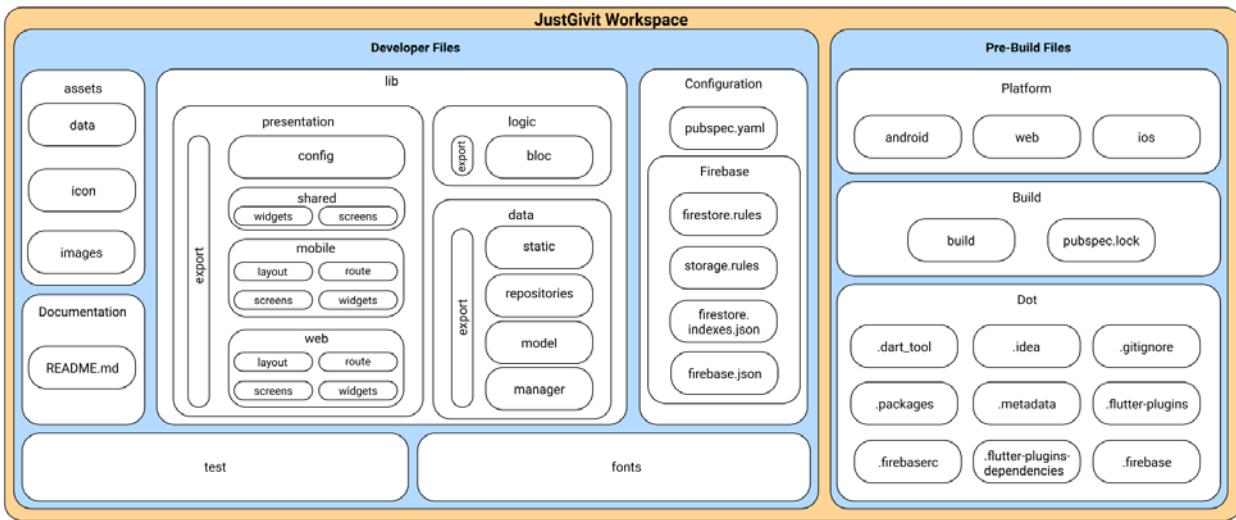


Fig 40, JustGivit workspace organisation design

Pre-Build Files

It refers to the files that are generated by the Flutter SDK command or Firebase

1. .../android/

- It contains a complete android native application project that supports android mobile devices.
- It is used to configure android native settings such as Gradle in both project and app-level to ensure it is compatible with Flutter and android configuration such as permission and dependencies.

2. .../web/

- It contains a web framework that will be compile during run-time
- It also uses add-in firebase configurations as well as tab “JustGivit” icons in the web browser

3. .../ios/

- It contains a complete ios native application project
- In our scope, we did not support ios platform that supports apple mobile device

4. .../build/

- It consists of multiple files that represent the compiled code of the flutter application.

5. .../pubspec.lock

- A configuration file such that it is locked when ‘pubspec’ file is used by Flutter SDK.

6. .../.dart_tool

- It is used to support the ‘pub’ libraries and other necessary tools.

7. .../.idea

- It holds the configuration for the android studio development environment. In our context, we are using VSCode. Thus it is not used.

- 8. .../.gitignore**
 - It contains a list of file extensions and folders that should be ignored when working with git such that changes to the work do not commit to the git repository. It is usually a localized type of file where it is unique on each computer.
- 9. .../.packages**
 - It is automatically generated content for Flutter SDK
 - It consists of a list of dependencies
- 10. .../.metadata**
 - It tracks properties of the Flutter project and manages by Flutter SDK
- 11. .../.flutter-plugins**
 - It consists of the external plugin from the Flutter workshop, it consists of a list of the plugin name, path, and version.
- 12. .../.flutter-plugins-dependences**
 - It consists of the dependency of the various plugins used.
- 13. .../.firebase**
 - It consists of assets and flutter's web services UID and information for firebase web hosting.

Developer Files

It refers to files created by the developer from scratch.

- 1. .../assets/data/**
 - It consists of static data such as the charities company details in .csv format
- 2. .../assets/icon/**
 - It is focused on storing icon-like images which are typically smaller than or equal to 64 by 64 pixels
- 3. .../assets/images/**
 - It is focused on storing big images that are bigger than 64 by 64 pixels.
- 4. .../README.md**
 - It consists of brief information about the project and setup details in documentation form.
- 5. .../test/**
 - It is used to describe and implement automatic test steps or cases.
 - Due to the limited duration of time, it is not used for implementation.
- 6. .../fonts/**
 - It consists of the style of fonts used on the user interface
- 7. .../pubspec.yaml**
 - It is a configuration file for the developer to be used such as adding a plugin from the Flutter workspace.
- 8. .../firebase.json**
 - It is used to deploy resources from a project directory that contains firebase-related operations.
- 9. .../firestore.indexes.json**
 - It is the index configuration file for firestore.

10. .../firestore.rules

- It consists of firestore rules which can also be set up on the firebase console area.

11. .../storage.rules

- It consists of storage rules which can also be set up on the firebase console area.

12. .../lib/presentation/config/

- It represents the UI configuration for both web and mobile platforms such as color theme, image strings, path strings, and scaling constants.

13. .../lib/presentation/shared/widgets/

- It consists of common widgets used by both mobile and web platforms such as buttons, textbox, checkbox, etc.

14. .../lib/presentation/shared/screens/

- It consists of screens with the same structure and behavior.

15. .../lib/presentation/{mobile|web}/layout/

- It consists of a common layout that persists mostly throughout the screen such as the header and footer of the screen.

16. .../lib/presentation/{mobile|web}/route/

- The navigation generated (lookup table) route live here with the pointer to respective screens.

17. .../lib/presentation/{mobile|web}/screens/

- It consists of the entire screens , each screen can be split into subscreen or separated by their roles and system type.
(admin,auth,company,public,user,user_donor,user_exchange)

18. .../lib/presentation/{mobile|web}/widgets/

- It consists of platform-based widgets.

19. .../lib/logic/bloc/

- The BLOC mechanism live here.

20. .../lib/data/static/

- It stores static data that do not change over time during the entire application life cycle such as fixed text.

21. .../lib/data/repositories/

- It stores all cache instances and responsible to fetch and retrieve respective data.

22. .../lib/data/model/

- The data model live here.

23. .../lib/data/manager/

- The data manager design to control firebase's data access lives here.

24. Export .dart files

- These are files that help bundle other .dart files such that a single import declaration statement is used without import them individually which helps better readability.

UI Design Implementation

This section will share the web UI how each key element interact, reasoning or purpose in the UI (if any). **Mobile UI** corresponding to the web UI as shown below will be placed in the appendix.

Public UI

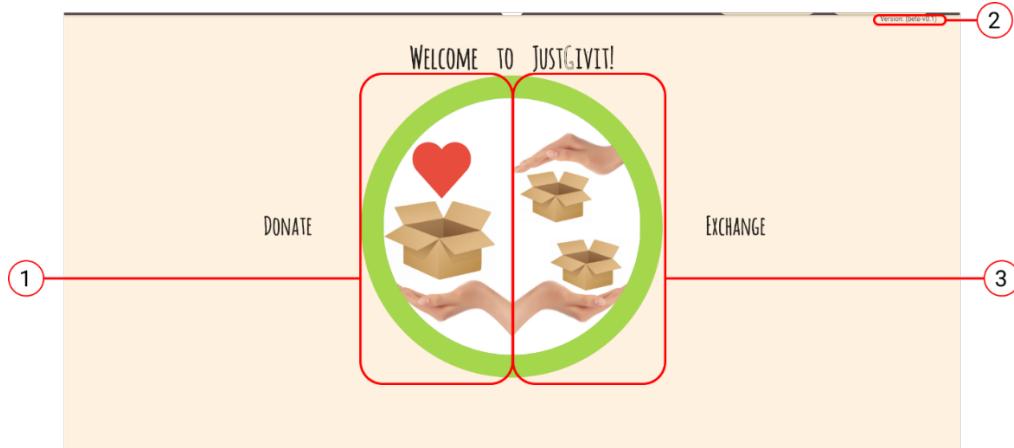


Fig 41, Public and Normal User Main Screen

1. This access to donate market screen for the user to donate their unwanted and reusable item(s).
2. This is to show the version of this application development, this helps the developer to troubleshoot, should the user encounter an error at this particular version. Ideally, when the user encounters an error, the version will be tagged along in the error report for the developer to review.
3. This access to exchange market screen for the user to exchange their unwanted and reusable item(s).

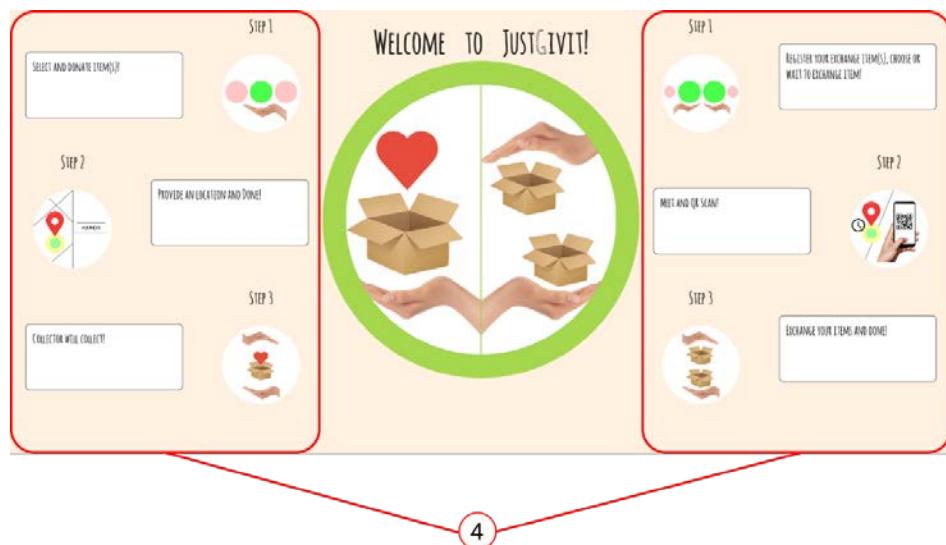


Fig 42, Public main screen with details

4. This information is to show the public user more information on how to use this application in summary.

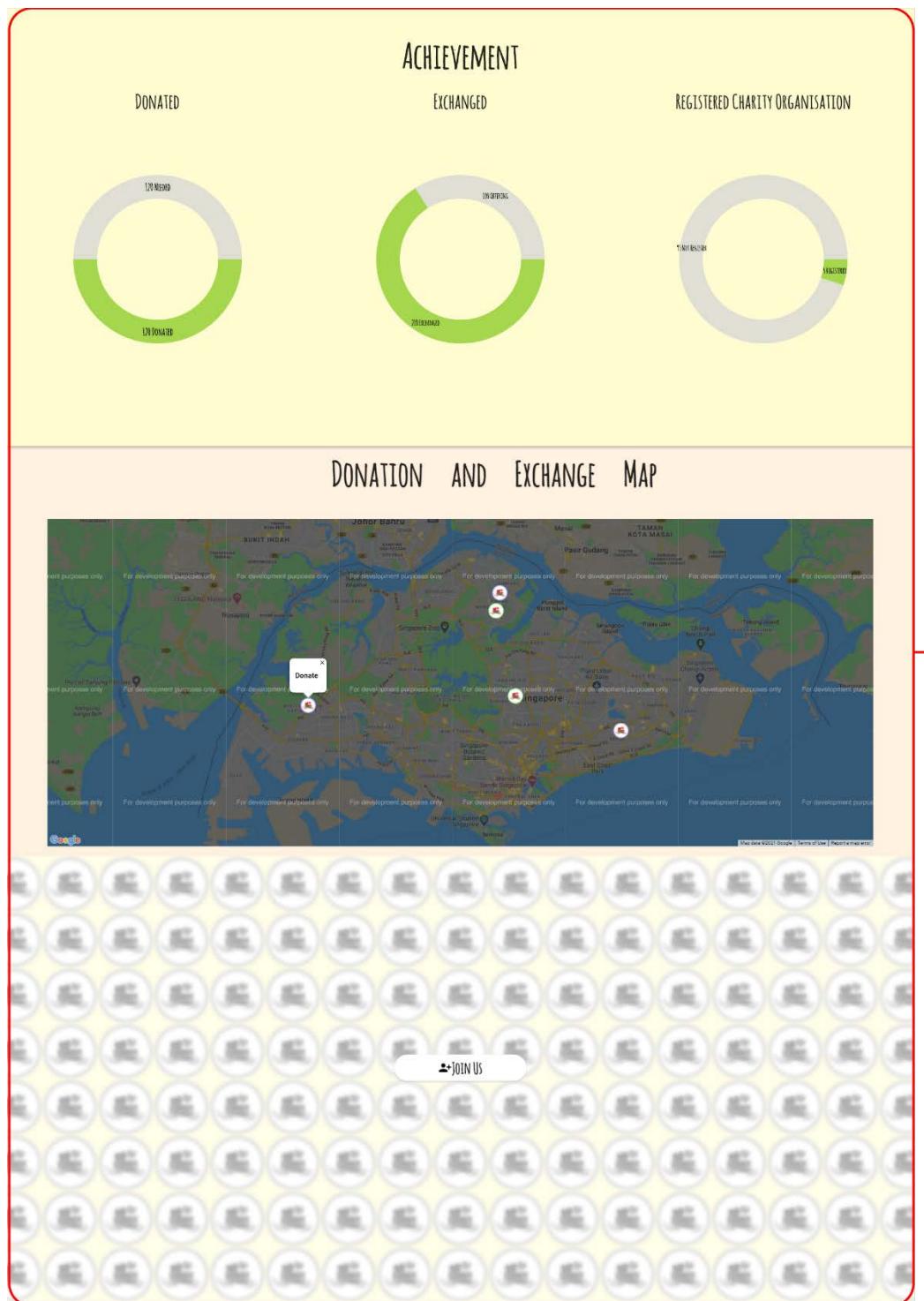


Fig 43, Public main Sub-screens with details

1. This is part of the main screen that encourages the public user to join the application through real facts. At the bottom part of the screen will show the normal users approved profile image use in the background and ideally with their reviews about this application's positive feedback. Besides that, the 'Join Us' button will lead to the Join screen.

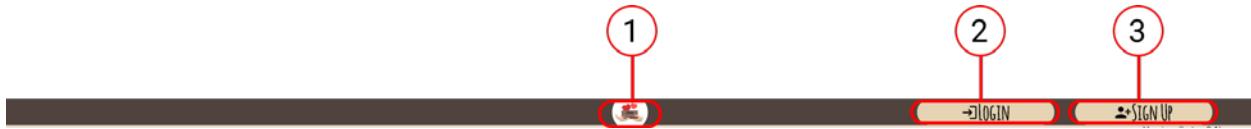


Fig 44, Public header of the screen

1. This represents the home button that will 'pop all' past screens and return to the main screen for the user's convenience.
2. This will access the login screen.
3. This will access the join screen.

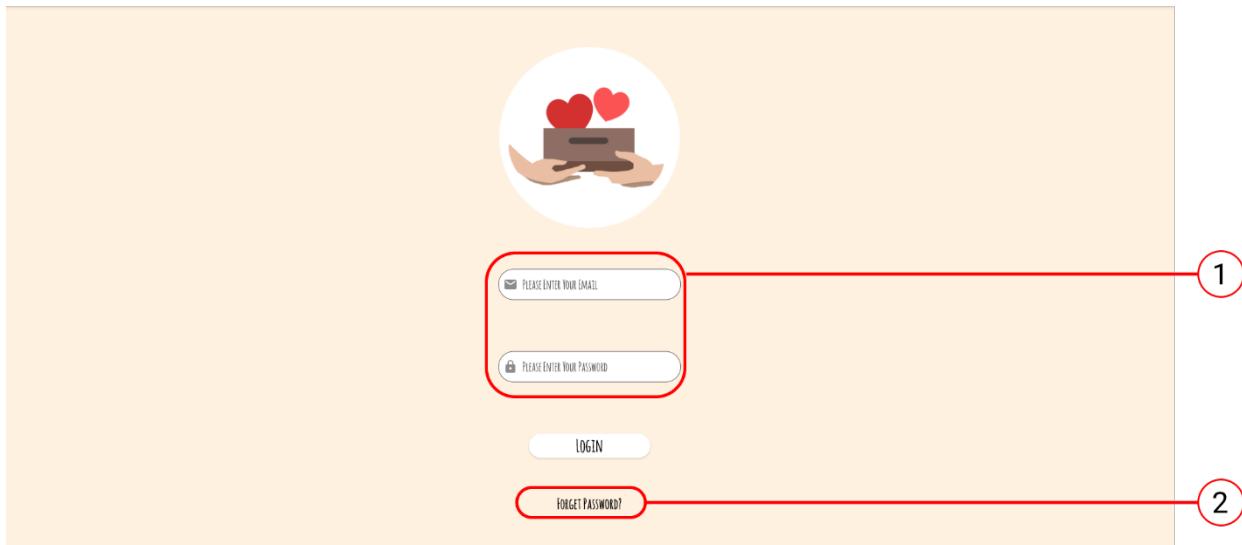


Fig 45, the login screen

1. This allows users to enter their account credentials to log in.
2. Should users forgot about their password, they can change their password.



Fig 46, Join screen

1. There are 2 options to join dependent on the role, as a company representative or normal representative.



NAME OF ORGANISATION

PLEASE ENTER CONTACT NUMBER

PLEASE ENTER YOUR PASSWORD

PLEASE RE-ENTER YOUR PASSWORD

I AGREE TO THE TERMS & CONDITIONS

REGISTER

Fig 47, Register company screen (top) and TNC screen (bottom)

1. To register needs, company representative users must register an account. They must provide the appropriate credentials with the agreement of the terms and conditions.

A registration form with a red box highlighting the logo and fields. The logo at the top features a heart-shaped cake with two hearts on it, enclosed in a circle. Below the logo are four input fields: 'PLEASE ENTER YOUR USERNAME' with a 'SIGN UP' button, 'EMAIL NAME' and 'GMAIL.COM' dropdown menus, 'PLEASE ENTER YOUR PASSWORD', and 'PLEASE RE-ENTER YOUR PASSWORD'. At the bottom are a checkbox for terms and conditions and a 'REGISTER' button.

Fig 48, Register normal screen

1. To donate or exchange an item, normal users must register an account. They must provide the appropriate credentials with the agreement of the terms and conditions.

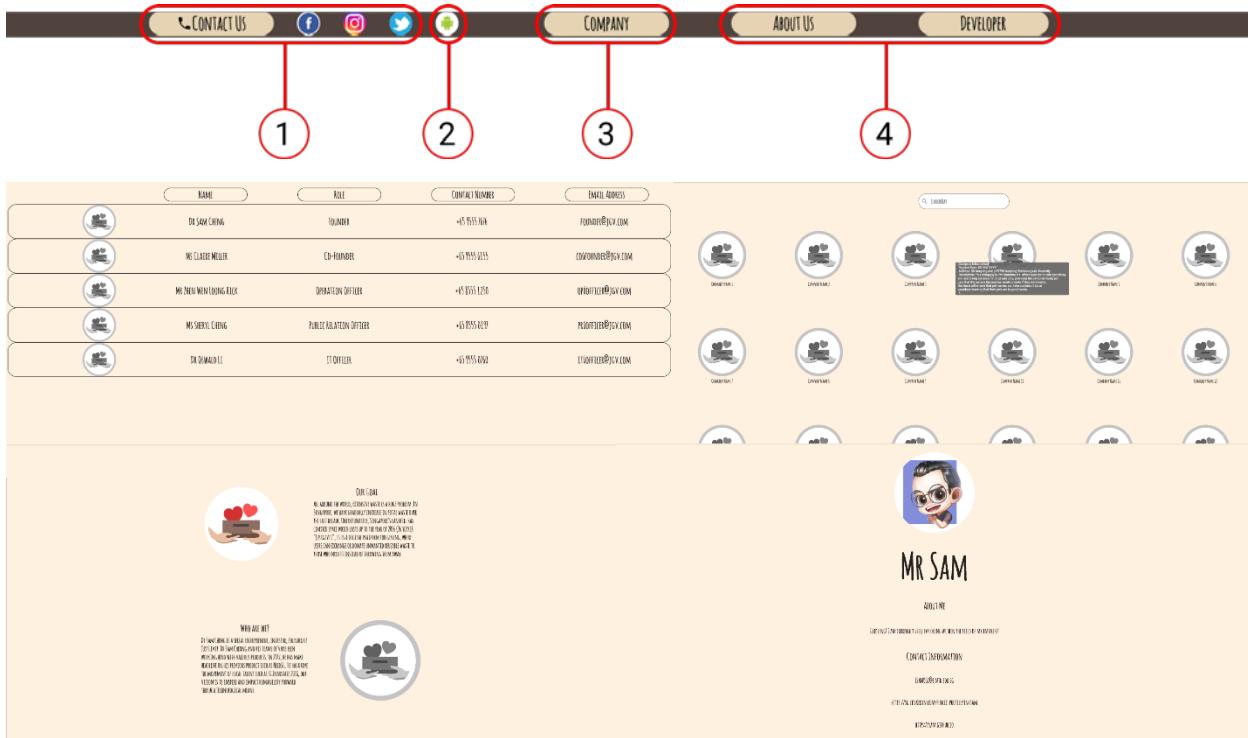


Fig 49, Footer of the screen (top) and 4 informative-related screens (bottom)

1. Users can find out more about the applications or people responsible for this application and operations.
2. Users can download the mobile(Android) version to use 'JustGivit' on their mobile phones.
3. Users can find more about which company joined and their information provided by government charity association.
4. Users can find more about people behind the scene who develop this application and the goal behind.

Normal UI

Shared UI

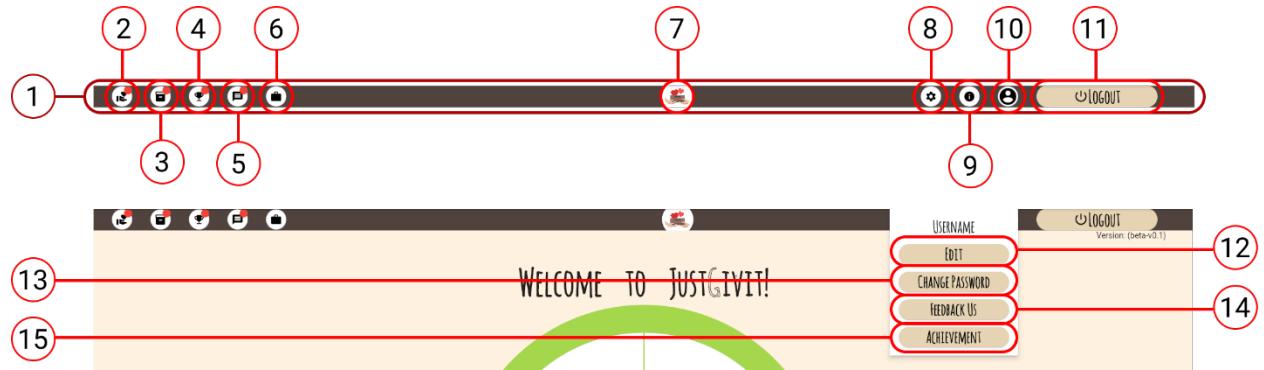


Fig 50, Normal header of the screen

1. The normal header of the screen is visible in most other normal's screens, which allows frequently used screens to be accessible easily.
2. This will allow the user to check the status of the donation.
3. This will allow the user to check the status of the exchange.
4. This will allow the user to claim their reward from the donation.
5. This will access the chat screen.
6. This will allow the user to access their exchange items.
7. This represents the home button that will 'pop all' past screens and return to the main screen for the user's convenience.
8. This will access the user's exchange preferences options.
9. This is to access the guidance page which will assist users with common enquiries about the procedure and operations of this system.
10. This will pop-up a profile menu of the users.
11. This is to logout of the 'JustGivit' system.
12. This will access to edit screen which allows users to customize their profile such as username and profile image.
13. This will user access to the change password screen, the user is required to provide an existing email.
14. This will access to application feedback screen.
15. This will access to achievement screen.

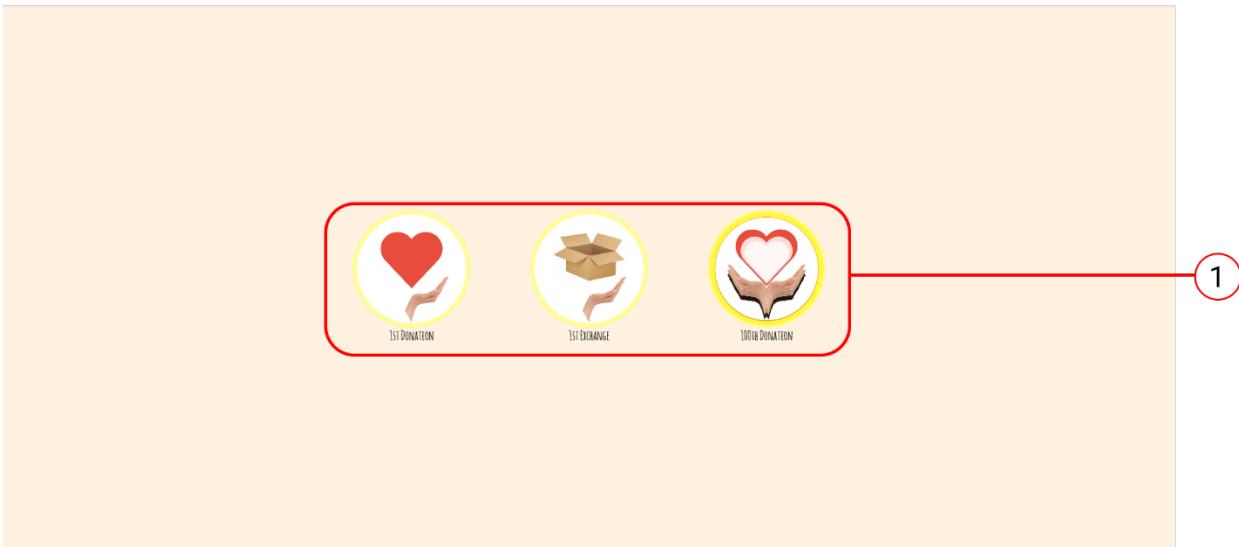


Fig 51, Achievement screen

1. This shows user's achievement has done after donation and exchange using this application. This may encourage and rewards user in return. Ideally, users can share their achievements with others in a way to help promote and spread awareness of this 'JustGivit' application.

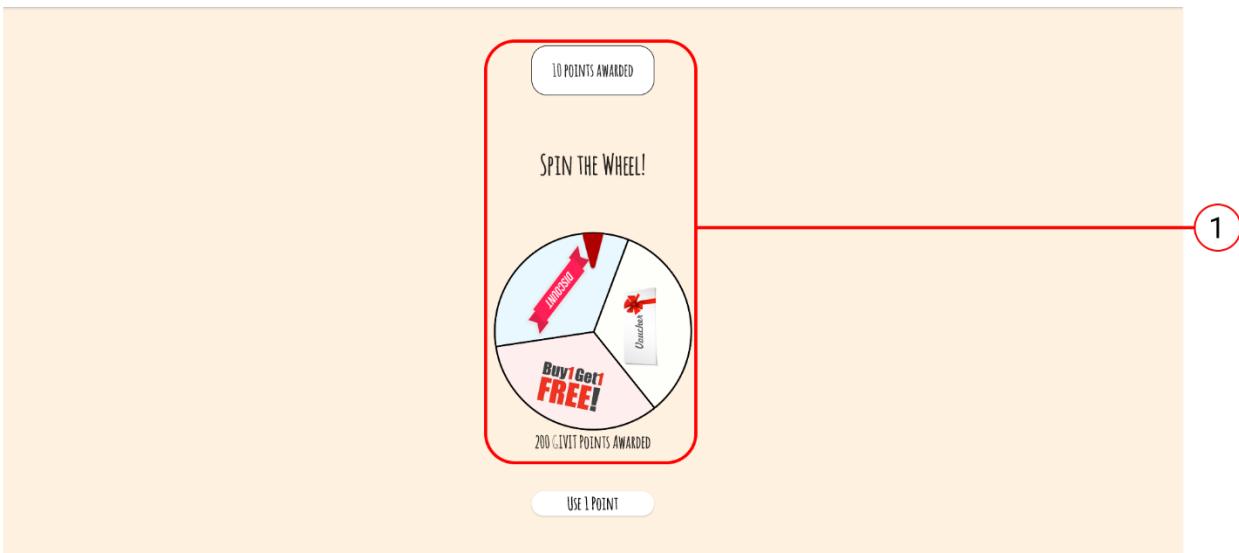


Fig 52, Reward screen

1. This is a simple incentive to encourage the user to donate more such that they can have benefits on their daily expenses in return. However, such incentives require sponsorship from businesses and companies that users spend on. The GIVIT point is an accumulative point that is used on long-term seasonal incentives that allow users re-visit this application from time to time. (Note: Long-Term incentives do not plan for implementation)

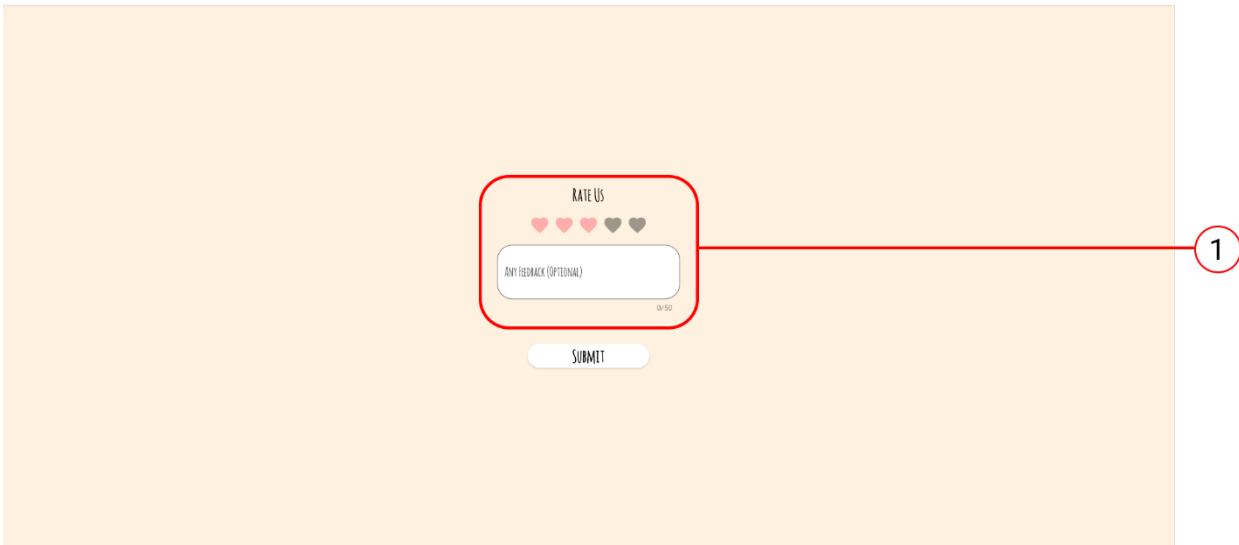


Fig 53, Application feedback screen

1. The user can provide multiple feedbacks to the application system which will be review by the administrator.

Donor UI

The image shows a 'Donate market' screen with a light beige background. At the top is a horizontal navigation bar with six circular icons, each containing a small illustration of a person carrying a load. Below this is a table with columns: S/N, ITEM TYPE, ITEM NAME, QUANTITIES, and ACTION. The table has 7 rows, all showing 'PAPER' as the item type, 'NEWSPAPER' as the name, and '99' as the quantity. Each row has a green 'GIVE' button in the ACTION column. A large red box highlights the first row, and a red bracket at the bottom spans the entire width of the table, connecting the first row's red box to the last row's green 'GIVE' button. Three numbered circles (1, 2, 3) are placed around the highlighted area: circle 1 is near the top row of icons, circle 2 is below the first row of the table, and circle 3 is below the last row of the table.

| S/N | ITEM TYPE | ITEM NAME | QUANTITIES | ACTION |
|-----|-----------|-----------|------------|-------------------|
| 1 | PAPER | NEWSPAPER | 99 | GIVE |
| 2 | PAPER | NEWSPAPER | 99 | GIVE |
| 3 | PAPER | NEWSPAPER | 99 | GIVE |
| 4 | PAPER | NEWSPAPER | 99 | GIVE |
| 5 | PAPER | NEWSPAPER | 99 | GIVE |
| 6 | PAPER | NEWSPAPER | 99 | GIVE |
| 7 | PAPER | NEWSPAPER | 99 | GIVE |

Fig 54, Donate market screen

1. This represents the item types of the application to allow an easier search.
2. This shows the list of needs by various charity companies.
3. This allows the user to access the 'give' screen to donate items. This is only visible to normal users.

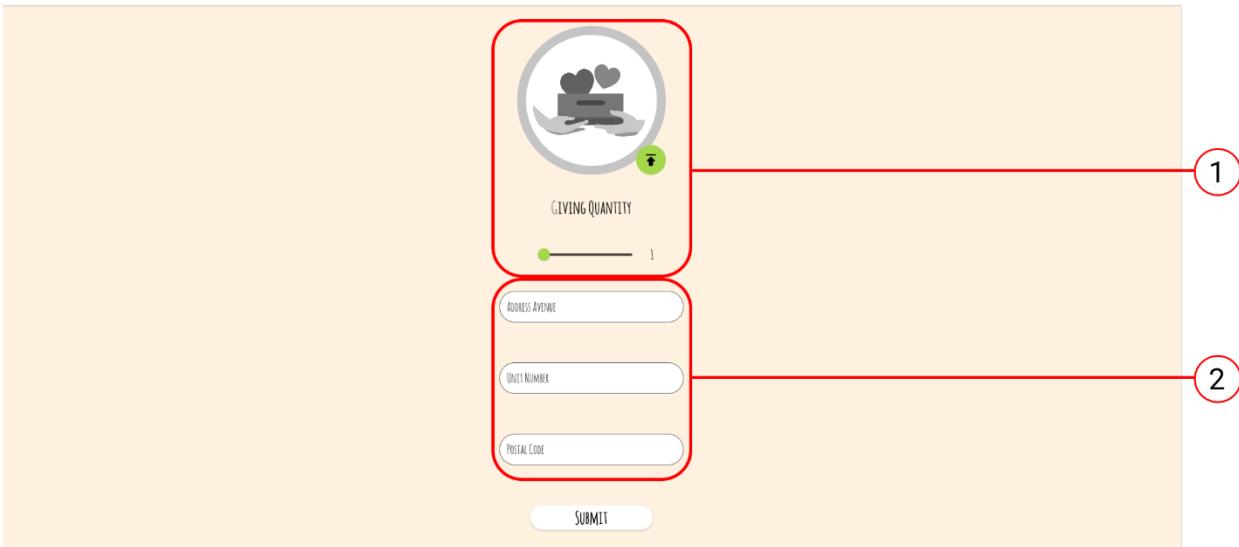


Fig 55, Give screen

1. The user must provide the quantity depending on the quantities needed. The user also needs to upload an image if the user is using a desktop.
For android normal users, the user will just require to capture an image, the image will be then uploaded to the system.
2. The users are required to fill in their current location accordingly, if they are using a phone to access then this part will be auto-filled.

The image shows a table titled 'ACTIVE' and 'ACHIEVE'. The columns are labeled: S/N, ID, ITEM TYPE, ITEM NAME, POSTED DATE, STATUS, and CLAIM REWARD. The 'STATUS' column contains two sub-sections: 'COLLECTING' and 'COLLECTED'. A red line labeled '1' connects the 'COLLECTING' section to the 'COLLECTED' section. A red line labeled '2' connects the 'ACTIVE' header to the 'ACHIEVE' header. The table has 7 rows, each with a 'CLAIM' button in the 'CLAIM REWARD' column. A red line labeled '3' connects the 'ITEM NAME' column to the 'ITEM TYPE' column. A red line labeled '4' connects the 'ITEM TYPE' column to the 'ITEM NAME' column.

| ACTIVE | | ACHIEVE | | | | |
|--------|-----------|-----------|-----------|-------------|------------|--------------------|
| S/N | ID | ITEM TYPE | ITEM NAME | POSTED DATE | STATUS | CLAIM REWARD |
| 1 | 123456789 | PAPER | NEWSPAPER | DD-MM-YYYY | COLLECTING | CLAIM |
| 2 | 123456789 | PAPER | NEWSPAPER | DD-MM-YYYY | COLLECTING | CLAIM |
| 3 | 123456789 | PAPER | NEWSPAPER | DD-MM-YYYY | COLLECTING | CLAIM |
| 4 | 123456789 | PAPER | NEWSPAPER | DD-MM-YYYY | COLLECTING | CLAIM |
| 5 | 123456789 | PAPER | NEWSPAPER | DD-MM-YYYY | COLLECTING | CLAIM |
| 6 | 123456789 | PAPER | NEWSPAPER | DD-MM-YYYY | COLLECTING | CLAIM |
| 7 | 123456789 | PAPER | NEWSPAPER | DD-MM-YYYY | COLLECTING | CLAIM |

Fig 56, Donate status screen

1. User can check their status to keep track of their donated item.
2. The user can review past and current donated items.
3. This represents the past and current donated items details.

- Once the user has donated, and the collector has successfully collected. The user is allowed to claim the reward. This will redirect the user to the donate feedback screen.

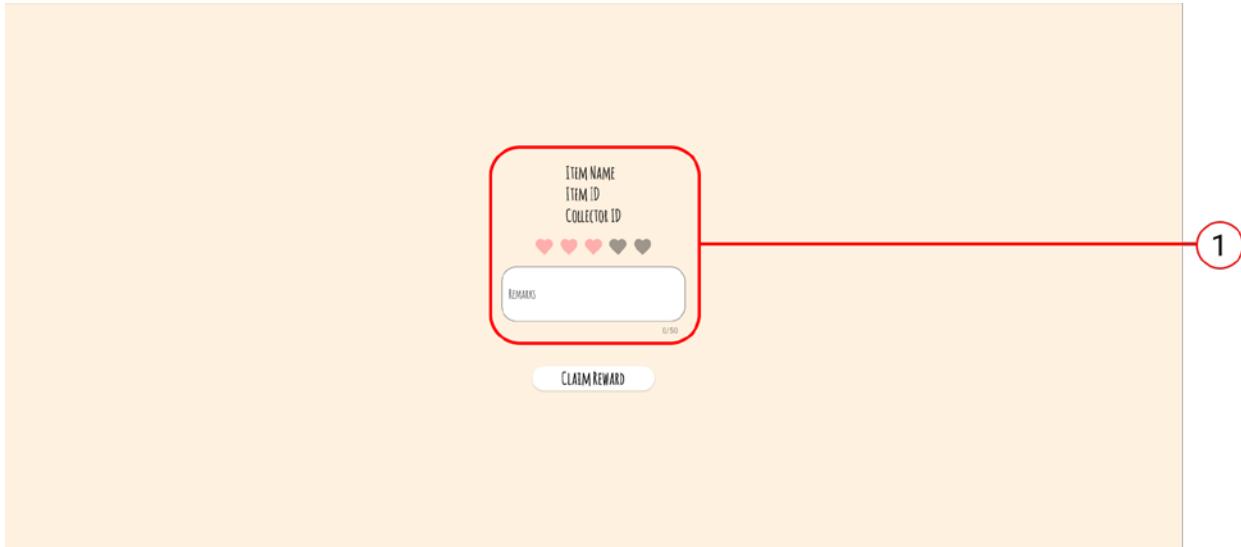


Fig 57, Donate feedback screen

- The user has to provide feedback before claiming rewards.

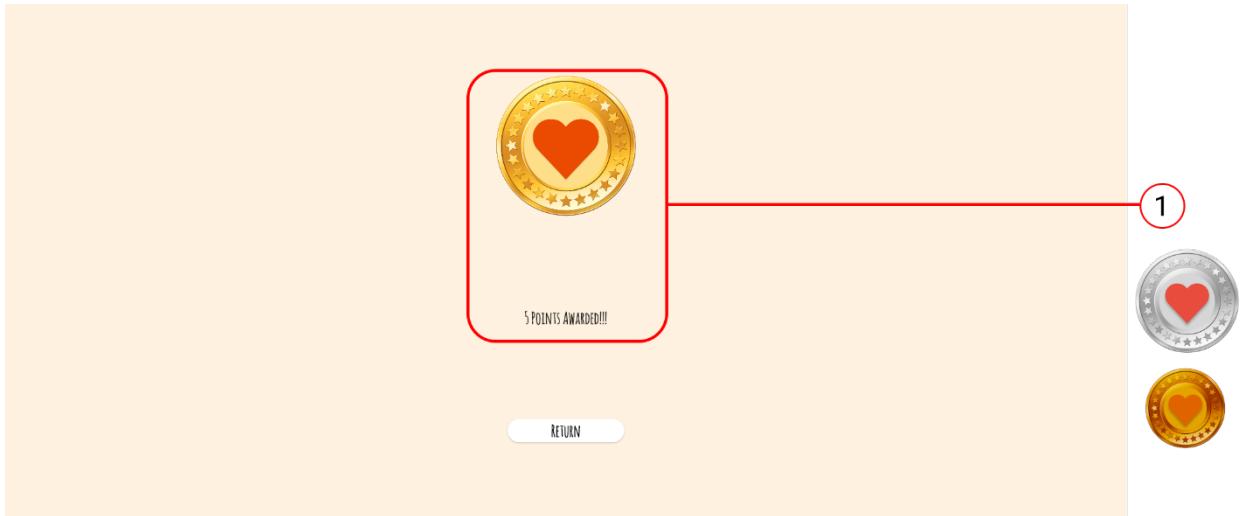


Fig 58, Claimed reward screen

- The user will be given 1,3,5 reward points which are randomly distributed. At the moment, the user will be given a static image corresponding to the point they received. Ideally, there should be a delay where the user can interact or an animation with the reward by ‘unwrapping’ it for the element of surprise like the Gacha system found in typical mobile games.

Exchange UI

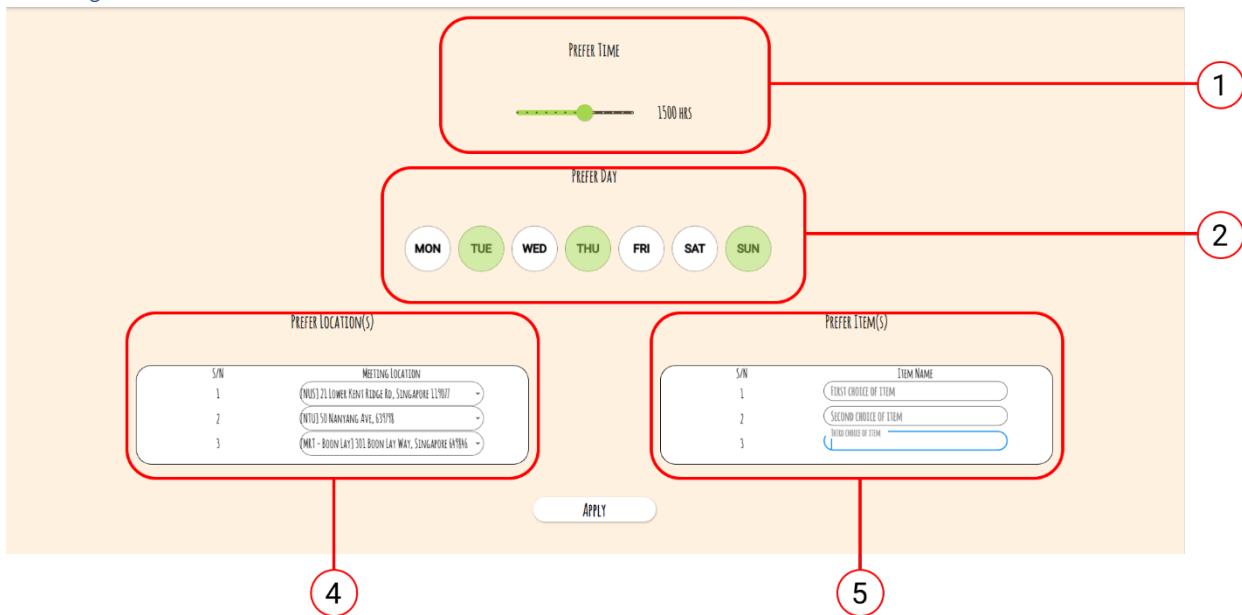


Fig 59, Exchange preferences screen

1. This sets the preferred time for meeting with other exchange users.
2. This sets the preferred day(s) for meeting with other exchange users.
3. This sets up to 3 preferred allocated location(s) for the meeting.
4. This sets up to 3 preferred items that other exchange users may negotiate or consider.



Fig 60, Manage exchange item(s) screen

1. This shows the exchange user's profile and its rating so far to estimate their exchange performance.
2. This shows their registered exchange items.

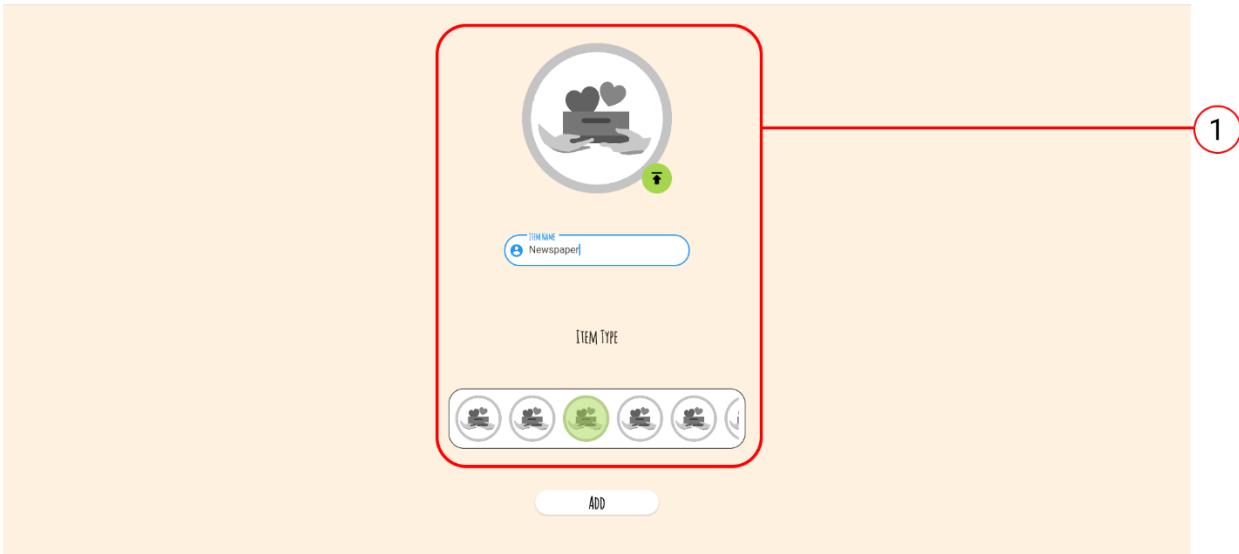


Fig 61, Register an exchange item screen

1. The required information to register an exchange item successfully.

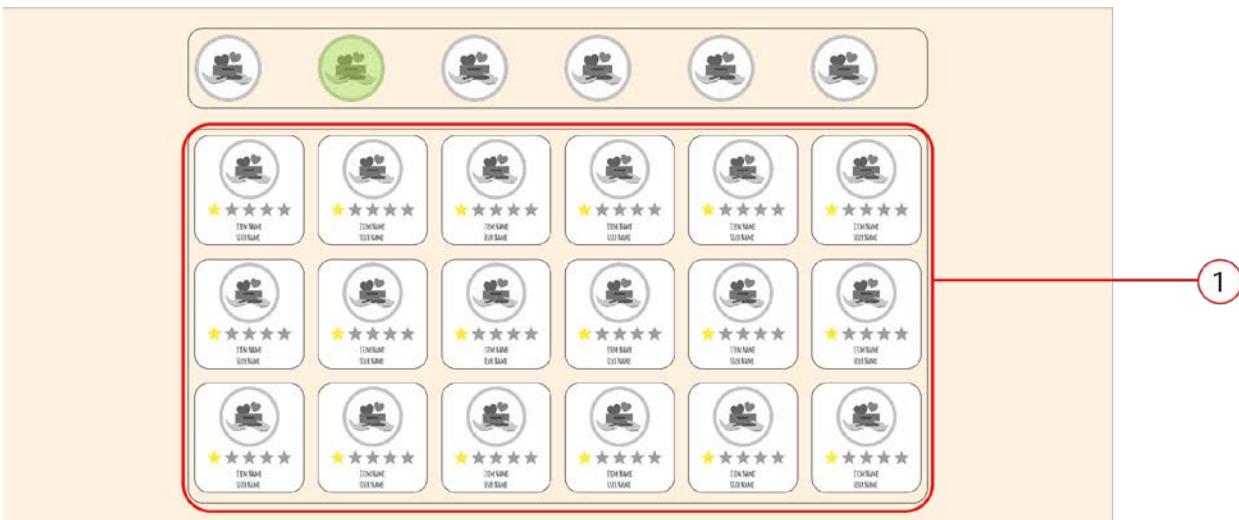


Fig 62, Exchange market screen

1. This shows other exchange users' items such that exchange user can offer their exchange item.

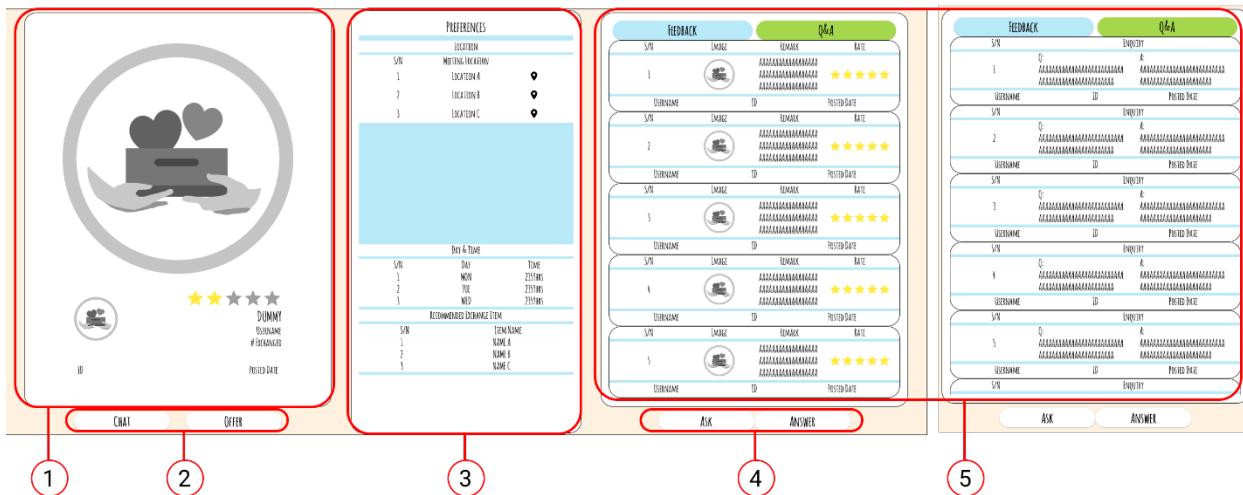


Fig 63, Exchange item detail screen

1. It consists of the exchange item details and the exchange user who registered this item and its facts.
2. The exchange user can chat or offer (this is only visible to normal users)
3. This shows the preferences details where the exchange user has set.
4. The exchange user can enquire or answer the enquiry (this is only visible to normal users)
5. This is the review of how the exchange user's reliability, experiences, and feedback with exchange items.

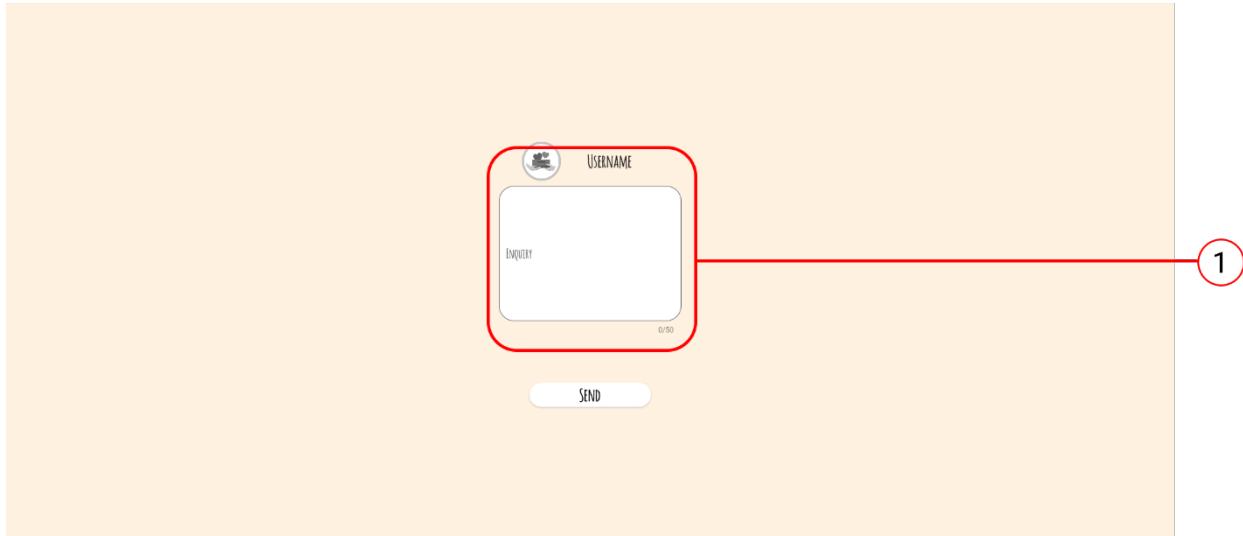


Fig 64, Enquiry screen

1. If the user has doubts or questions, they may choose to enquire here.

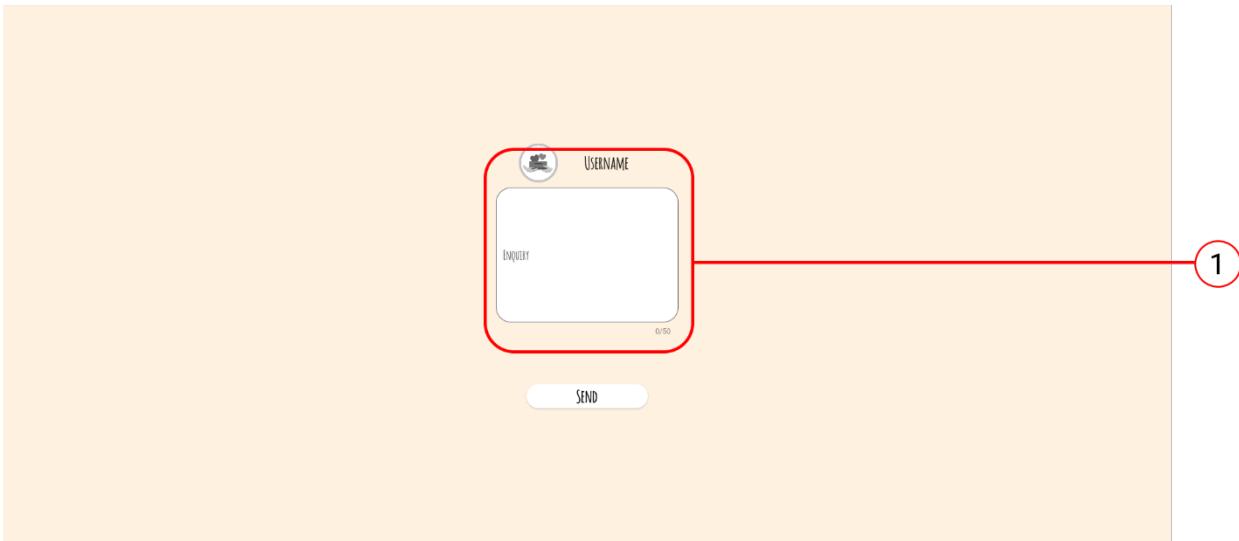


Fig 65, Answer screen

1. The exchange user may answer other exchange user enquiries here.



Fig 66, Chat screen

1. This is the chat that exists with other exchange users.
2. This will show the conversation details with the other exchange users.
3. This will send a message to other exchange users, the other exchange user should receive the updated conversation list. The users can further discuss, negotiate about the exchange items.

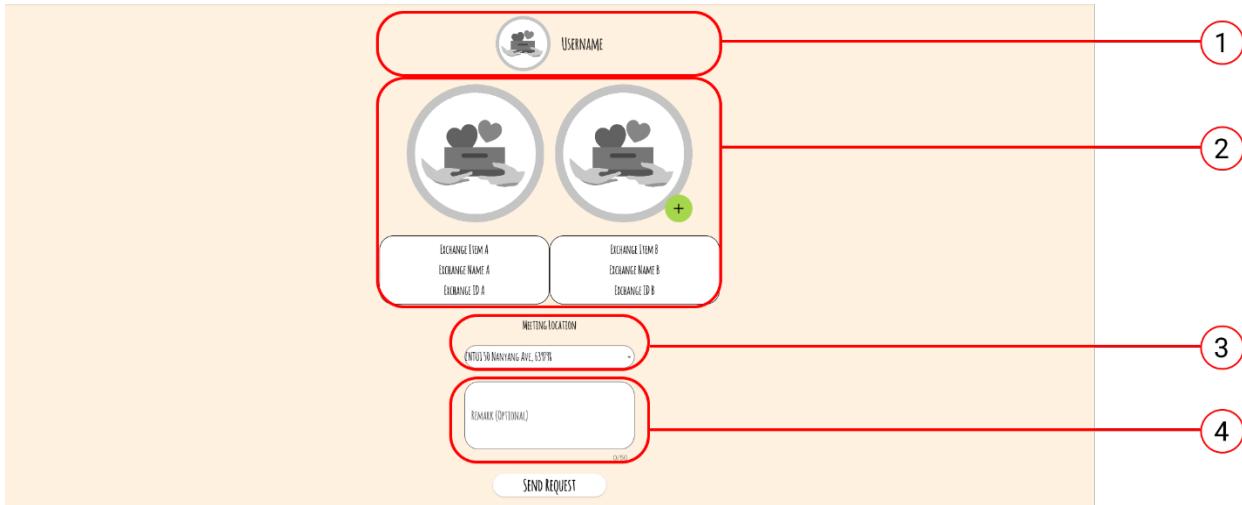


Fig 67, Offer exchange item screen

1. This describes the exchanger information
2. This describes the details of other exchange items, the user is required to add their exchange item.
3. The user must input a preferred location by the other exchange users.
4. For special arrangements, users can put a remark here.

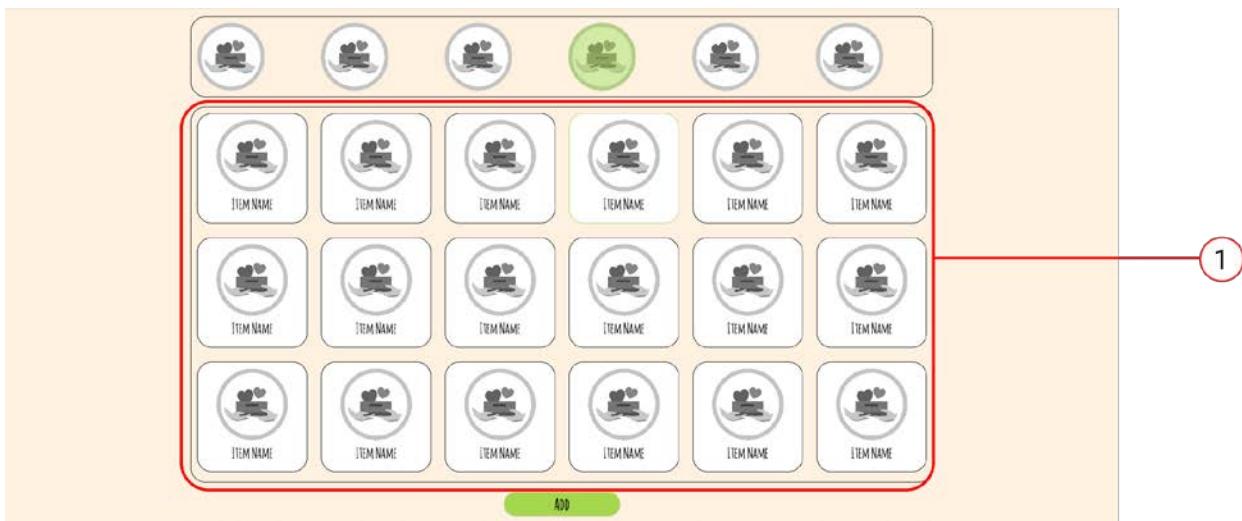


Fig 68, Add exchange item to offer screen

1. This is the list of exchange items current users have entered.

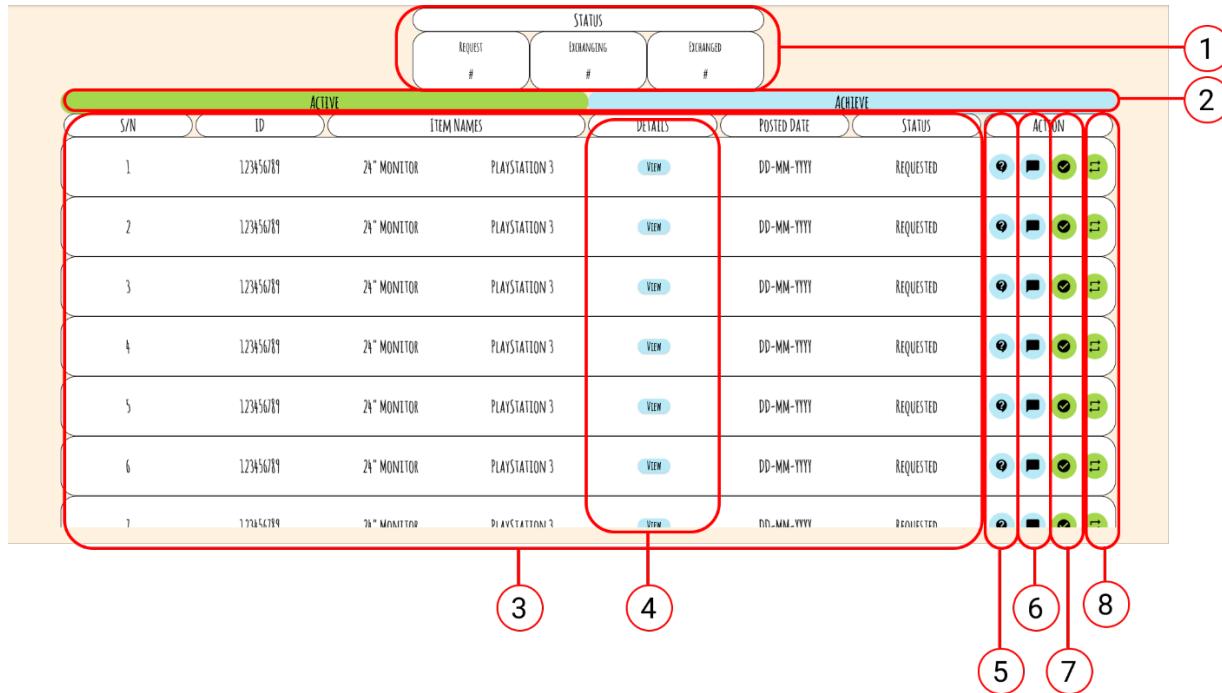


Fig 69, Exchange status screen

1. Once the exchange users have offered a request status should be increment by 1 in the status. The other exchange users are required to accept the offer first before exchange.
2. This toggle between past and present exchange item details.
3. This is to show the details about the exchange items.
4. This further expands the details about the exchange items.
5. If users have any other doubts or questions, they can ask the other exchange users.
6. Alternatively, the user can choose to further chat with them.
7. This is the accept icon, this will only appear for the requested item, once accepted it will turn into exchanging status.
8. This is the icon to exchange when both meet.

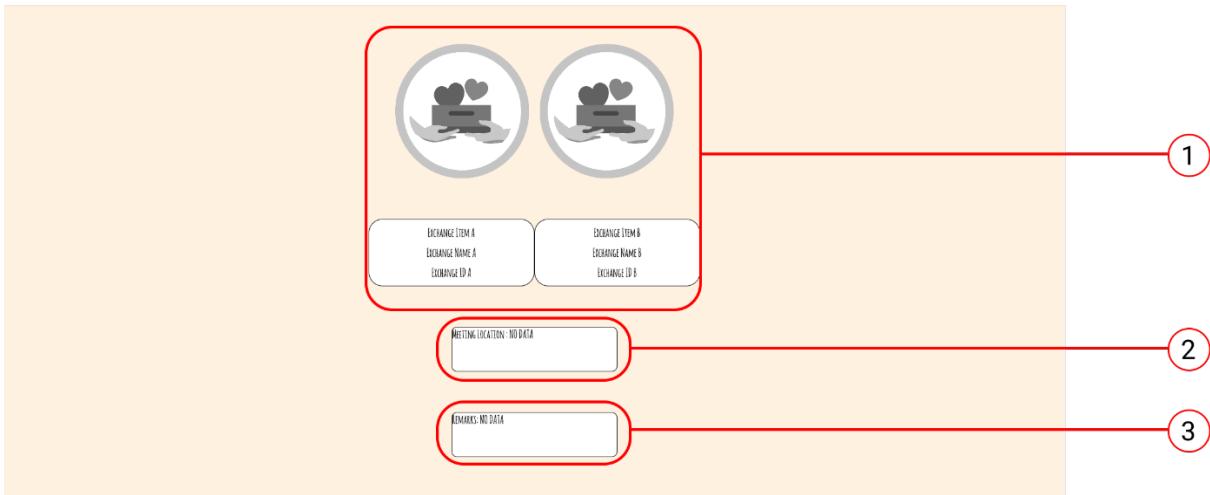


Fig 70, View exchanging or exchanged screen

1. This shows the details of bot exchange items
2. This shows the meeting location
3. This shows the text of the remarks if there is any

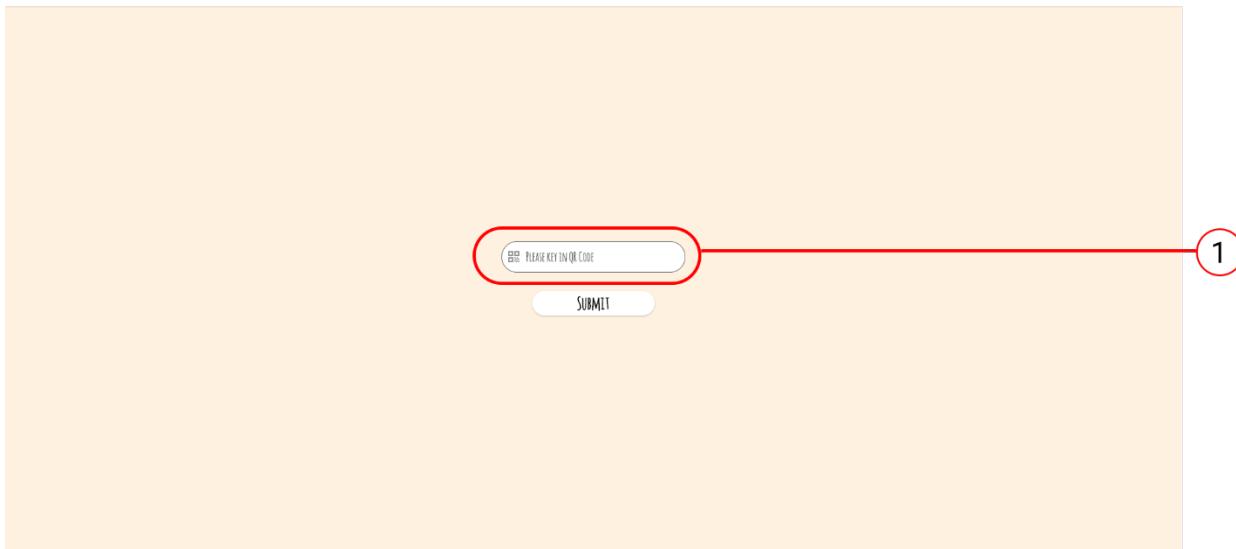


Fig 71, QR scan / QR code screen

1. During the meeting both users are supposed to scan each other QR codes via phone, in the event they do not have a phone QR scanner, they can use the web version to type in the QR code to exchange.

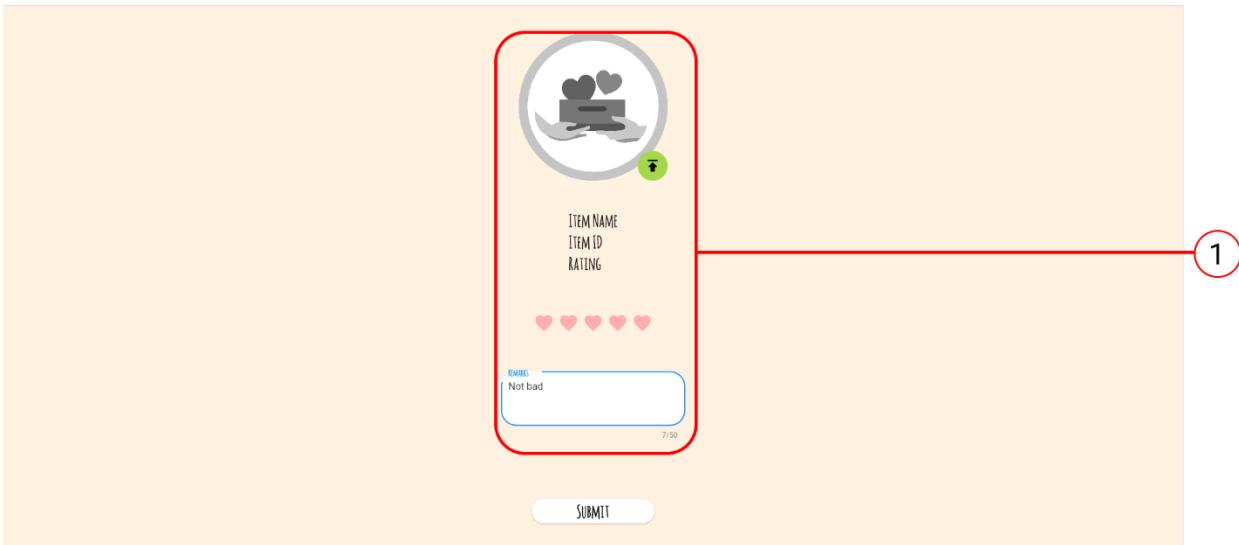


Fig 72, Exchange feedback screen

1. After exchange users will require to do exchange feedback to do the next exchange.

Company UI



Fig 73, Company main screen

1. This is the header of the company main screen with only a single button.
2. This is to keep track of the status of the registered need.
3. This shows the lists of registered needs details.
4. This is the required information to register a needed item
5. This is to prompt confirmation to re-confirm with the user to prevent false decisions.
6. This will register a needed item into the system.

Administrator UI (Web Only)



Fig 74, Administrator header of the screen

1. This is the header of the administrator screen, which is shared with other screens.
2. This is the main page of the administrator where it shows the map with interactive pin-points to monitor the changes or spot abnormality.
3. This is to access the newly join companies list.
4. This is to access the unapproved needs from the registered companies.
5. This is to access the collector management screen.
6. This is to access images from normal users to monitor any abnormality or conduct manual filtering.
7. This manages the feedback about the applications such that the administrator can RSVP to the developer for improvement and exchange of new ideas.

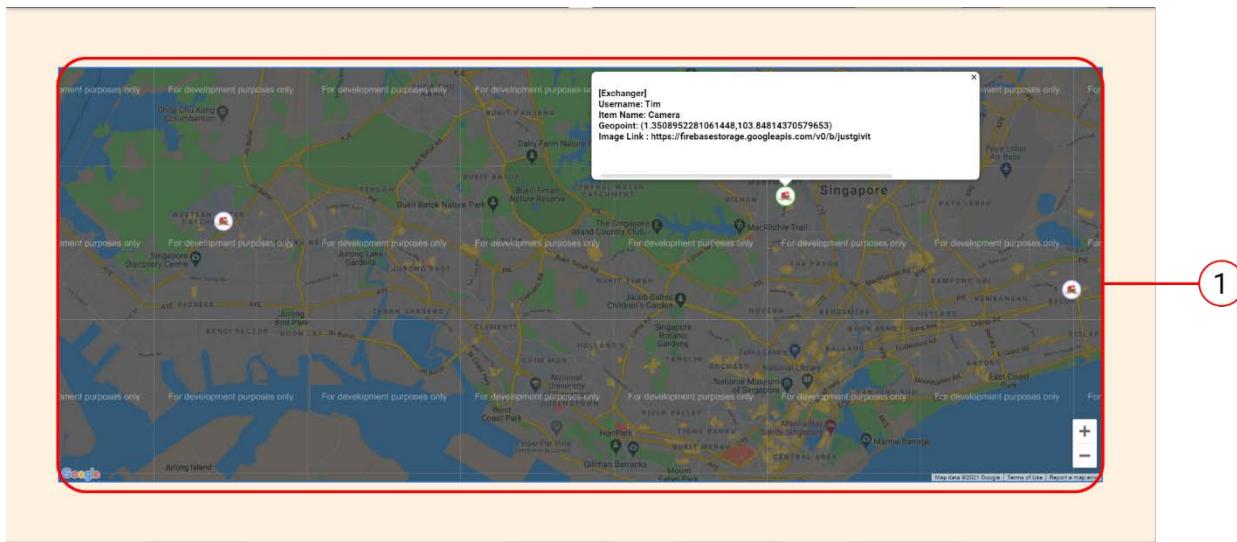


Fig 75, Monitor screen

1. This should show a Singapore map with donate and exchange pin-points.

| S/N | COMPANY LOGO | COMPANY NAME | DESCRIPTION | ADDRESS | CONTACT | ACTION |
|-----|--------------|---------------------|--------------------------------------------------------------------|--------------|---------|-----------------------------------------------------------------------------------|
| 1 | | CHARITY ORG PTE LTD | AAAAAAAAAAAAAaaaaaaaA BLK123 BISHAN AVE 1 #01-1111 S(123456) | 65 123456789 | | ACCEPT REJECT |
| 2 | | CHARITY ORG PTE LTD | AAAAAAAAAAAAAaaaaaaaA BLK123 BISHAN AVE 1 #01-1111 S(123456) | 65 123456789 | | ACCEPT REJECT |
| 3 | | CHARITY ORG PTE LTD | AAAAAAAAAAAAAaaaaaaaA BLK123 BISHAN AVE 1 #01-1111 S(123456) | 65 123456789 | | ACCEPT REJECT |
| 4 | | CHARITY ORG PTE LTD | AAAAAAAAAAAAAaaaaaaaA BLK123 BISHAN AVE 1 #01-1111 S(123456) | 65 123456789 | | ACCEPT REJECT |
| 5 | | CHARITY ORG PTE LTD | AAAAAAAAAAAAAaaaaaaaA BLK123 BISHAN AVE 1 #01-1111 S(123456) | 65 123456789 | | ACCEPT REJECT |
| 6 | | CHARITY ORG PTE LTD | AAAAAAAAAAAAAaaaaaaaA BLK123 BISHAN AVE 1 #01-1111 S(123456) | 65 123456789 | | ACCEPT REJECT |
| 7 | | CHARITY ORG PTE LTD | AAAAAAAAAAAAAaaaaaaaA BLK123 BISHAN AVE 1 #01-1111 S(123456) | 65 123456789 | | ACCEPT REJECT |

Fig 76, Company management screen

1. This shows the company details, the administrator may choose to verify them via contacting them.
2. This is to approve or reject the company being not faithful or fake.

COLLECTOR

USERNAME

FULL NAME

CONTACT

ADDRESS AVENUE

UNIT NUMBER

POSTAL CODE

ADD

| | | | | | | | | | |
|--|----------------------------------|--|----------------------------------|--|----------------------------------|--|----------------------------------|--|----------------------------------|
| | USERNAME FULL NAME CONTACT |
| | USERNAME FULL NAME CONTACT |
| | USERNAME FULL NAME CONTACT |
| | USERNAME FULL NAME CONTACT |
| | USERNAME FULL NAME CONTACT |

Fig 77, Collector management screen

1. This is the information required to register a collector into the system.
2. This is the list of registered collector and they can be selected and removed.

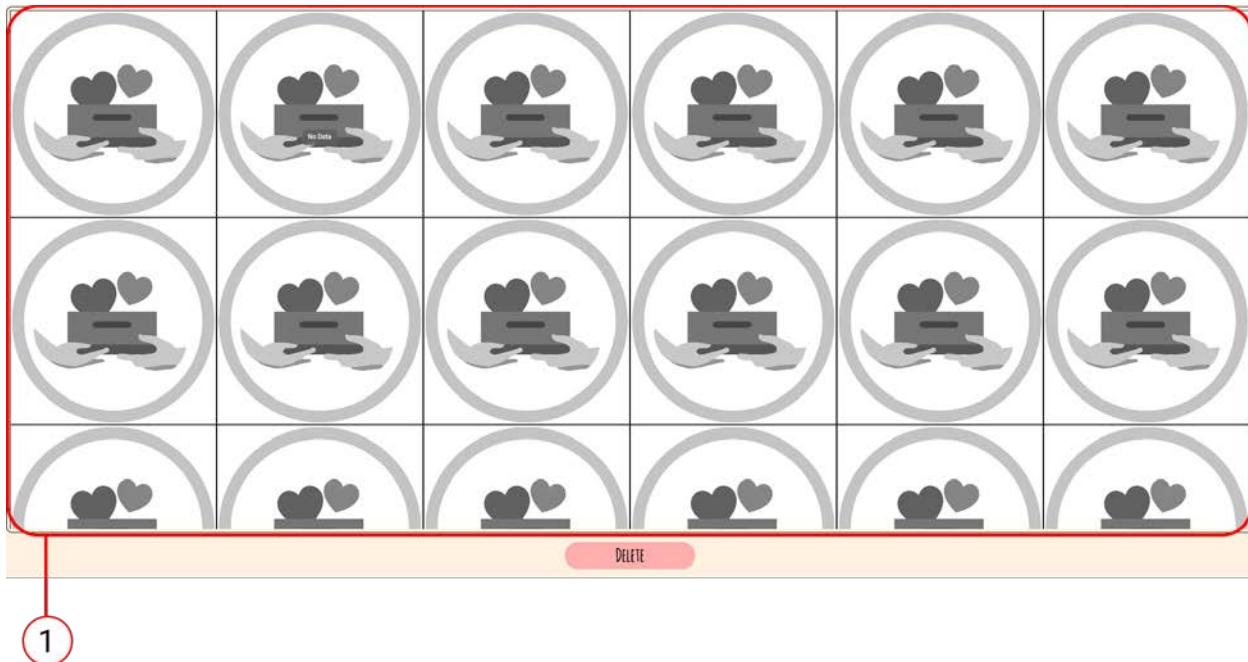


Fig 78, normal users' images management

1. This shows all the donate or exchange images about the normal users. It can be removed by the administrator if it is deemed wrong or abnormal.

| S/N | ITEM TYPE | ITEM NAME | QUANTITIES | COMPANY NAME | CONTACT | ACTION |
|-----|-----------|-----------|------------|-----------------|--------------|-----------------------------------------------------------------------------------|
| 1 | | NEWSPAPER | 100 | CHARITY PTE LTD | 65 123456789 | ACCEPT REJECT |
| 2 | | NEWSPAPER | 100 | CHARITY PTE LTD | 65 123456789 | ACCEPT REJECT |
| 3 | | NEWSPAPER | 100 | CHARITY PTE LTD | 65 123456789 | ACCEPT REJECT |
| 4 | | NEWSPAPER | 100 | CHARITY PTE LTD | 65 123456789 | ACCEPT REJECT |
| 5 | | NEWSPAPER | 100 | CHARITY PTE LTD | 65 123456789 | ACCEPT REJECT |
| 6 | | NEWSPAPER | 100 | CHARITY PTE LTD | 65 123456789 | ACCEPT REJECT |
| 7 | | NEWSPAPER | 100 | CHARITY PTE LTD | 65 123456789 | ACCEPT REJECT |

Fig 79, Needs management screen

1. This shows the companies registered needs in detail, the administrator may choose to verify them via contacting them.
2. This is to approve or reject the company requested needs.

| S/N | GENERATED ID | RATING | REMARK | PUBLISHED DATE | ACTION |
|-----|--------------|--------|---------------------------------------------------------------------------------------|----------------|---------------|
| 1 | 123456789 | ★★★☆☆ | AAAAAAAAAAAAA.....AAAAA AAAAAAA.....AAAAA.....AAAAA AAAAAAA.....AAAAA.....AAAAA | DD-MM-YYYY | ACCEPT REJECT |
| 2 | 123456789 | ★★★★☆ | AAAAAAA.....AAAAA.....AAAAA AAAAAAA.....AAAAA.....AAAAA | DD-MM-YYYY | ACCEPT REJECT |
| 3 | 123456789 | ★★★★☆ | AAAAAAA.....AAAAA.....AAAAA AAAAAAA.....AAAAA.....AAAAA | DD-MM-YYYY | ACCEPT REJECT |
| 4 | 123456789 | ★★★★☆ | AAAAAAA.....AAAAA.....AAAAA AAAAAAA.....AAAAA.....AAAAA | DD-MM-YYYY | ACCEPT REJECT |
| 5 | 123456789 | ★★★★☆ | AAAAAAA.....AAAAA.....AAAAA AAAAAAA.....AAAAA.....AAAAA | DD-MM-YYYY | ACCEPT REJECT |
| 6 | 123456789 | ★★★★☆ | AAAAAAA.....AAAAA.....AAAAA AAAAAAA.....AAAAA.....AAAAA | DD-MM-YYYY | ACCEPT REJECT |
| 7 | 123456789 | ★★★★☆ | AAAAAAA.....AAAAA.....AAAAA AAAAAAA.....AAAAA.....AAAAA | DD-MM-YYYY | ACCEPT REJECT |

1

2

Fig 80, Application feedback review screen

1. This shows the application feedback in detail.
2. This is to approve or reject the feedback needs. Approved feedback will be published to the public for other public users to consider before using the JustGivit system. Rejected feedback will be achieved.

Test Cases

Partial test cases will be conducted to ensure core functionality is met properly base on the functional requirements.

| S/N | Area | Name | Input | Expect Output | Actual Output (Platform) | |
|-----|-----------------------|-------------------|-----------|-------------------------------------------|-------------------------------------------|-------------------------------------------|
| | | | | | Android | Web |
| 1 | Register (Company) | Contact Number | | “Please Enter Contact Number” | “Please Enter Contact Number” | “Please Enter Contact Number” |
| 2 | | | a | “Exactly 8 Character and SG phone format” | <FORBIDDEN> | “Exactly 8 Character and SG phone format” |
| 3 | | | A | “Exactly 8 Character and SG phone format” | <FORBIDDEN> | “Exactly 8 Character and SG phone format” |
| 4 | | | A1 | “Exactly 8 Character and SG phone format” | <FORBIDDEN> | “Exactly 8 Character and SG phone format” |
| 5 | | | . | “Exactly 8 Character and SG phone format” | “Exactly 8 Character and SG phone format” | “Exactly 8 Character and SG phone format” |
| 6 | | | 1234567 | “Exactly 8 Character and SG phone format” | “Exactly 8 Character and SG phone format” | “Exactly 8 Character and SG phone format” |
| 7 | | | 12345678 | “Exactly 8 Character and SG phone format” | “Exactly 8 Character and SG phone format” | “Exactly 8 Character and SG phone format” |
| 8 | | | 123456789 | “Exactly 8 Character and SG phone format” | “Exactly 8 Character and SG phone format” | “Exactly 8 Character and SG phone format” |
| 9 | | | 61234567 | “Valid Phone Number” | “Valid Phone Number” | “Valid Phone Number” |
| 10 | | | 91234567 | “Valid Phone Number” | “Valid Phone Number” | “Valid Phone Number” |
| 11 | | | 81234567 | “Valid Phone Number” | “Valid Phone Number” | “Valid Phone Number” |
| 12 | | Password | | “Please Enter Your Password” | “Please Enter Your Password” | “Please Enter Your Password” |
| 13 | | | a | “Minimum 8 characters” | “Minimum 8 characters” | “Minimum 8 characters” |

| | | | | | | |
|----|----------------------------|----------------|-----------|----------------------------------|----------------------------------|----------------------------------|
| 14 | | | 1234567 | “Minimum 8 characters” | “Minimum 8 characters” | “Minimum 8 characters” |
| 15 | | | 12345678 | “At least one capital letter” | “At least one capital letter” | “At least one capital letter” |
| 16 | | | a1234567 | “At least one capital letter” | “At least one capital letter” | “At least one capital letter” |
| 17 | | | abcdefgh | “At least one capital letter” | “At least one capital letter” | “At least one capital letter” |
| 18 | | | A1234567 | “Valid Password” | “Valid Password” | “Valid Password” |
| 19 | | | A12345679 | “Valid Password” | “Valid Password” | “Valid Password” |
| 20 | Register (Normal) | User name | | “Please Enter Your Username” | “Please Enter Your Username” | “Please Enter Your Username” |
| 21 | | | a | “Minimum 5 Characters” | “Minimum 5 Characters” | “Minimum 5 Characters” |
| 22 | | | 1 | “No Digits or Special Character” | “No Digits or Special Character” | “No Digits or Special Character” |
| 23 | | | @ | “No Digits or Special Character” | “No Digits or Special Character” | “No Digits or Special Character” |
| 24 | | | 1234 | “Alphabets character only” | “Alphabets character only” | “Alphabets character only” |
| 25 | | | Abcd1 | “No Digits or Special Character” | “No Digits or Special Character” | “No Digits or Special Character” |
| 26 | | | Abcde\$ | “No Digits or Special Character” | “No Digits or Special Character” | “No Digits or Special Character” |
| 27 | | | Abcde | “Valid Username” | “Valid Username” | “Valid Username” |
| 28 | | | Abcdef | “Valid Username” | “Valid Username” | “Valid Username” |
| 29 | Donate (Give Screen) | Postal Code | | “Postal Code” | “Postal Code” | “Postal Code” |
| 30 | | | a | “Invalid Postal Code” | <FORBIDDEN> | “Invalid Postal Code” |
| 31 | | | 1 | “Invalid Postal Code” | “Invalid Postal Code” | “Invalid Postal Code” |
| 32 | | | a1 | “Invalid Postal Code” | <FORBIDDEN> | “Invalid Postal Code” |
| 33 | | | 12345 | “Invalid Postal Code” | “Invalid Postal Code” | “Invalid Postal Code” |
| 34 | | | 1234567 | “Invalid Postal Code” | “Invalid Postal Code” | “Invalid Postal Code” |
| 35 | | | 123456 | “Valid Postal Code” | “Valid Postal Code” | “Valid Postal Code” |

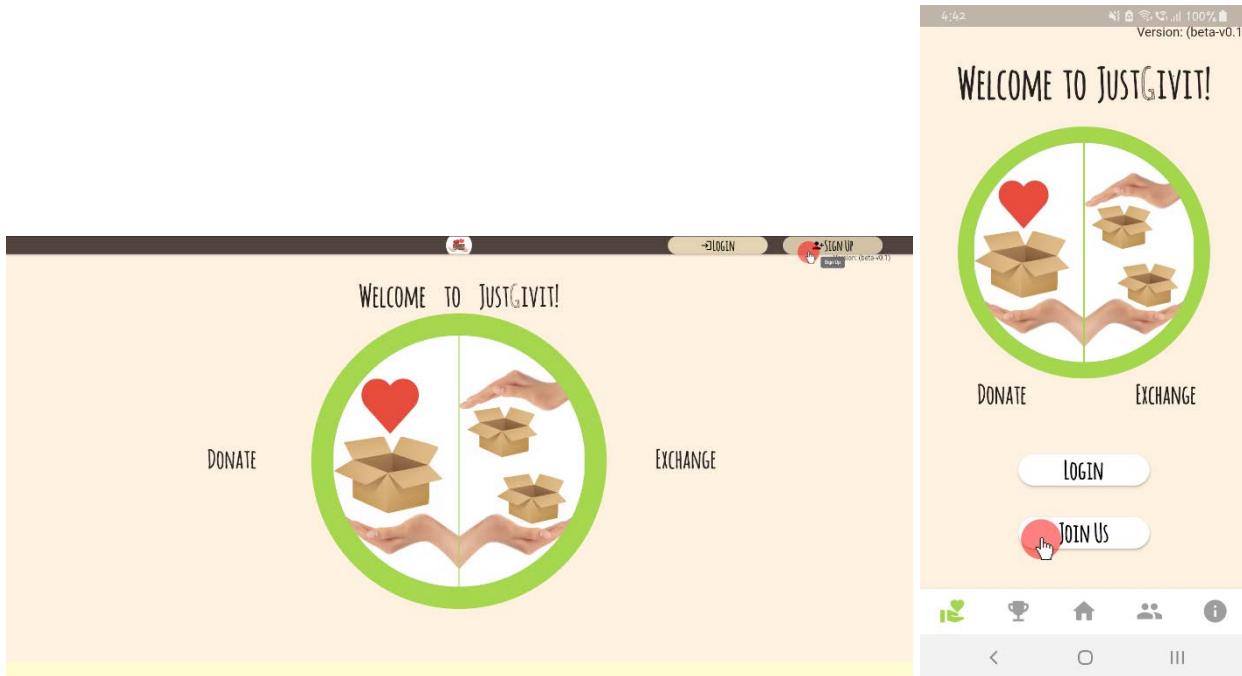
| | | | | | |
|-----------|-------------|----------|-----------------------|-----------------------|-----------------------|
| 36 | Unit Number | | “Unit Number” | “Unit Number” | “Unit Number” |
| 37 | | a | “Invalid Unit Number” | “Invalid Unit Number” | “Invalid Unit Number” |
| 38 | | @ | “Invalid Unit Number” | “Invalid Unit Number” | “Invalid Unit Number” |
| 39 | | #01234 | “Invalid Unit Number” | “Invalid Unit Number” | “Invalid Unit Number” |
| 40 | | #1-123 | “Invalid Unit Number” | “Invalid Unit Number” | “Invalid Unit Number” |
| 41 | | #a1-123 | “Invalid Unit Number” | “Invalid Unit Number” | “Invalid Unit Number” |
| 42 | | #01-12 | “Valid Unit Number” | “Valid Unit Number” | “Valid Unit Number” |
| 43 | | #01-123 | “Valid Unit Number” | “Valid Unit Number” | “Valid Unit Number” |
| 44 | | #01-1234 | “Valid Unit Number” | “Valid Unit Number” | “Valid Unit Number” |

Legend: FORBIDDEN – The input is restricted and not visible to the user.

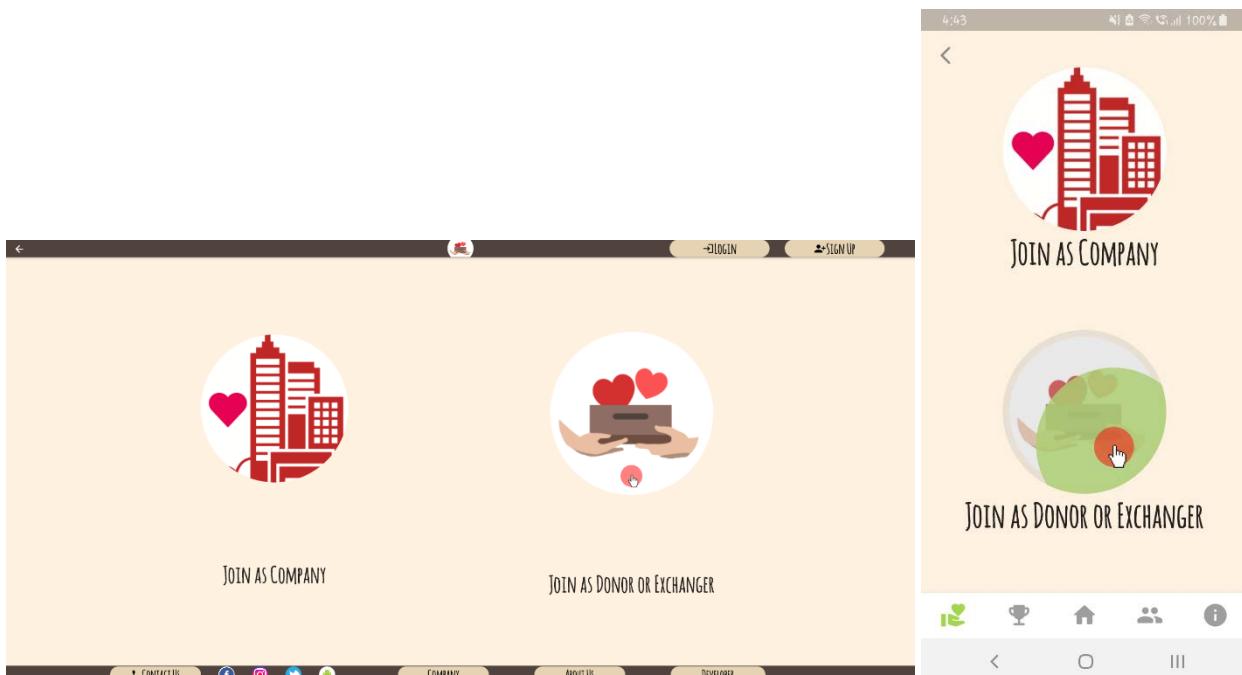
Walkthrough

This section consists of the key walkthrough base on the scope of our project from register a normal account to donate and exchange an item successfully for both mobile and web platforms.

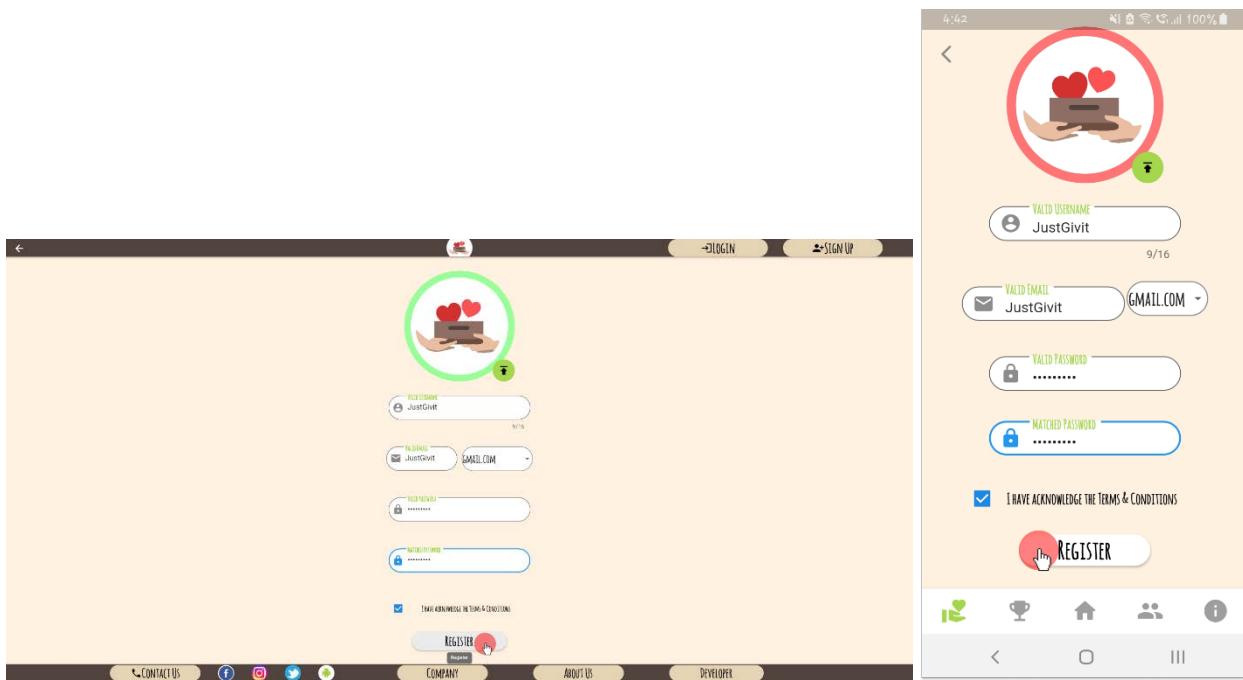
Register an account



Click on register related button

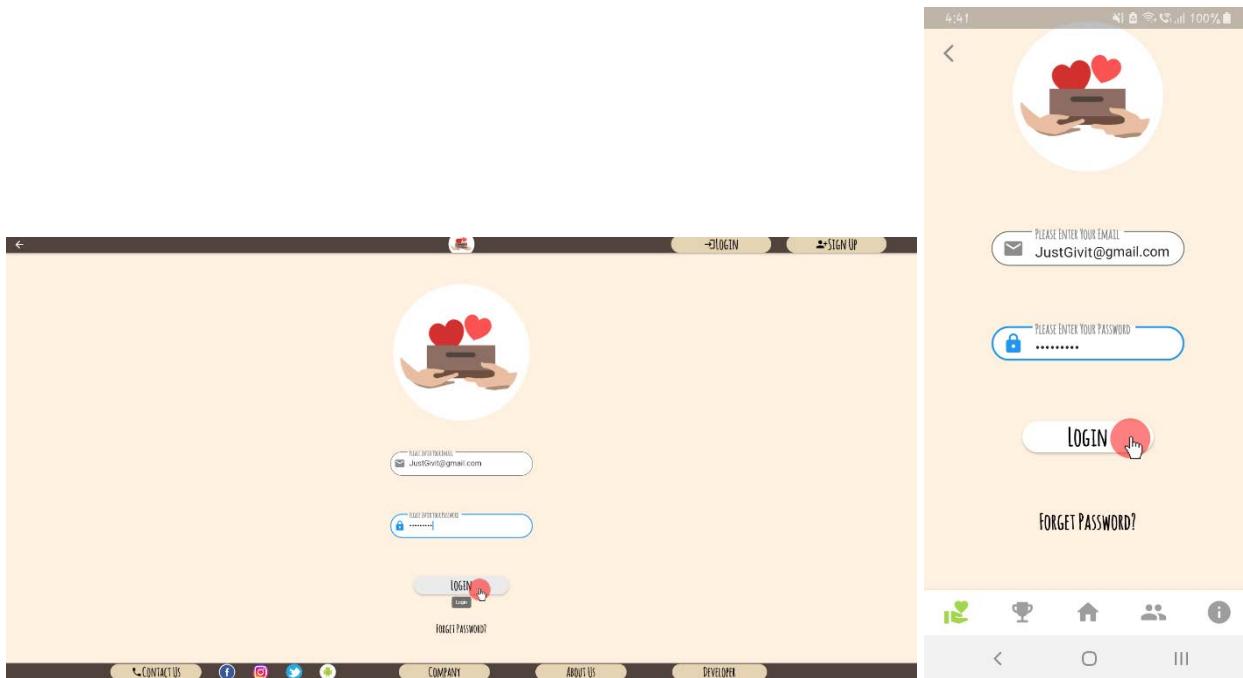


Click on joining as a donor or exchange button



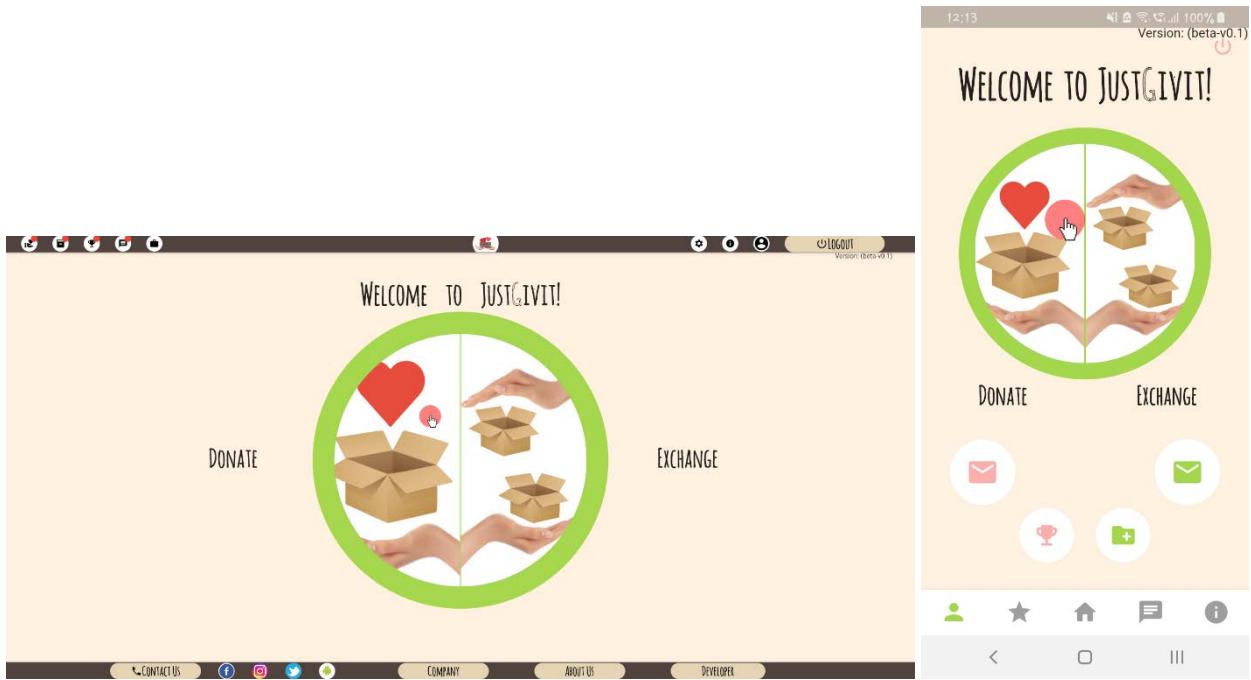
Fill in the inputs correctly

Login the account



Login into the “JustGivit” system

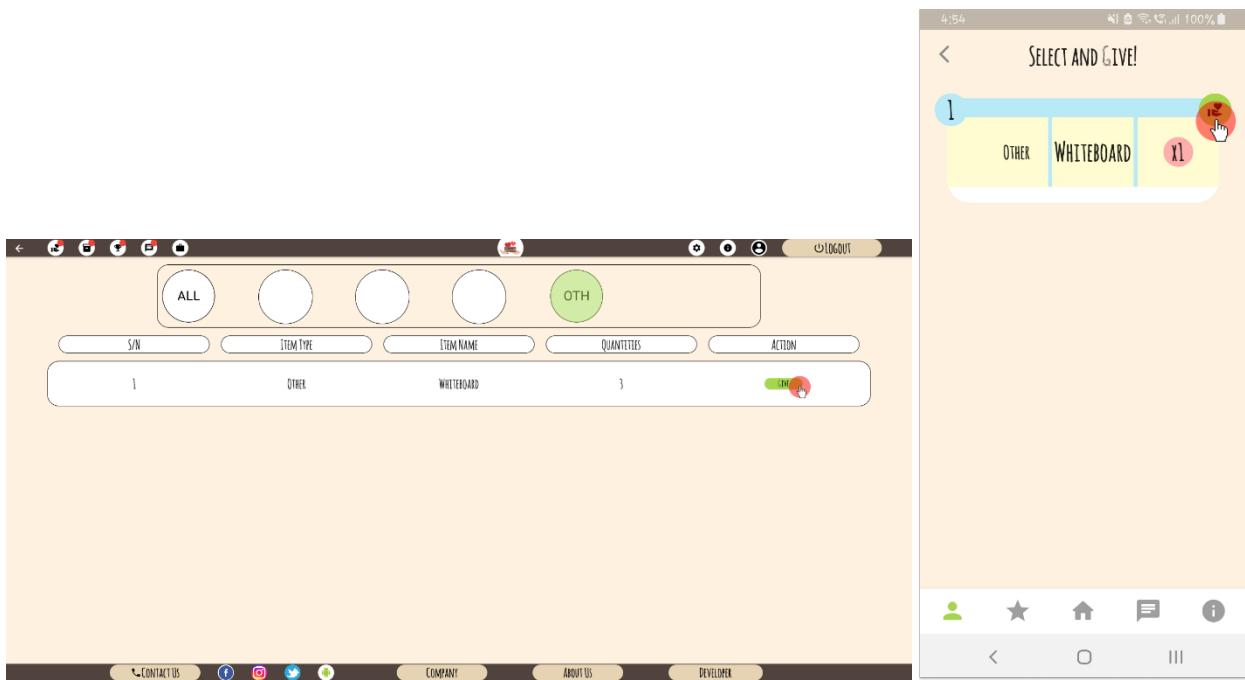
Donate an Item



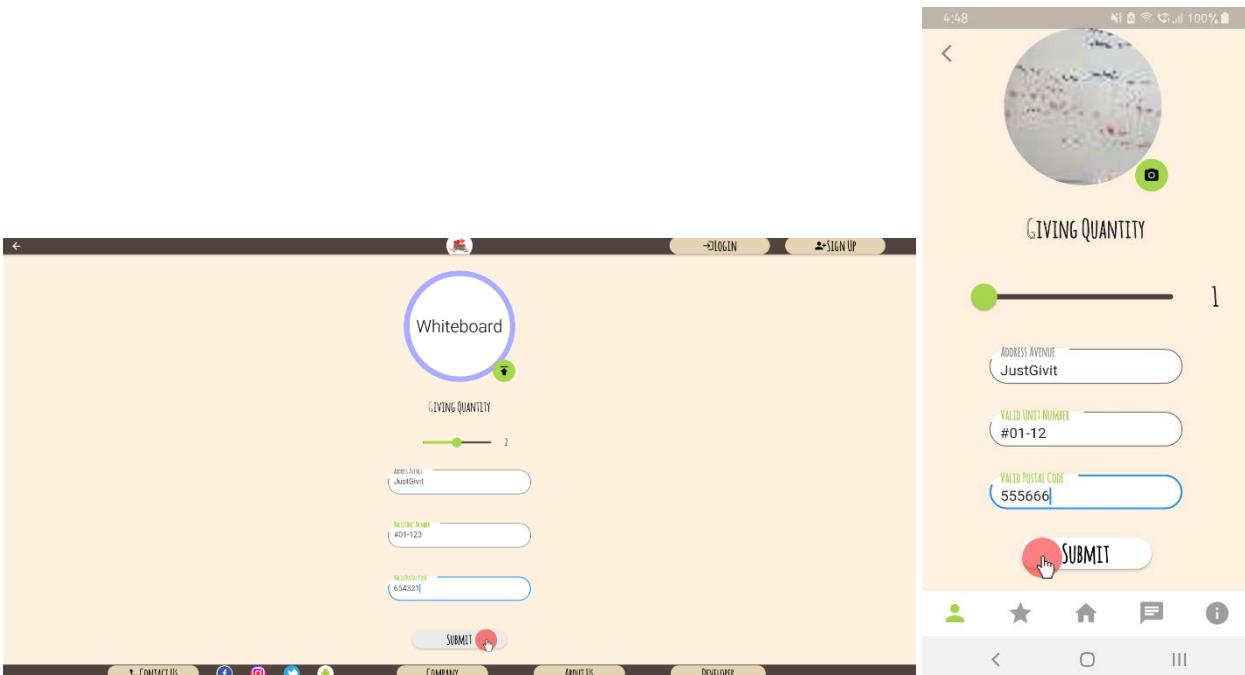
Click on the Donate representation button

| S/N | ITEM TYPE | ITEM NAME | QUANTITIES | ACTION |
|-----|------------|------------|------------|----------------------|
| 1 | PAPER | NEWSPAPER | 5 | <button>ADD</button> |
| 2 | CLOTH | CLOTHES | 17 | <button>ADD</button> |
| 3 | PAPER | NEWSPAPER | 7 | <button>ADD</button> |
| 4 | OTHER | WHITEBOARD | 3 | <button>ADD</button> |
| 5 | CLOTH | SOCKS | 2 | <button>ADD</button> |
| 6 | ELECTRONIC | SPAKER | 5 | <button>ADD</button> |
| 7 | | | | <button>ADD</button> |

Select the item type (In this case 'Other')

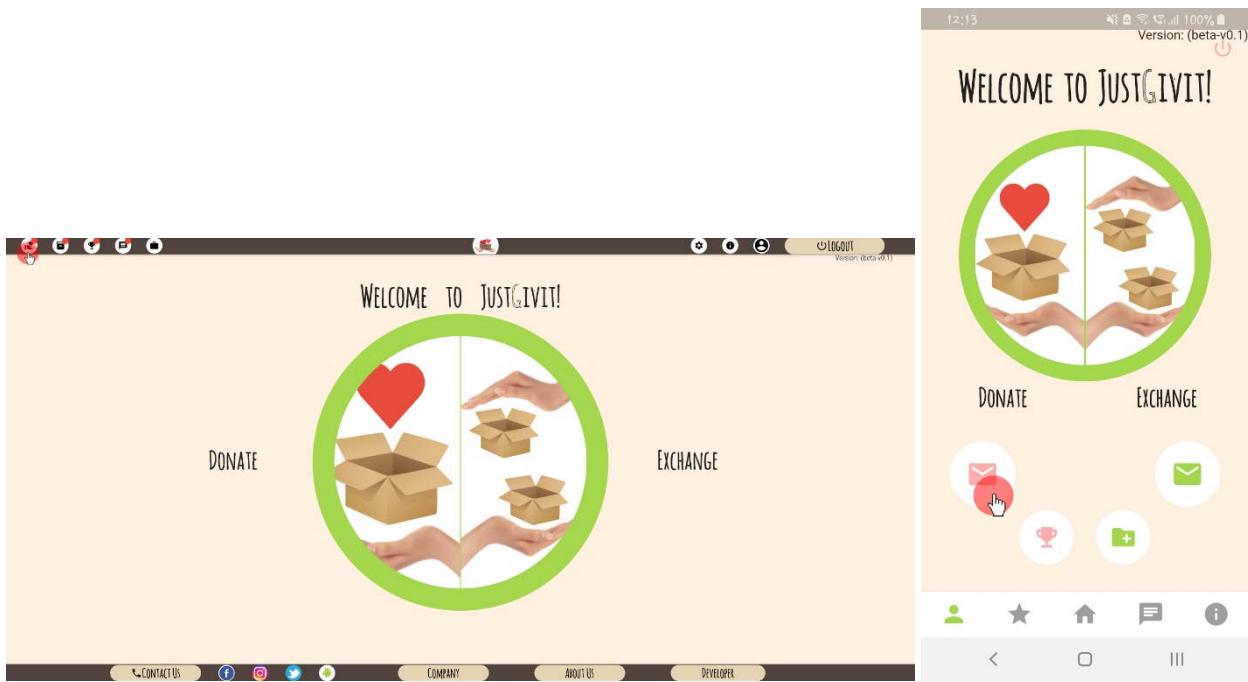


Click on the give representative button



Fill in the giving information and submit

Claimed Reward



Back to the main screen, click on donate status representative button

The screenshot shows the JustGivit application interface. At the top right, it displays the time (4:48), battery level (100%), and version (beta-v0.1). The main title "WELCOME TO JUSTGIVIT!" is centered above a large circular icon. This icon is divided vertically; the left half contains a red heart icon and a box, while the right half shows hands holding boxes. Below the icon, the word "DONATE" is on the left and "EXCHANGE" is on the right. The bottom navigation bar includes links for CONTACT US, COMPANY, ABOUT US, and DEVELOPER.

On the left side of the screen, there is a table with the following data:

| S/P | ACTIVE | ID | ITEM TYPE | ITEM NAME | POSTED DATE | ACHIEVE | STATUS | CLAIM REWARD |
|-----|--------|--------|-----------|------------|---------------------|---------|------------|--------------|
| 1 | ACTIVE | 202108 | OTHER | WHITEBOARD | 2021-04-15T19:29:37 | 0 | COLLECTING | |

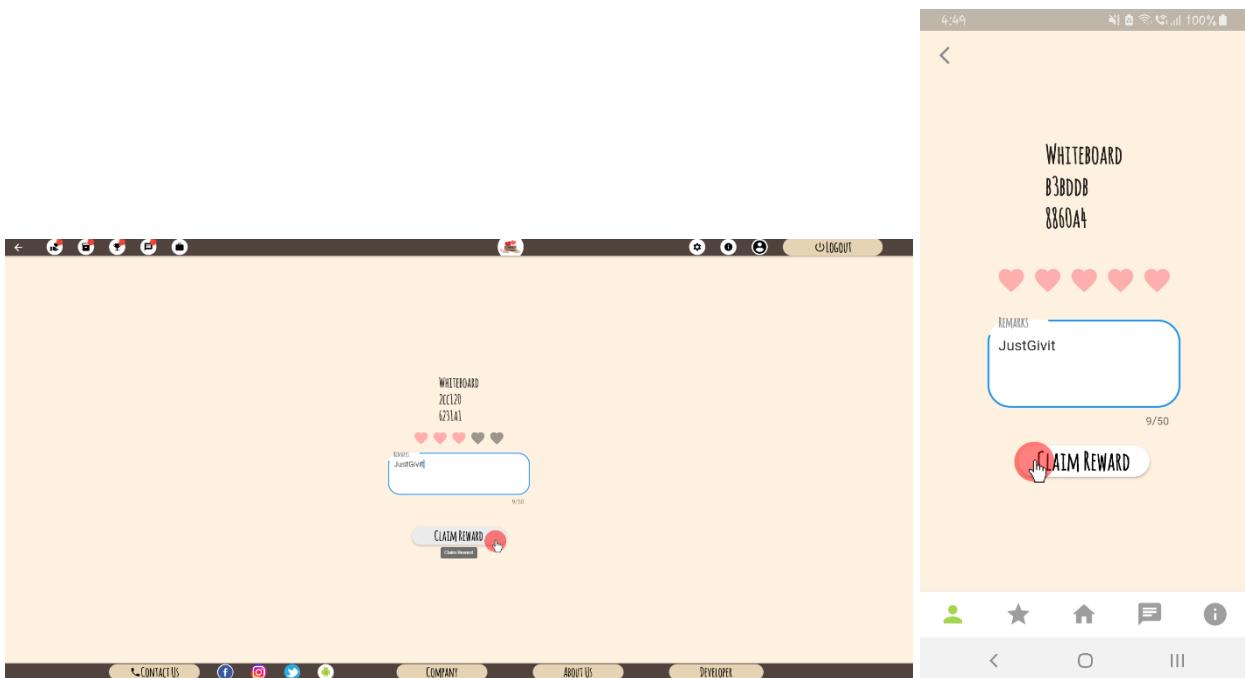
On the right side of the screen, there is a "STATUS" section with two buttons: "COLLECTING" (0) and "COLLECTED" (1). Below this, there is a card with the following information:

1 STATUS: COLLECTED

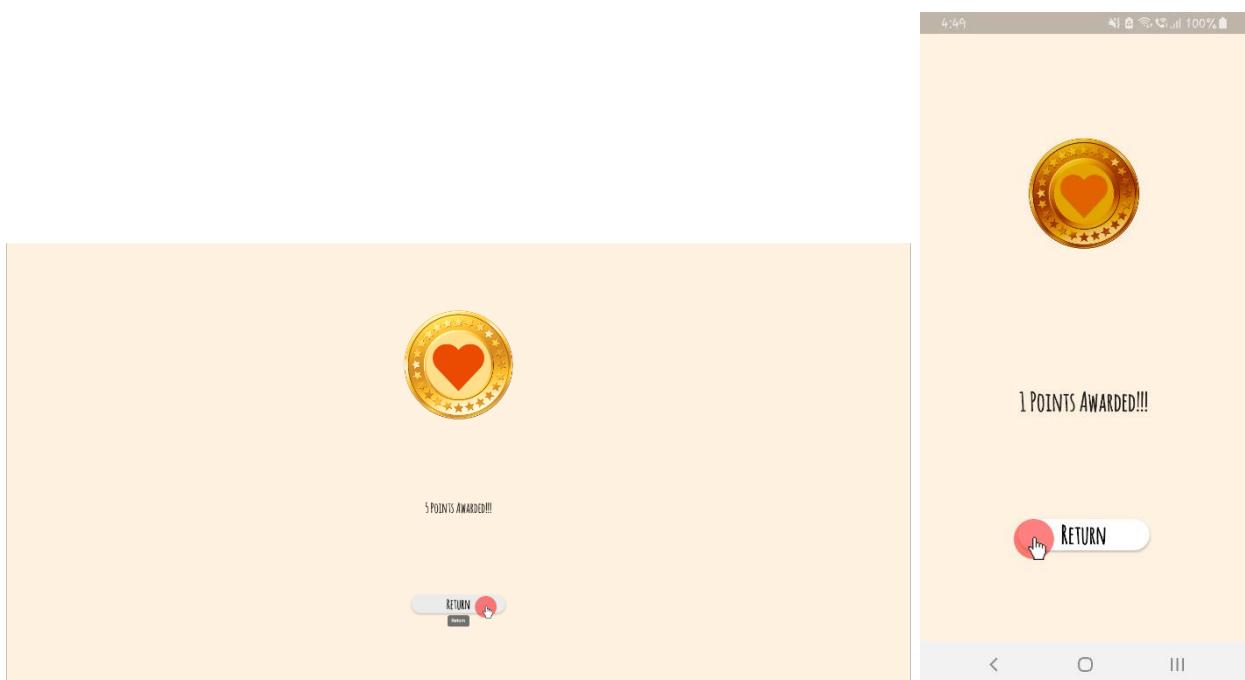
OTHER WHITEBOARD ID: 38008 2021-04-15

At the bottom, there are tabs for "ACTIVE" and "ACHIEVE".

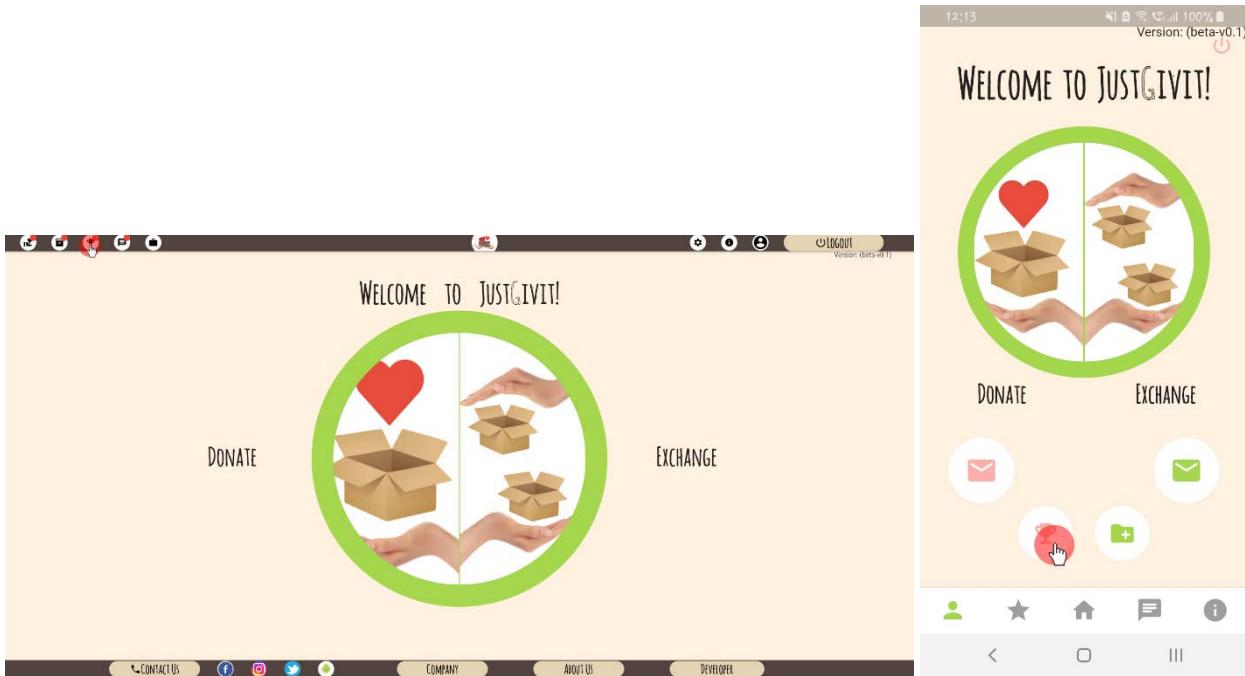
Clicked on the claim representative button



Fill up the collector's feedback and submit



Awarded a reward and click the return button

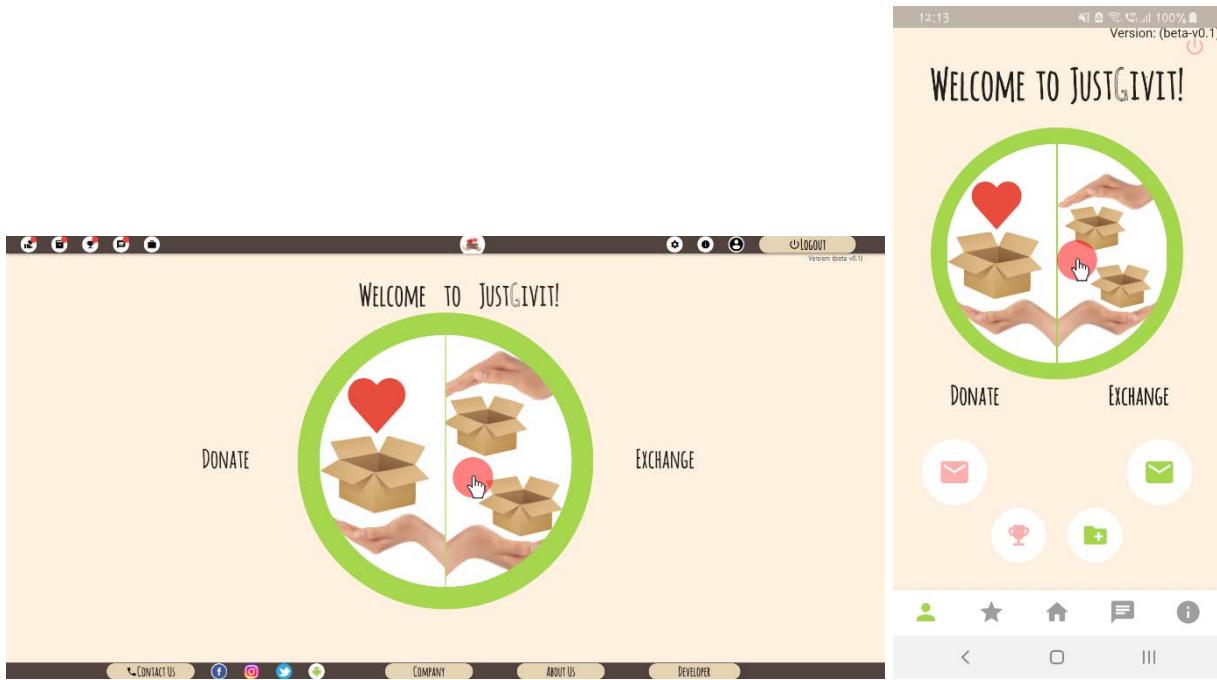


Click on the claim reward representative button

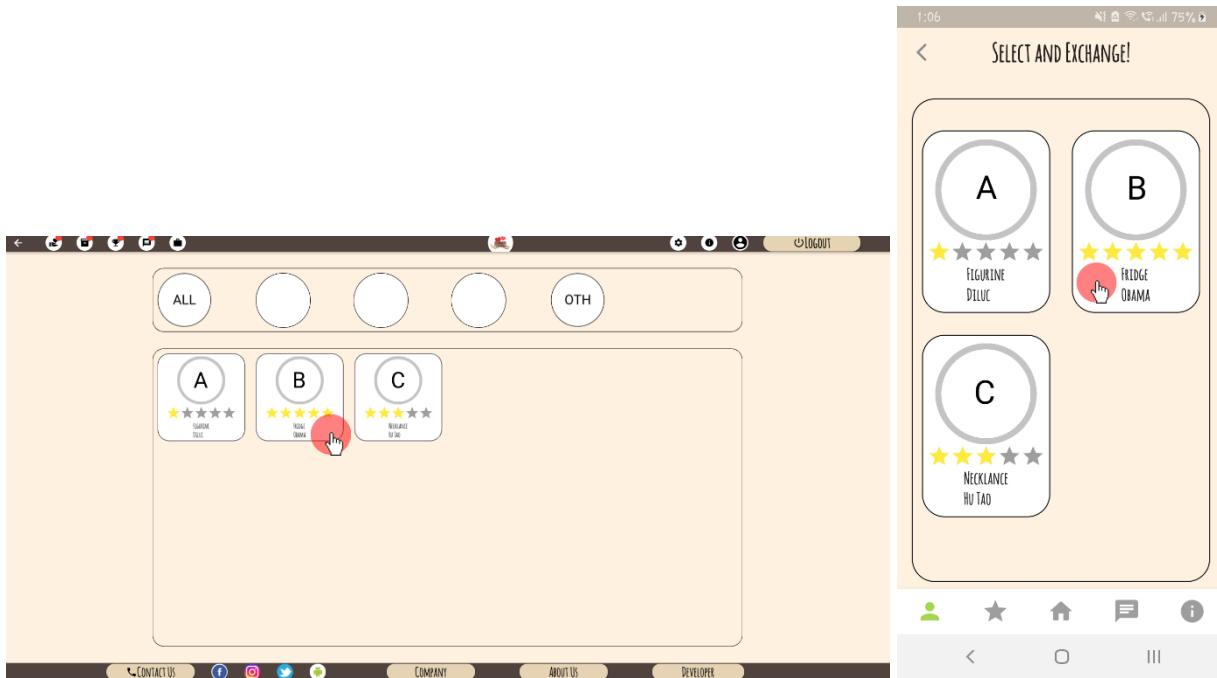


Click on the 'Use to point' button to spin the wheel to earn 'Givit' points!

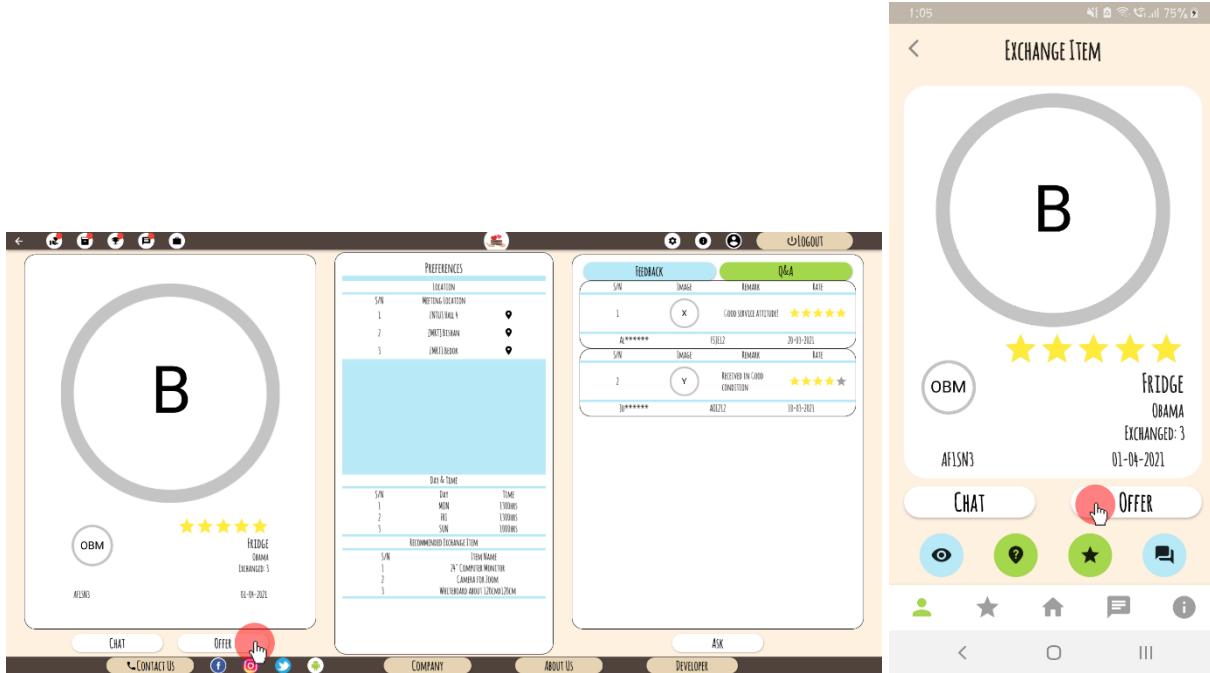
Exchange an item



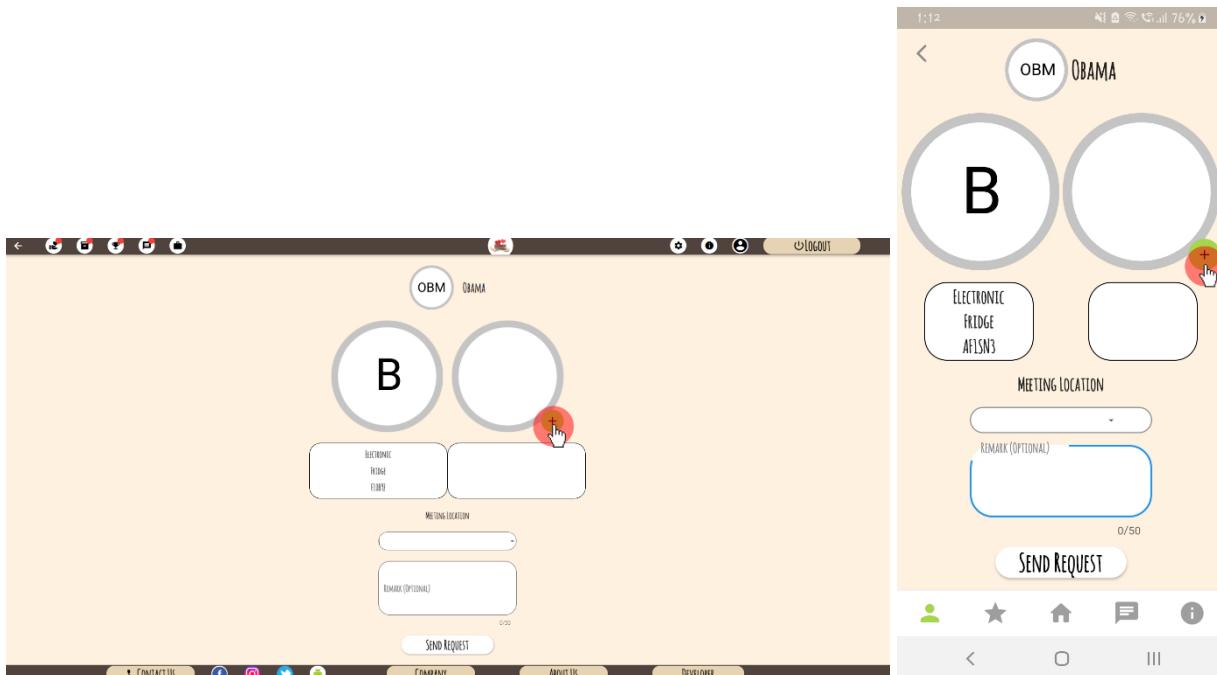
On the main screen, click on the exchange representative button



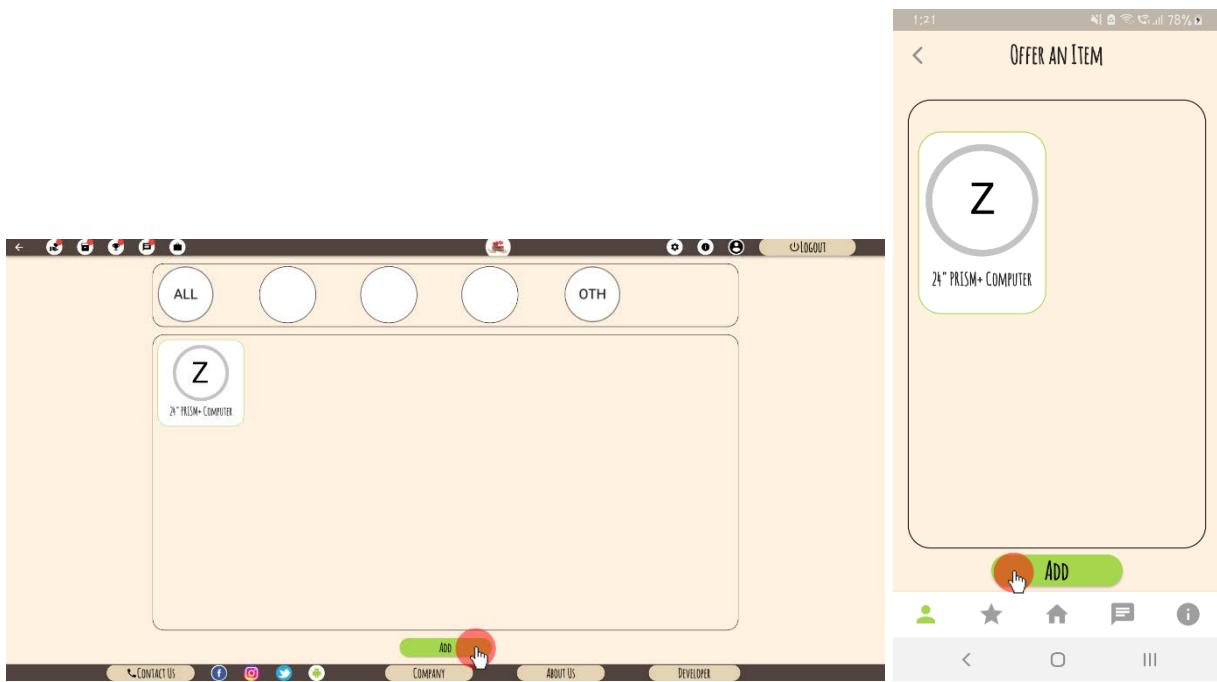
Click on an exchange item



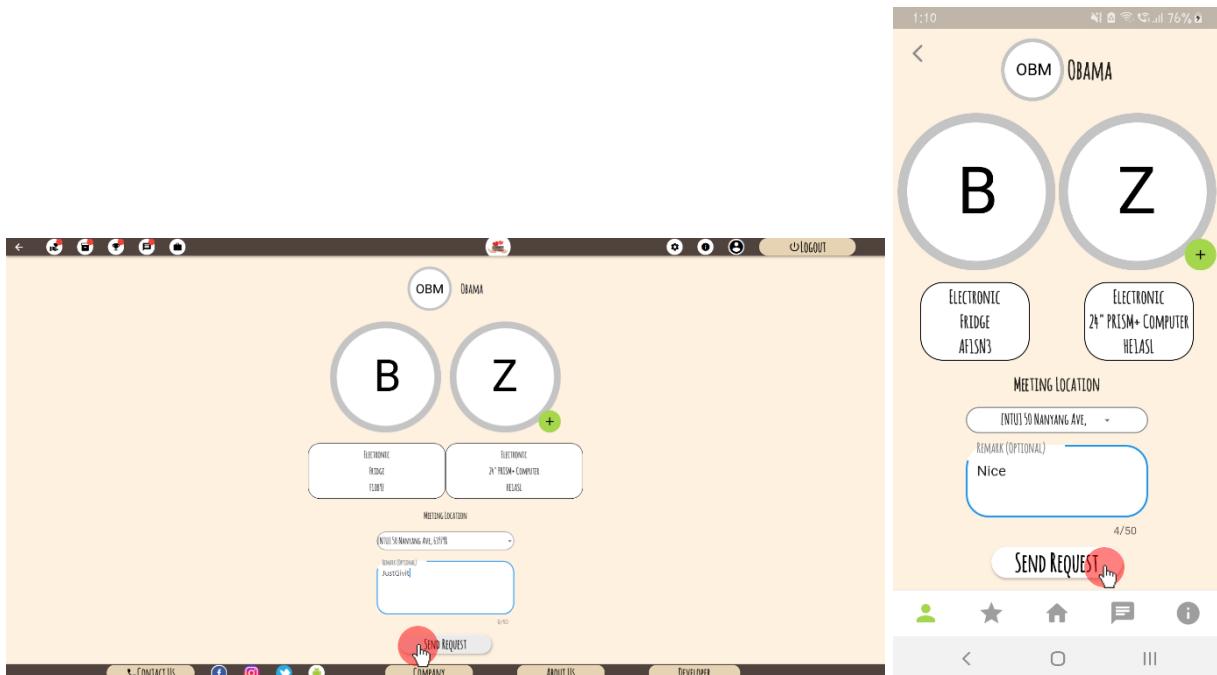
Review and click on 'offer' to offer an exchange item to exchange



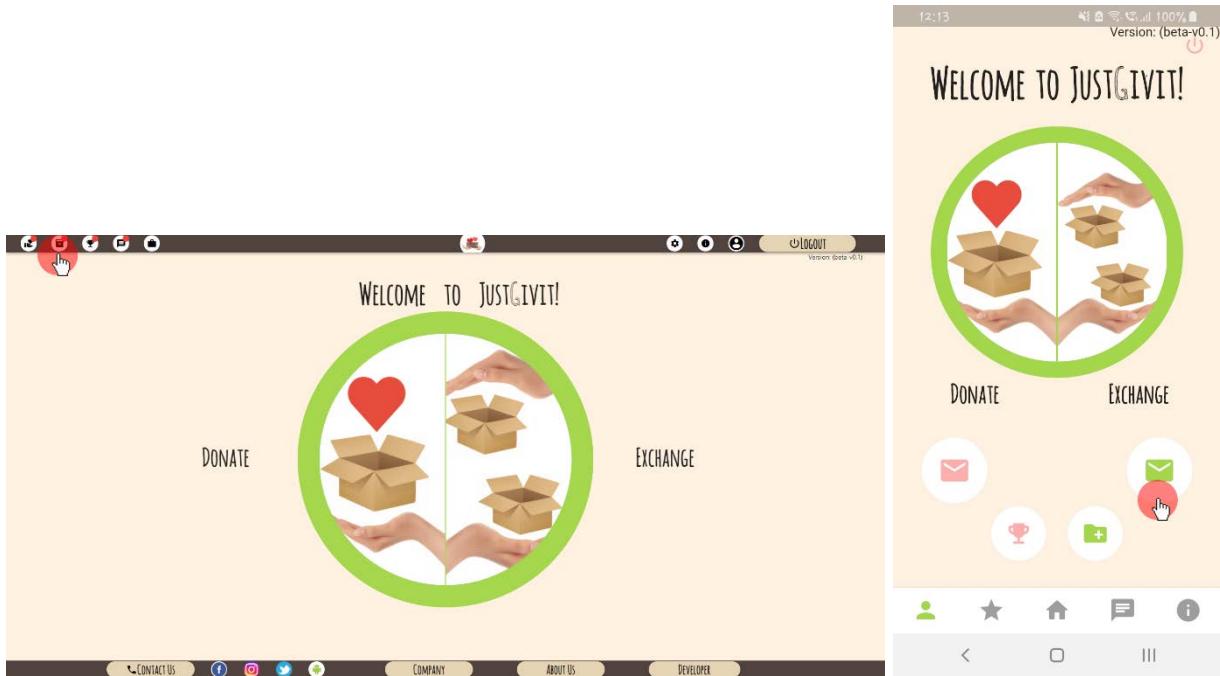
Click on the 'add' exchange representative button



Select an exchange item and click on add representative button



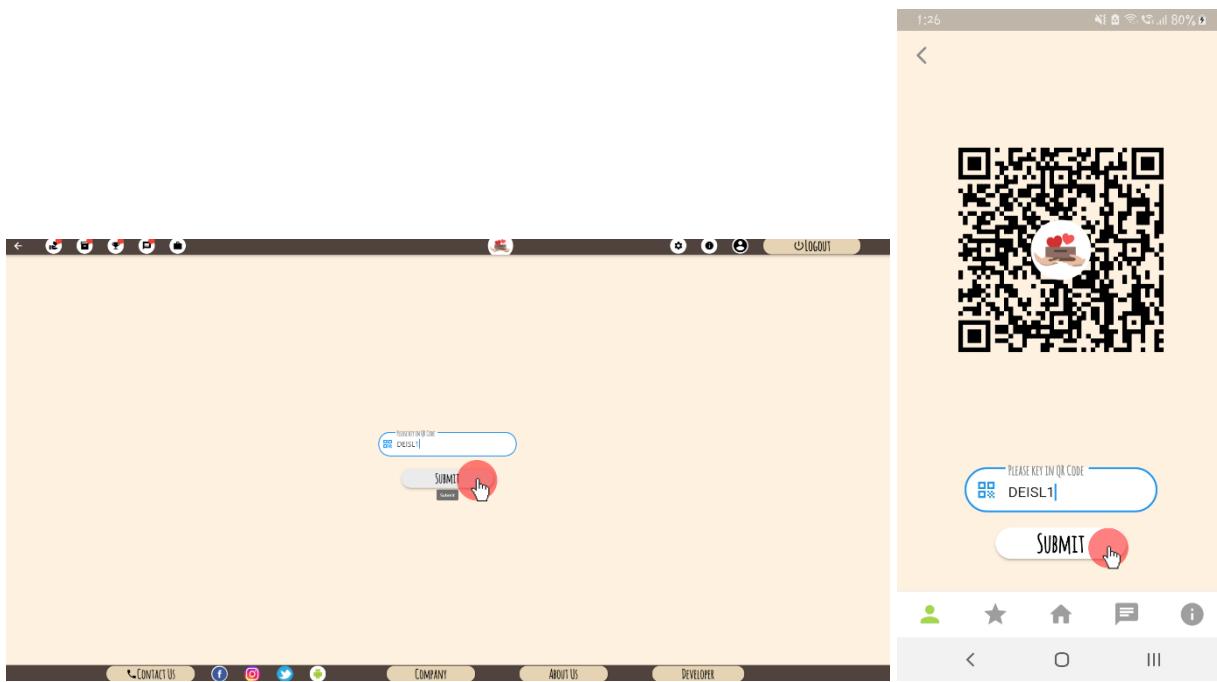
Fill in the necessary information and click on submit button



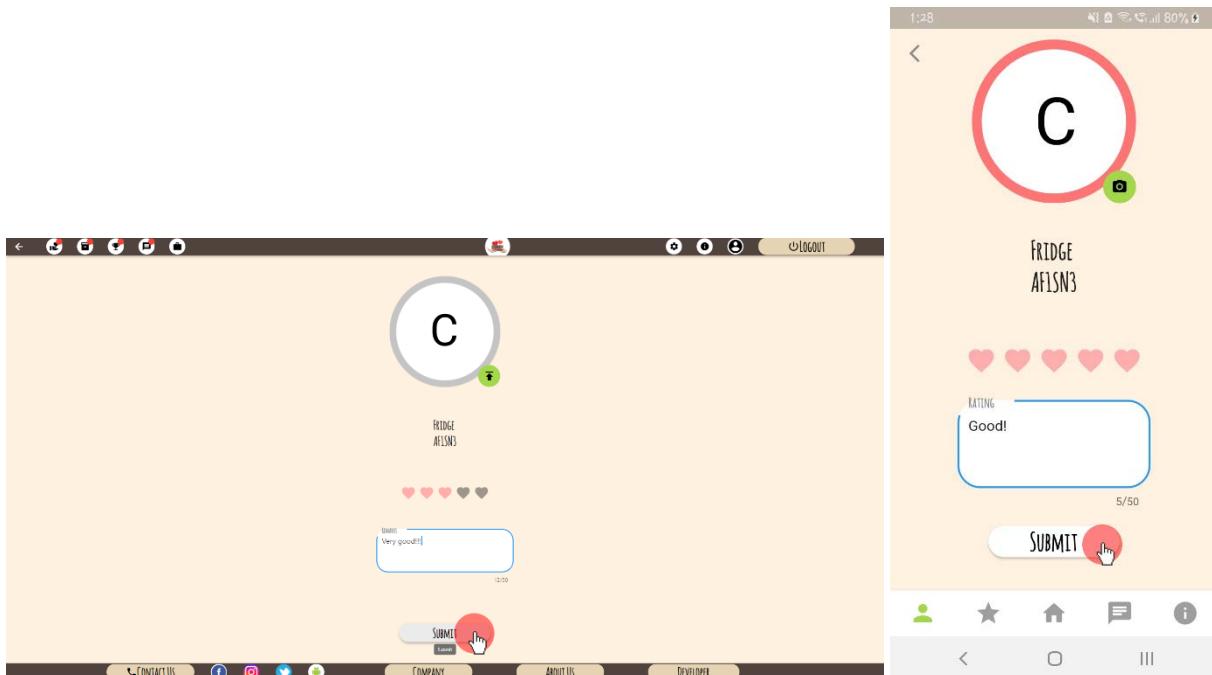
Back to the main screen, click on the exchange status representative button

TRIM+ COMPUTER' with ID 'L1EYSA'. The status bar above the table shows 'REQUEST 0', 'EXCHANGING 1', and 'EXCHANGED 0'. The right screenshot shows a detailed view of the selected item. At the top, it says 'STATUS' with three tabs: 'REQUEST' (0), 'EXCHANGING' (1, highlighted in blue), and 'EXCHANGED' (0). Below that, it shows '1 STATUS-EXCHANGING' for '24" PRISM+ COMPUTER' with 'FRIDGE' as the exchange item. The bottom of the screen has 'ACTIVE' and 'ACHIEVE' buttons, along with the same navigation icons as the main screen."/>

Click on the exchange now representative button



Scan or key in QR Code and click on submit to the exchange



Fill in feedback for other exchange user and click on submit button

Database

Overview of the database details.

Firebase's Firestore

The screenshot shows the Firebase Firestore interface. On the left, there is a sidebar with a tree view of collections: app_feedback, chat, chat_list, collector_feedback, donate_item, exchange_feedback, exchange_item, exchange_qa, needs_item, pin_points, role, user_collector, user_company, and user_normal. The main area shows a document under the app_feedback collection with the ID 1617478354282000. This document contains fields: image_path (a URL to a Google Cloud Storage image), published_date (April 4, 2021 at 3:32:34 AM UTC+8), rating (5), remark ("Good Application, highly recommended"), and status (1).

Fig 106, the result of the Firebase's Firestore database

Firebase's Firestorage

The screenshot shows the Firebase Storage interface. It displays a list of three folders: usercollector/, usercompany/, and usernormal/. Each folder is represented by a folder icon and a name.

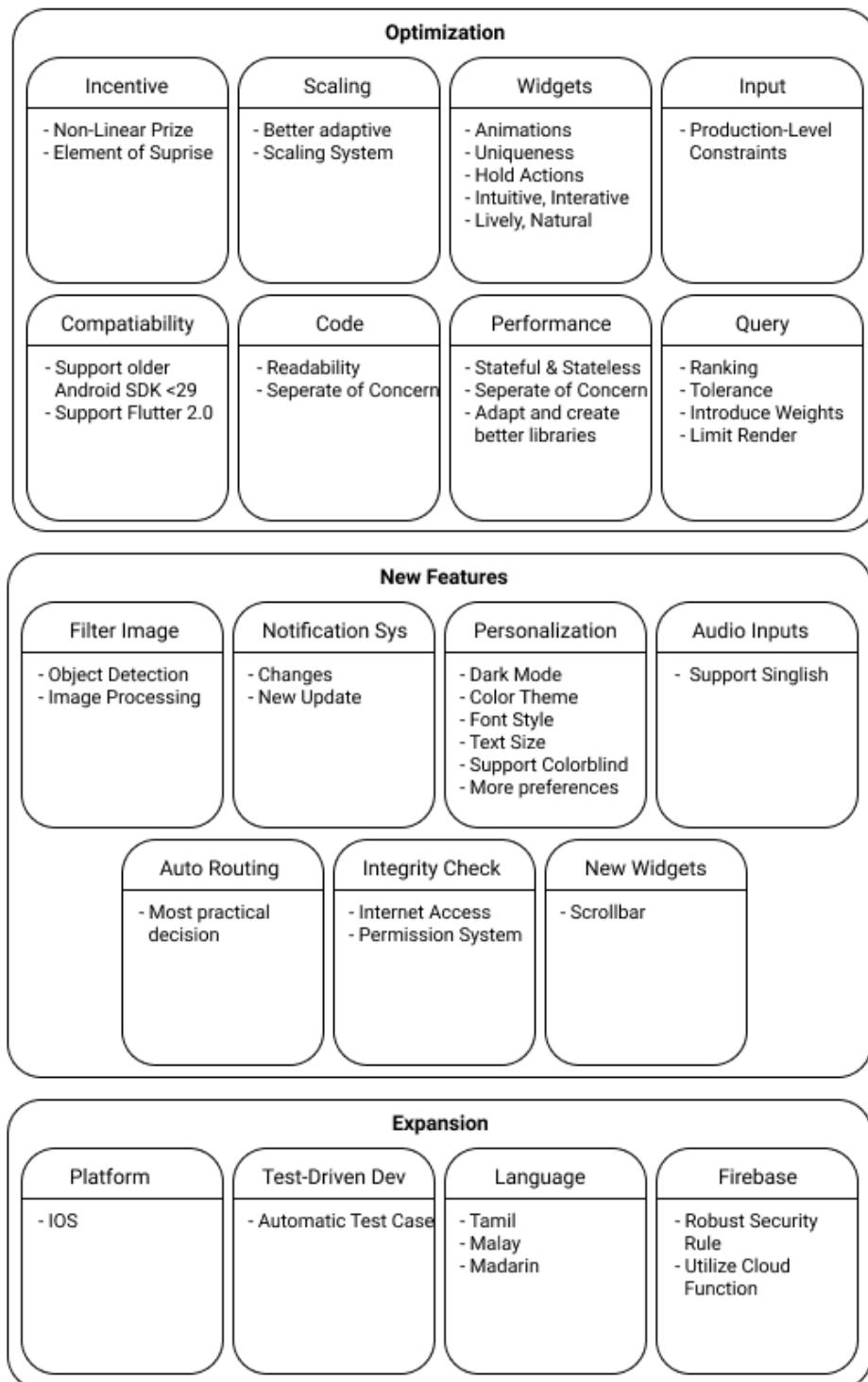
Fig 107, the result of Firebase's Firestorage database

The screenshot shows the Firebase Storage interface. It displays a list of seven images with their details: Name, Size, Type, and Last modified. The images are: 1618298016852000 (49 KB, image/png, Apr 13, 2021), 1618299209797000 (49 KB, image/png, Apr 13, 2021), 1618299558245000 (609.09 KB, image/png, Apr 13, 2021), 1618299716286000 (49 KB, image/png, Apr 13, 2021), 1618309253772000 (246.1 KB, image/png, Apr 13, 2021), 1618300536745000 (49 KB, image/png, Apr 13, 2021), and 1618300762826000 (49 KB, image/png, Apr 13, 2021). A detailed view of the image 1618414194898325 is shown on the right, showing its name (1618414194898325), size (1,481 bytes), type (image/jpeg), creation date (Apr 14, 2021, 11:29:54 PM), and update date (Apr 14, 2021, 11:29:54 PM). The image itself is a small white cat face.

Fig 108, the result of Firebase's Firestorage donated image database

What's Next?

There are 3 classify ways to improve the “JustGivit” application as shown below.



Challenges

There are countless challenges throughout this final year project, everyone's difficulty is unique to one another. Below are the 3 main challenges for me.

Learning Curve

I started with little to no experience in this project. I required a lot of patience and discipline to conduct self-test and self-research on foreign topics especially for BLOC mechanism, Firebase services, DART language, setting up Flutter on android properly. This process can be frustrating at times as it is time-consuming. Persistence, proper planning, step-by-step, and divide and conquer approach make this complex problem achievable. On top of that, dealing with a flutter's web on its beta channel, the sheer amount of changes in the framework and bugs is disruptive to the progression of the development.

Maintaining Scope

The initial proposal and ideas were largely ambitious without considering the duration of this project and may perceive as an unrealistic goal. At that point in time, my goal was to achieve as much as possible, as a way to challenge myself. In return, this has pushed me beyond my capable limit and able to learn many different development processes in such a short period of time. With this, I am more comfortable trying out complex mechanisms as a full-stack developer such as building a better and clean architecture, ability to think of the object-instance at an abstract level.

Coping

The whole duration of this project was about 9 months only. On top of that, there are school modules to handle concurrently especially heavy modules such as MDP, projects from TE modules. This has created a big obstacle for FYP given the scope I am trying to achieve since I also want to do my best for other modules too. Maintain a positive mentality and moral support from friends, family and supervisors are essential to survive well in this project.

Conclusion

All in all, this project has established a solid foundation for the future entrepreneur(s) or student(s) who would like to carry on this existing project. This project has explored and achieved the basic echo system with different possible roles of users, in making sure a functional donation and exchanging process for the end-users. Besides that, this project has room for improvement in many different ways and potentially moving its development to production-level. I hope that this product will soon embark on its journey, create an impact, using technologies to make a difference in our society for the better.

Appendix

Disclaimer

This project at the current state materials includes images and icons is intended use for education, open-source, and no copyright infringement is intended. These images and icons include all images used in this JustGivit project **except the developer's assets which is developer representative self-image and logo image representative of this product.**

Credits

I would like special thanks to Flutter's community, creating such wonderful libraries to speed up the development process. Below is the list of the dependent libraries used for this project and its purpose. These codes are found in pubspec.yaml file.

```
version: 1.0.0+1

environment:
  sdk: ">=2.10.4 <3.0.0"

dependencies:
  flutter_spinning_wheel: ^1.1.0          # Prize Award Animation
  flutter_rating_bar: ^3.2.0+1            # Rating Bar Widgets
  fl_chart: ^0.12.2                      # Allow easy use of pie chart
  flutter_spinkit: ^4.1.2+1              # Loading Indicator Animation
  csv: ^4.1.0                           # Allow easy read and write on CSV files
  http: ^0.12.2                         # Allow accessibility to HTTP
  google_maps_flutter: ^1.0.6             # Allow google map feature on (android) platform
  google_maps_flutter_web: ^0.1.0+8       # Allow google map feature on (web) platform
  location: ^3.0.0                       # Allow easy access to location services on (android) platform
  location_web: ^1.0.1                   # Allow easy access to location services on (web) platform
  url_launcher: ^5.7.10                  # Allow accessing to a web page given URL
  qr_code_scanner: ^0.2.1                # Ease of use for QR Scanner
  qr_flutter: ^3.1.0                     # QR Code generator given a string of values
  image_picker: ^0.6.7+17                # Select image from native mobile gallery
  image_picker_for_web: ^0.1.0+2         # Select image from desktop directory (web)
  flutter_session: ^0.1.1                # Allow data persistence feature via session
  firebase_core: ^0.5.3                  # Support firebase services
  firebase_auth: ^0.18.4+1               # Support firebase authentication
  firebase_storage: ^5.2.0                # Support firebase storage
  cloud_firestore: ^0.14.4               # Support firebase firestore
  flutter_bloc: ^6.1.1                  # Support BLOC mechanism
  equatable: ^1.2.5                     # This is use to compare if 2 objects belong to the same instances
  meta: ^1.2.4                          # Define annotation such as @required to allow better readability in code
  email_validator: ^1.0.6               # A general email validator
```

Access guide

Web

1. Enter this **URL** in your web browser: [justgivit-7ec8a.firebaseio.com.](https://justgivit-7ec8a.firebaseio.com/)

Android

1. Enter this **URL** in your mobile browser: [justgivit-7ec8a.firebaseio.com.](https://justgivit-7ec8a.firebaseio.com/)
2. If you have not login.
 1. **Scroll down** until you see Android Icon **at the footer** as shown below.

 2. Click the **Android icon** should redirect to the download APK screen.
 3. Click on the '**Download JustGivit**' Application as shown below, this should redirect to a SharePoint in Microsoft OneDrive's folder.


4. Click on the '**Download**' button, the application should prompt permission to install the application.


3. If you have login into the system.
 1. At the **footer of the screen**, click the Android icon should redirect to the download APK screen.
 2. Click the **Android icon** should redirect to the download APK screen.



3. Click on the '**Download JustGivit**' Application as shown below, this should redirect to a SharePoint in Microsoft OneDrive's folder.

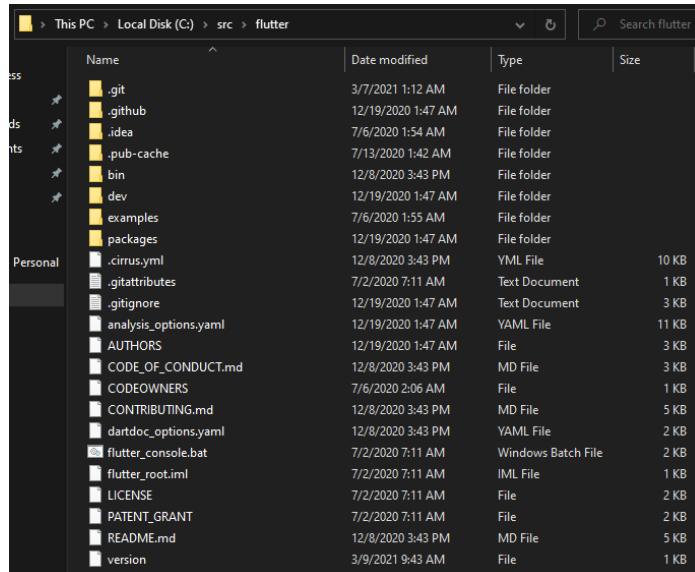

4. Click on the '**Download**' button, the application should prompt permission to install the application.


Source code guide

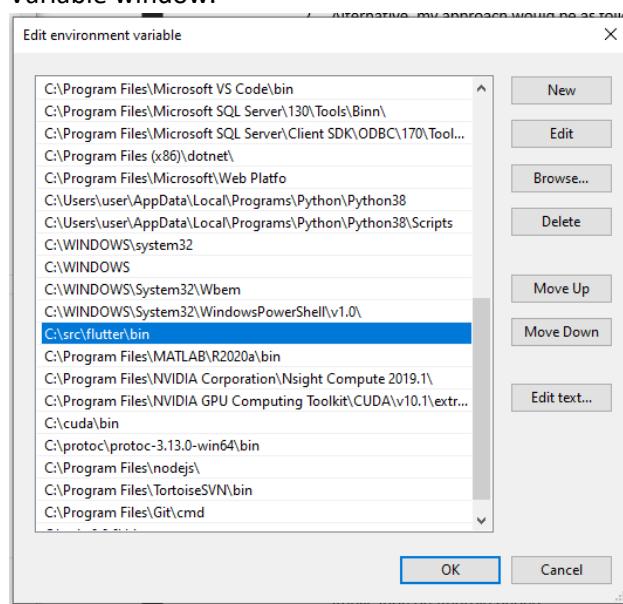
Always important to search essential for more guidelines such as Google, Youtube, StackOverFlow, they are your best survival tools! All the best!

Setup Flutter

1. Follow closely the given guideline provided by the Flutter team: <https://flutter.dev/docs/get-started/install>, the instruction easy to follow.
2. Alternatively, my approach would be as follow (Window friendly only) :
 1. Download the Flutter SDK package from the link above
 2. Place in this recommended directory as shown



3. Make sure ENV PATH is set up, such that the terminal can recognise the Flutter SDK package as shown below, this can be accessed via Edit Variables -> System Variables Section -> Click on 'Path' variable and press 'Edit' at the lower right of the Environment Variable window.



- Once done, to test if it is working open command prompt (CMD) type 'flutter doctor' If there is instruction just follow accordingly.

```

C:\Users\user>flutter -doctor
A new version of Flutter is available!
To update to the latest version, run "flutter upgrade".

Manage your Flutter app development.

Common commands:
  flutter create <output directory>
    Create a new Flutter project in the specified directory.

  flutter run [options]
    Run your Flutter application on an attached device or in an emulator.

Usage: flutter <command> [<arguments>]

Global options:
  -h, --help           Print this usage information.

```

This output requirement below should be sufficient to run this project

```

C:\Users\user>flutter doctor
Doctor summary (to see all details, run flutter doctor -v):
[✓] Flutter (Channel beta, 1.25.0-8.1.pre, on Microsoft Windows [Version
    10.0.18363.1379], locale en-US)
[!] Android toolchain - develop for Android devices (Android SDK version 30.0.0)
    X Android license status unknown.
        Run 'flutter doctor --android-licenses' to accept the SDK licenses.
        See https://flutter.dev/docs/get-started/install/windows#android-setup for
        more details.
[✓] Chrome - develop for the web
[✓] Android Studio (version 4.1.0)
[✓] VS Code, 64-bit edition (version 1.52.1)
[✓] Connected device (2 available)

! Doctor found issues in 1 category.

C:\Users\user>

```

Setup Android Simulator

- Install Android Studio software.
- Create a Virtual Device: <https://developer.android.com/studio/run/managing-avds#createavd>.
- May need to enable Virtual Technologies in BIOS of the computer: <https://stackoverflow.com/questions/39542853/enable-vt-x-in-your-bios-security-settings-refer-to-documentation-for-your-comp>.
- To check if it is working try run a.

Setup VSCode

- Create a directory of your naming folder.
- Download the JustGivit's source code.

3. Unzip and paste them into the directory you created in Step 1.
4. Install and open VSCode software.
5. Install Flutter and Firebase supported Extension on VSCode's Extension Workshop as shown below:



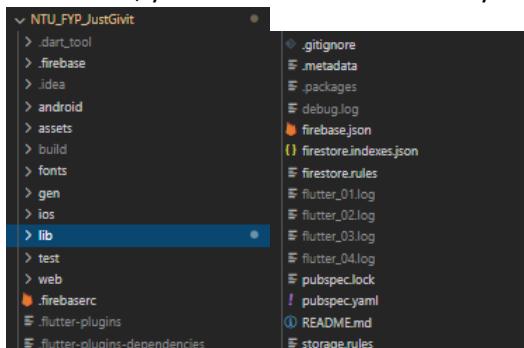
QOL Extension:



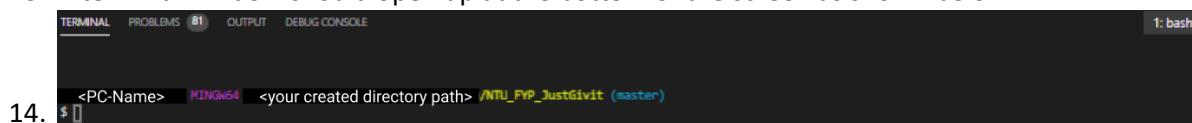
6. At the top left corner of the VSCode window select 'File' -> 'Add Folder as workspace'.
7. A directory selection window should pop-up.
8. Navigate to the directory folder u created in Step 1.
9. Once done, press this icon below to check if the repository has been installed.



10. Once done, you should see this directory in the explorer section as shown below.



11. Select and right-click 'NTU_FYP_JustGivit' at explorer.
12. Select 'Open in Integrated Terminal'.
13. A terminal window should open up at the bottom of the screen as shown below.



14. (Note: Can have multiple terminals by repeat step 11 again, useful if you want to run multiple platforms concurrently)
15. (Optional) Setup version control software such as Github, SVN, Git, etc. Git and Github Instructions: <https://www.jcchouinard.com/install-git-in-vscode>
16. This point will require to set up properly on the Android platform (android folder) and Web platform (web folder). This requires some extensive trial and error for beginners and expect unresolved issues and Flutter and Android compatibility issues. At the point of writing this, Flutter has recently release ver 2.0 which moving web support to a stable version from the beta version. I believe should be easier now. Key things to consider below:
 1. Android: SDK Support, Dependencies, Gradle, Native Permission, Logo.
 2. Web: Javascript, HTML, CSS Support settings (if needed), Logo.

Setup Firebase:

1. Create a Firebase account
2. Follow the instruction will do:
<https://firebase.google.com/docs/flutter/setup>
- Other useful links:
<https://flutter.dev/docs/development/data-and-backend/firebase>
3. Make sure it is set up on both side:
 1. Firebase's Console Setup respective services.
 2. VSCode Setup Firebase to run support flutter with firebase and allow your firebase to recognise your Flutter's workspace.
4. May require a bit of trial and error for it to work well as sometimes they may have bugs in the latest version on Firebase or Flutter.

Execute Web

1. Flutter Debug
 1. Ensure your terminal current directory is pointing the same as 'Setup VSCode's Step 14'.
 2. At Integrated terminal type 'flutter run -d chrome'. This will fire up a chrome web browser with localhost and an assigned port number. (Note: The performance is expected to be worst compare to Firebase Local Hosting as it is running on debug mode, press 'r' at the terminal for 'Hot Reload', 'R' at the terminal for 'Hard Refresh', 'q' to exit)
2. Firebase Local Hosting
 1. Ensure your firebase is set up correctly in 'Setup Firebase'.
 2. Ensure your terminal current directory is pointing the same as 'Setup VSCode's Step 14'.
 3. At Integrated terminal type 'firebase serve –only hosting'. This will fire up a chrome web browser with localhost and an assigned port number. (Note: Make sure you re-run compiled file to update the changes correctly, press CTRL+C to exit)
3. Firebase Deploy Hosting
 1. Ensure your firebase is set up correctly in 'Setup Firebase'.
 2. Ensure your terminal current directory is pointing the same as 'Setup VSCode's Step 14'.
 3. At Integrated terminal type 'firebase deploy –only hosting'. This deploys and published to your firebase hosting and access the given URL or your custom domain name to access. (Note: It takes a while for firebase hosting to update the changes. Make sure you re-run compiled file to update the changes correctly. Best to test if there is an update via access in incognito mode as it does not hold any browser cache data)

Execute Android

1. Simulator
 1. Ensure Android Simulator is set up properly.
 2. Ensure the bottom right of the VSCode window is the mobile name of the simulator by select and choose the options. This should fire up the simulator application.



3. Ensure your terminal current directory is pointing the same as 'Setup VSCode's Step 14'.
4. At Integrated terminal type 'flutter run'. (Note: It should take some time to install dependent on your computer processor speed)
2. Android Phone
 1. Ensure Android Simulator is set up properly.
 2. Ensure your supported android device is plug into the computer via USB wire
 3. Ensure the bottom right of the VSCode window is the mobile name of your device by select and choose the options.



4. Ensure your terminal current directory is pointing the same as 'Setup VSCode's Step 14'.
5. At Integrated terminal type 'flutter run'. (Note: It should take some time to install dependent on your mobile processor speed)

Common Integrated Terminal Commands

Re-build respective platform compiled file

1. flutter clean // clear compiled files in the build directory
2. flutter build web // build compiled code for web
3. flutter build appbundle // build compiled code for android (App Bundle format)
4. flutter build apk --split-per-abi // build compiled code for android (APK format)

Test guide

Dummy Data Test

1. Enter this **URL** in your web browser: justgivit-7ec8a.firebaseio.com or Access via “JustGivit” Application on Android phone.
2. At the footer of the screen, click on the ‘**Developer**’ button as shown below, this would redirect to the developer screen.



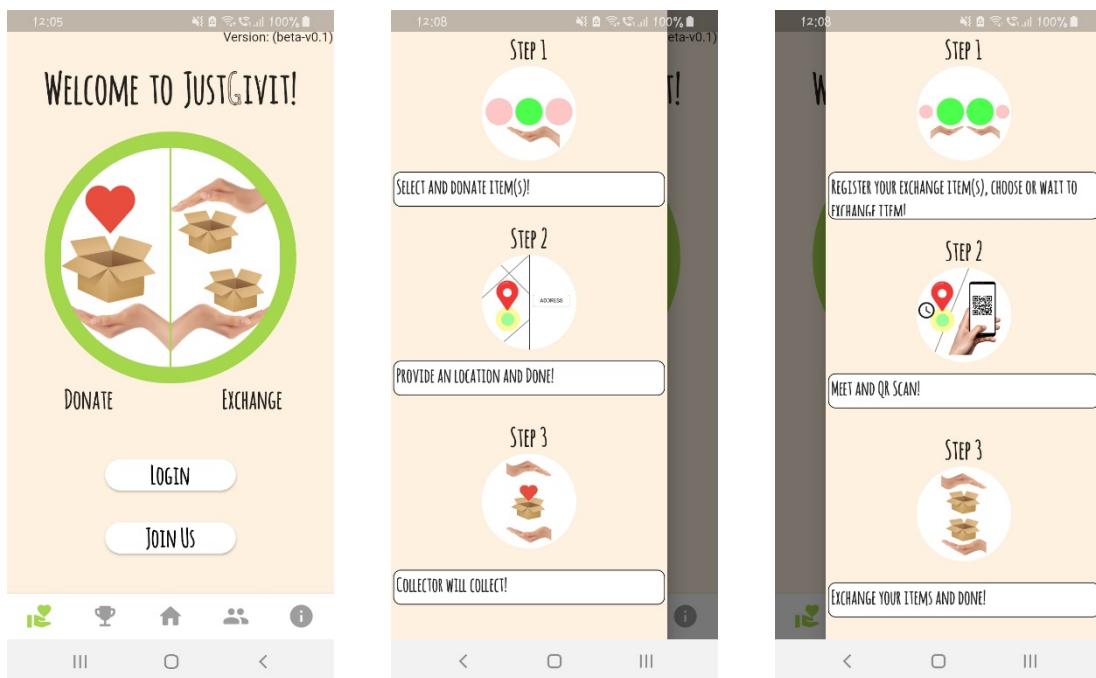
3. Click on the hidden ‘**Sample Data**’ icon in the web platform as shown below, this will repopulate or create data at the Firebase database.



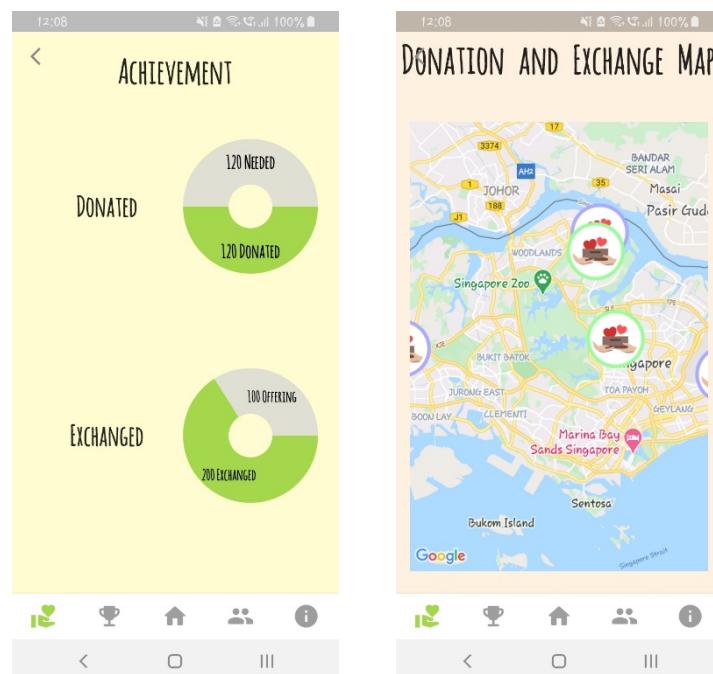
4. The terminal console will print the output if lib/data/conf/dev.dart, boolean variable ‘DEBUG’ is value ‘true’.

Mobile UI

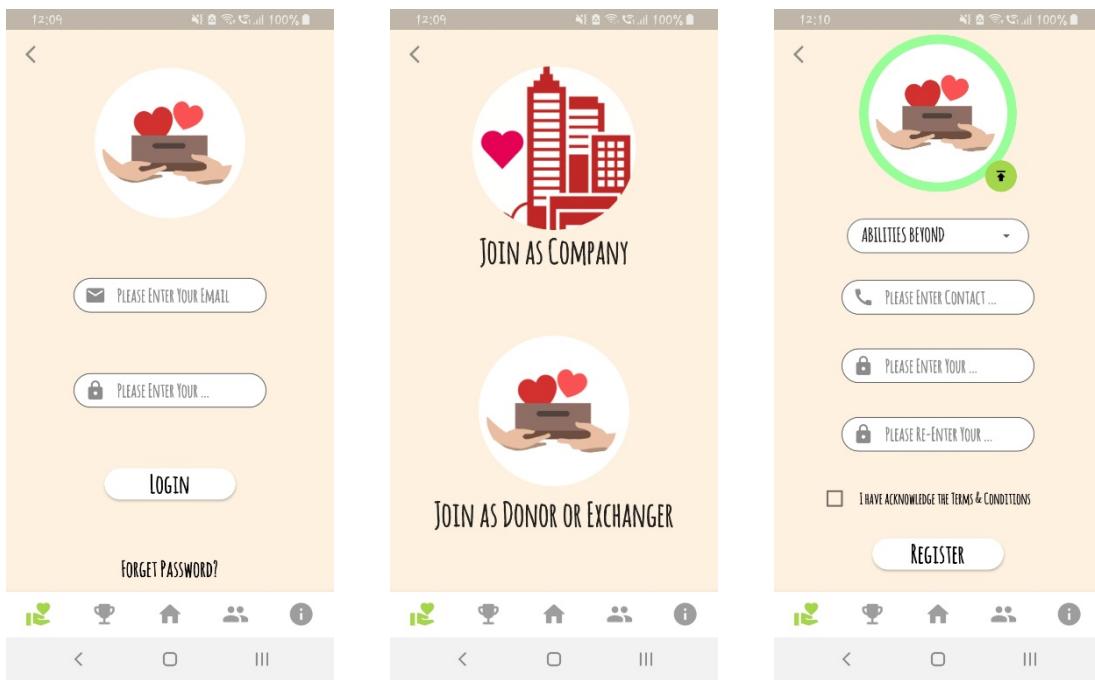
Public User



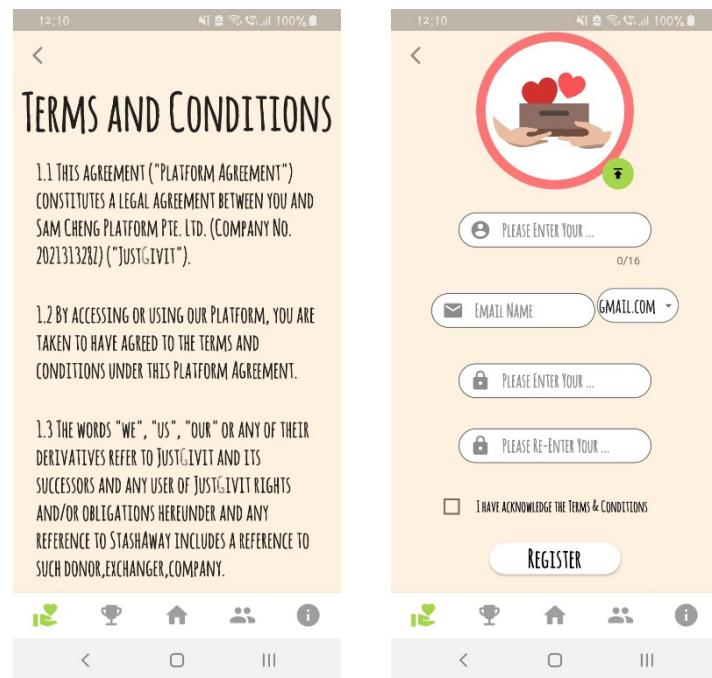
Public user main screen (refer to Fig 41) (Left), Public user main screen with details (refer to Fig 42) (Middle, Right)



Public main Sub-screens with details (refer to Fig 43)



Logic screen (refer to Fig 45) (left), Join screen (refer to Fig 46) (Middle), Register company screen (refer to Fig 47) (Right)



Register normal screen (refer to Fig 48)

OUR GOAL

ALL AROUND THE WORLD, EXTENSIVE WASTE IS A HUGE PROBLEM. IN SINGAPORE, WE HAVE GRADUALLY INCREASE IN TOTAL WASTE OVER THE LAST DECADE. UNFORTUNATELY, SINGAPORE'S LANDFILL HAS LIMITED SPACE WHICH LASTS UP TO THE YEAR OF 2056 (36 YEARS). "JUSTGIVIT", IT IS A DIGITAL PLATFORM FOR GIVING, WHERE USERS CAN EXCHANGE OR DONATE UNWANTED REUSABLE WASTE TO THOSE WHO NEED IT INSTEAD

WHO ARE WE?

DR TAN SHEN IS A SERIAL ENTREPRENEUR, INVESTOR, FOUNDER OF JUSTGIVIT. DR SAM CHEONG AND HIS TEAMS HAVE BEEN WORKING HARD WITH VARIOUS PRODUCTS. IN 2017, HE HAS MADE HEADLINE ON HIS PREVIOUS PRODUCT SUCH AS NEEDSG+. HE HAS DRIVEN THE MOVEMENT OF LOCAL TALENT SUCH AS SG INNOVATE 2018. OUR VISION IS TO INSPIRE AND IMPACT HUMANITY FORWARD THROUGH

DEVELOPER

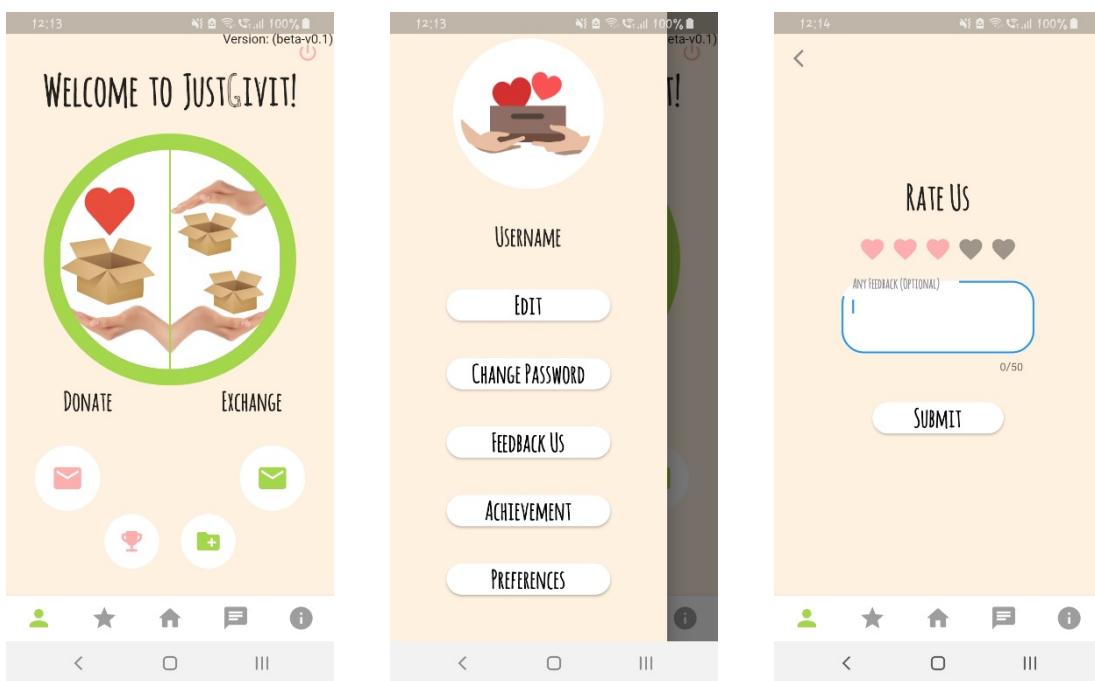
| Name | Role | Contact Number | Email Address |
|---------------------|--------------------------|----------------|---------------------------|
| Dr Tan Shen | Founder | +65 9555 7676 | FOUNDER@JGV.COM |
| Ms Sheryl Cheng | Co-Founder | +65 9555 6155 | COFOUNDER@JGV.COM |
| Mr Obama On Freeman | Operations Officer | +65 8555 1250 | OPERATIONSOFFICER@JGV.COM |
| Ms Zuri Hayden | Public Relations Officer | +65 8555 0197 | PROFFICER@JGV.COM |
| Dr Letizia Chatra | IT Officer | +65 9555 0760 | ITOFFICER@JGV.COM |

CONTACT INFORMATION

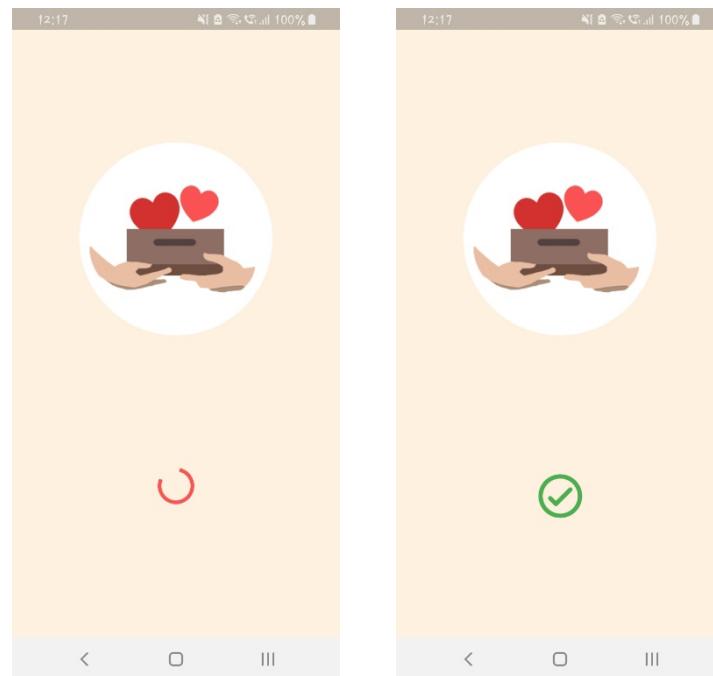
jsam002@e.ntu.edu.sg
<https://sg.linkedin.com/public-profile/in/sam>
<https://sam.github.io>

Informative related screens (refer to Fig 49)

Normal User

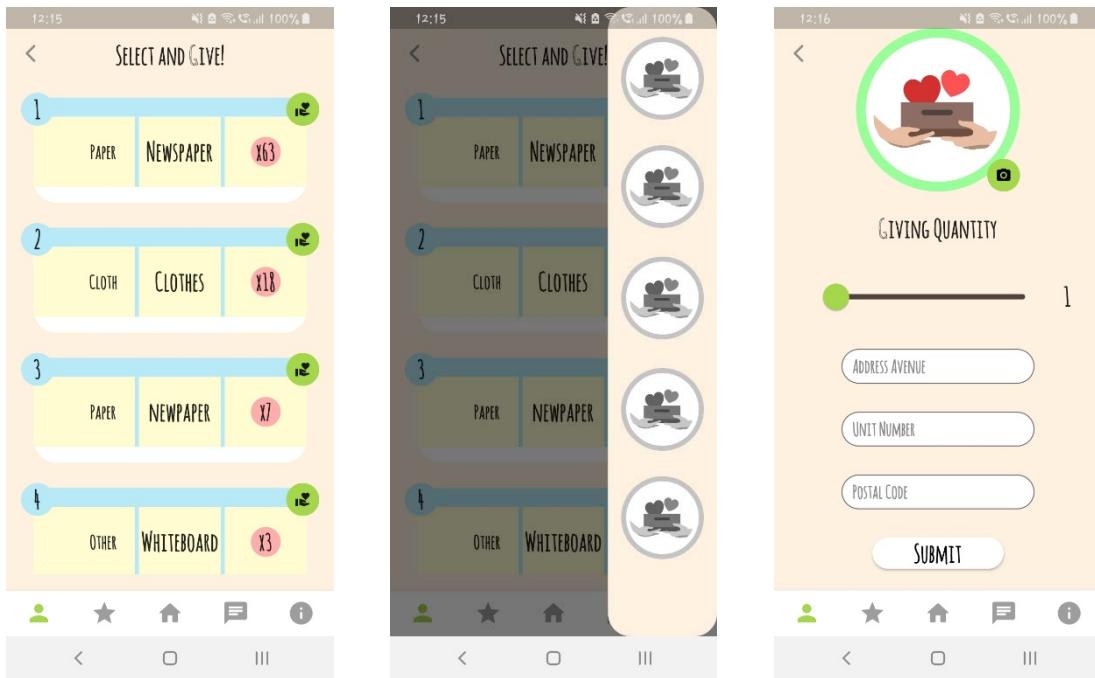


Normal main screen (Left), Profile-related screen (refer to Fig 50,51,53,59) (Middle), Application feedback screen (refer to Fig 53) (Right)

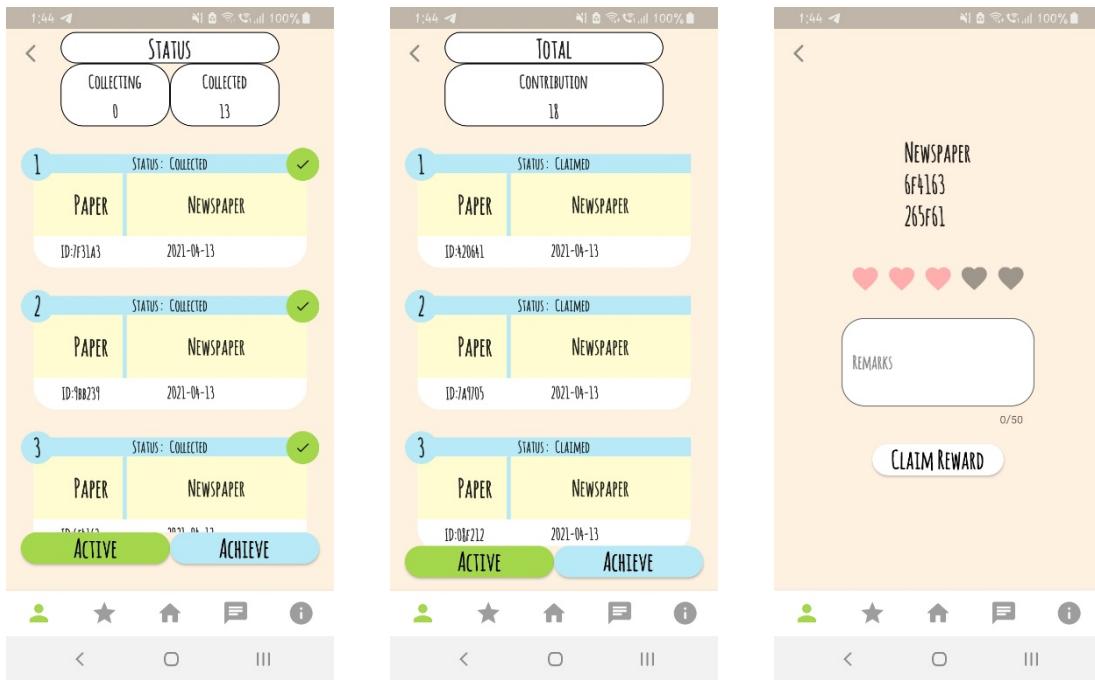


Transition screens

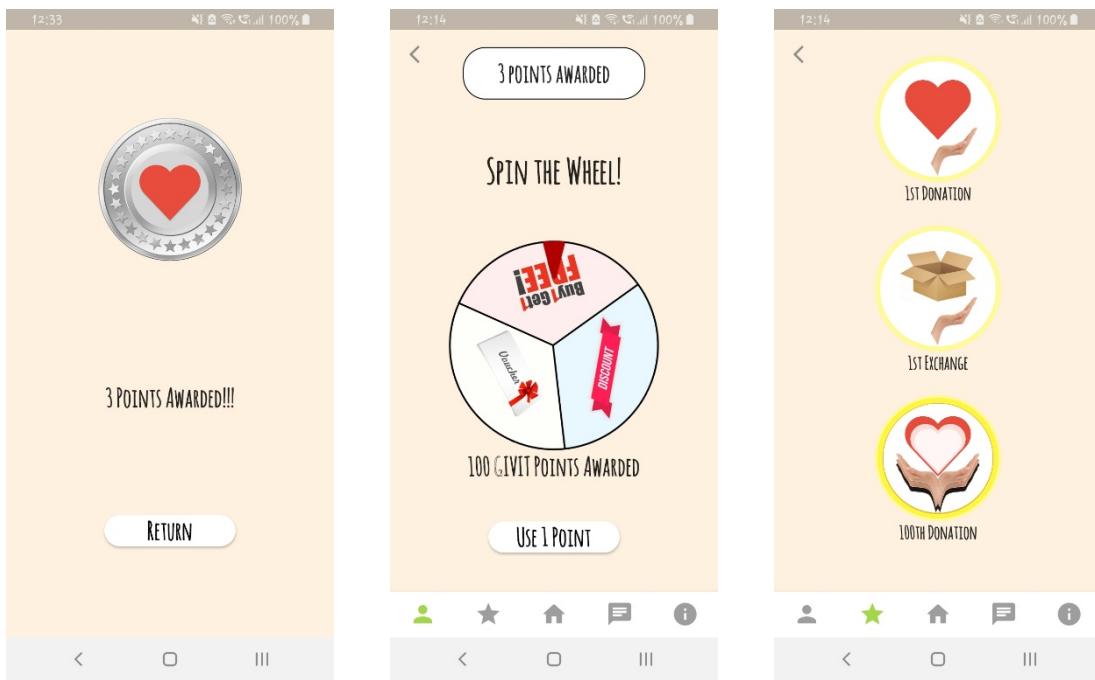
Donate



Donate market screen (refer to Fig 54) (Left, Middle), Give screen (refer to Fig 55) (Right)

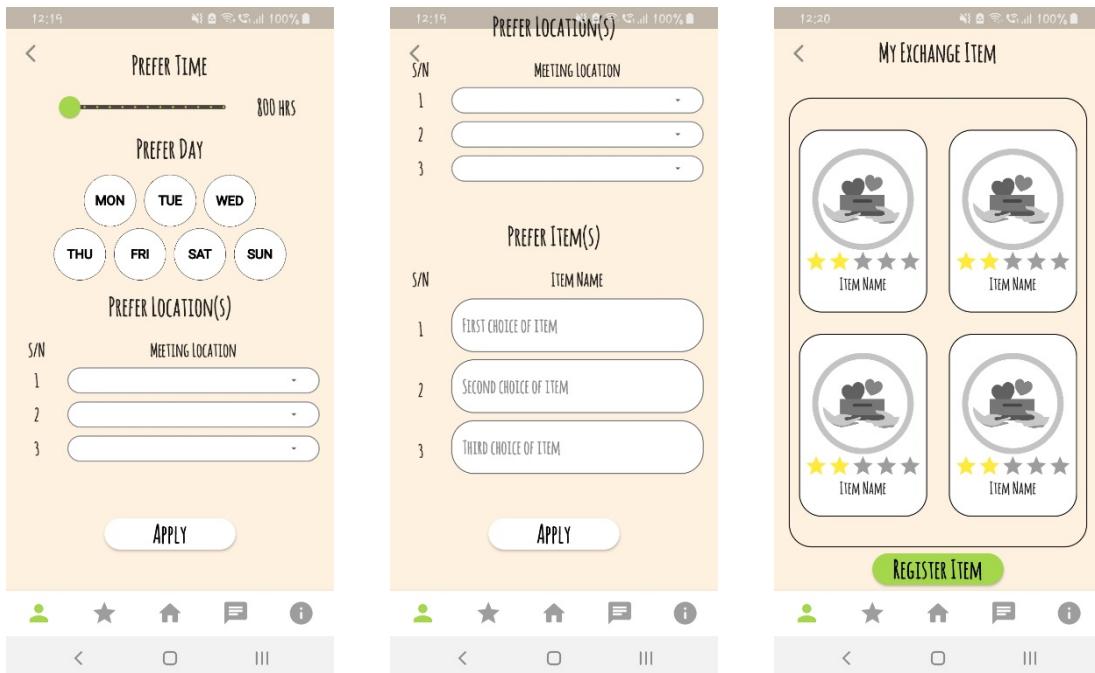


Donate status screen (refer to Fig 56) (Left, Middle), Donate feedback screen (refer to Fig 57) (Right)

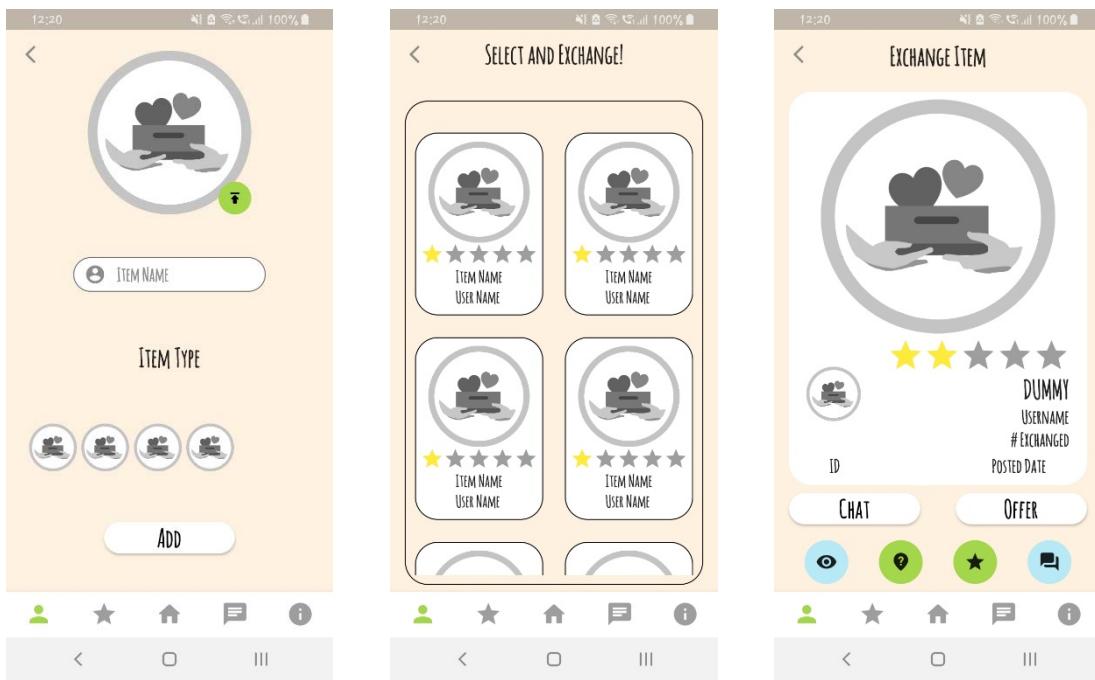


Claimed reward screen (refer to Fig 58) (Left), Reward screen (refer to Fig 52) (Right), Achievement screen (refer to Fig 51) (Right)

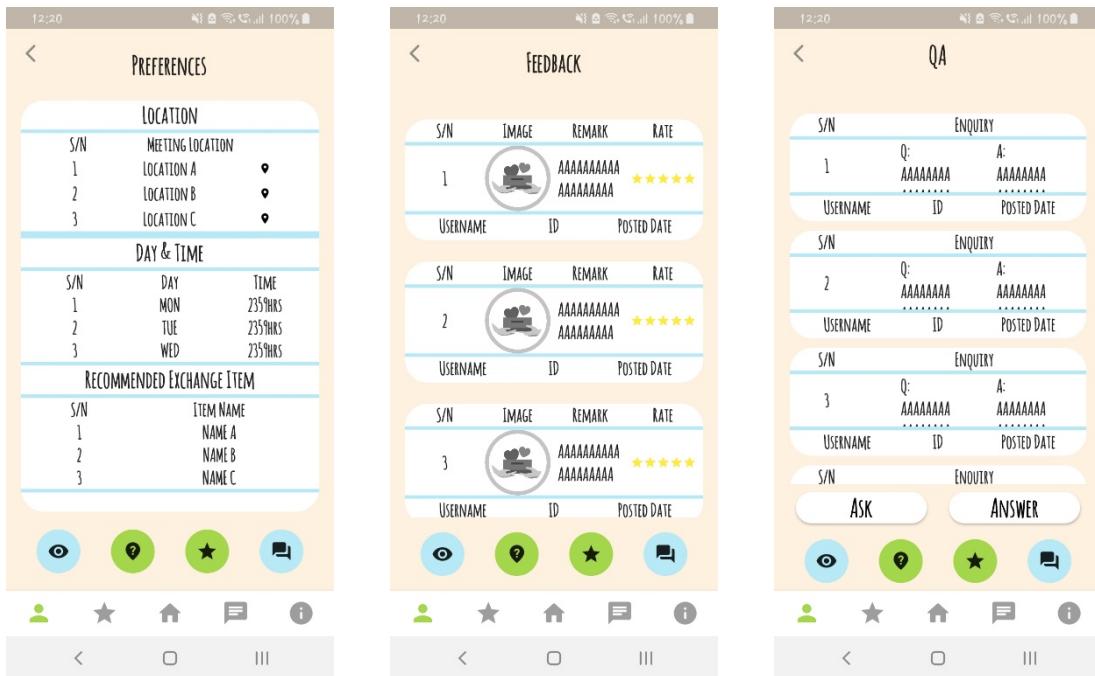
Exchange



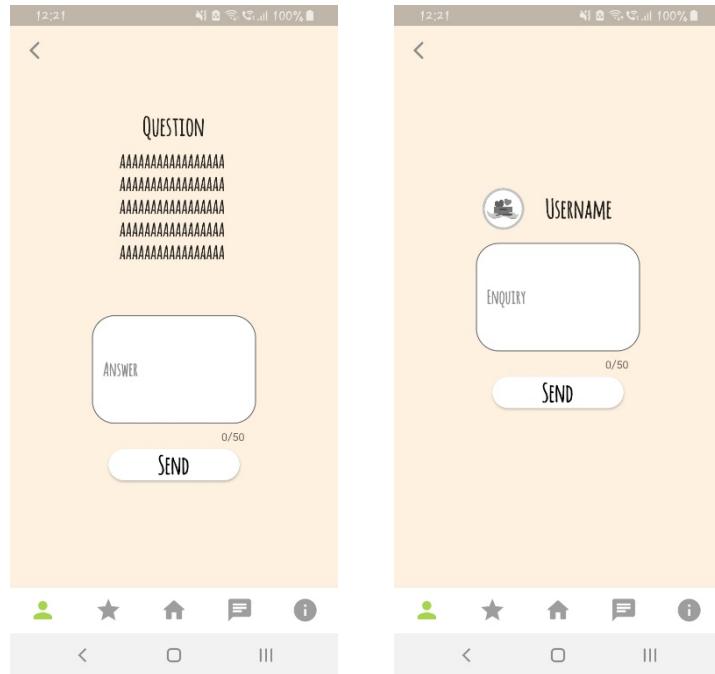
Exchange preference screen (refer to Fig 59) (Left, Middle), Manage exchange screen (refer to Fig 60) (Right)



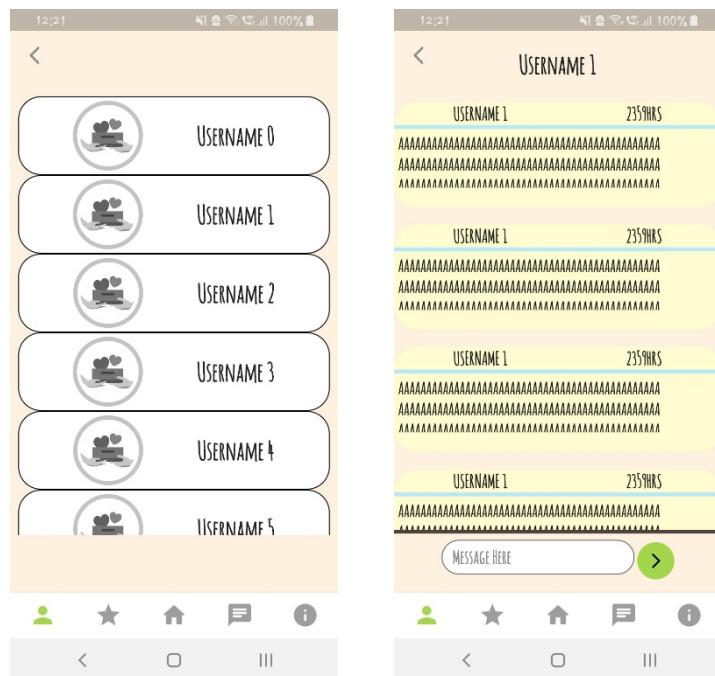
Register an exchange item screen (refer to Fig 61) (Left), Exchange market screen (refer to Fig 62) (Middle), Exchange item detail screen (refer to Fig 63) (Right) (1st Part)



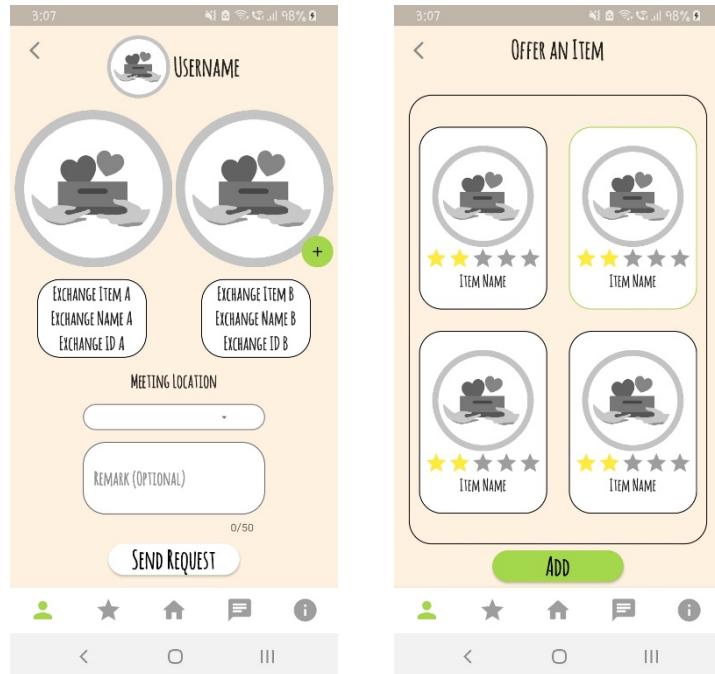
Exchange item detail screen (refer to Fig 63) (2nd Part)



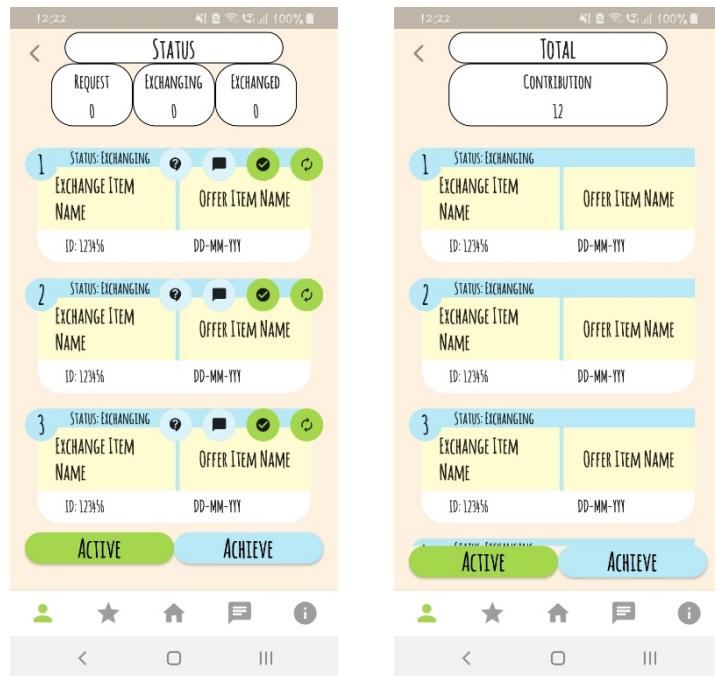
Enquiry and Answer screen (refer to Fig 64,65)



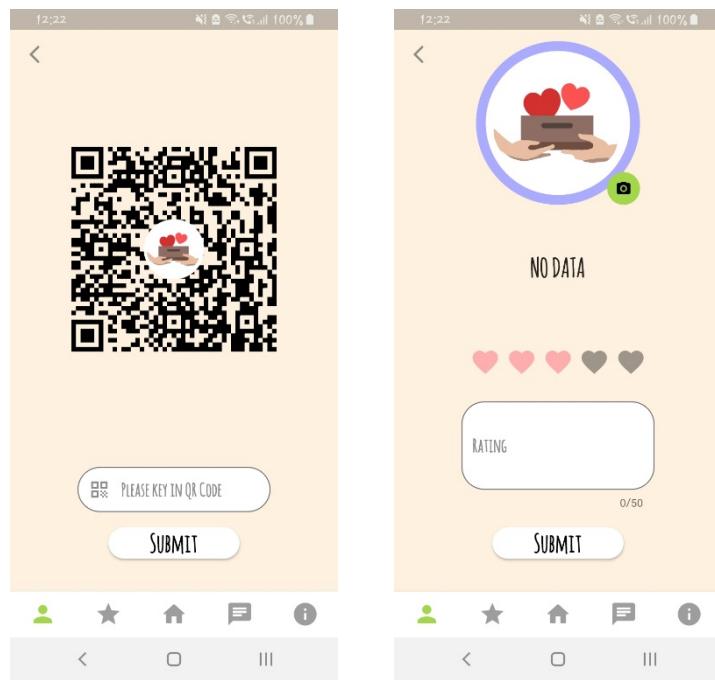
Chat screen (refer to Fig 66)



Offer exchange item screen (refer to Fig 67) (Left), Add exchange item to offer screen (refer to Fig 68) (Right)



Exchange status screen (refer to Fig 69)



QR Code screen (refer to Fig 71) (Left), Exchange feedback screen (refer to Fig 72) (Right)

Links

| S/N | Description | URL |
|------|------------------------------------------------------------------------------------------------------------------------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| {1} | A screenshot of the charity website created by the National Volunteer & Philanthropy Centre (NVPC) for financial-related donation means. | https://www.giving.sg/ |
| {2} | A commonly distributed leaflet from a charity organization image. | https://www.reddit.com/r/singapore/comments/b0mcfb/no_wonder_i_dont_hear_karang_guni_men_honking/ |
| {3} | A Singapore-based government charity website. | https://www.charities.gov.sg/Pages/Home.aspx |
| {4} | The photo was taken by Gaya Chandramohan of a security guard personnel. | https://www.channelnewsasia.com/news/singapore/private-security-officers-abused-wages-suss-use-12844238 |
| {5} | 'Trash Nothing' mobile application on the play store page. | https://play.google.com/store/apps/details?hl=en&id=com.trashnothing.app |
| {6} | 'Trash Nothing' web application main page. | https://trashnothing.com/beta/ |
| {7} | 'Trash Nothing' opens the APIs page. | https://redocly.github.io/redoc/?url=https%3A//trashnothing.com/api/v1.2/trashnothing-openapi.yaml#operation/search_groups |
| {8} | Starbucks reach out innovations page. | https://ideas.starbucks.com/ |
| {9} | Government charity website static registered charity companies/organization datasets. | https://www.charities.gov.sg/_layouts/MCYSCPSearch/MCYSCPSearchCriteriaPage.aspx |
| {10} | Firebase pricing plans. | https://firebase.google.com/pricing |
| {11} | Flutter web beta information. (outdated news) | https://flutter.dev/web |
| {12} | Google Map API services information. | https://developers.google.com/maps/faq |
| {13} | Flutter's 404 representative images. | https://medium.com/flutter/handling-404-page-not-found-error-in-flutter-731f5a9fba29 |
| {14} | LOFI design of JustGivit project. | https://www.figma.com/file/V6NXE5wsfpmek00pg5weUZ/FYP_Justgivit?node-id=17%3A2 |
| {15} | HIFI design of JustGivit project. | https://www.figma.com/file/V6NXE5wsfpmek00pg5weUZ/FYP_Justgivit?node-id=426%3A836 |
| {16} | "Matsuoka Shuzo retired Japanese tennis star harvests clams in freezing water reminds you to never give up" image | https://www.youtube.com/watch?v=KxGRhd_iWuE&ab_channel=Ryuujin131 |
| {17} | Shocked Tom Meme image. | https://imgflip.com/memegenerator/174088636/Shocked-Tom |
| {18} | Goldfish wallpaper image. | https://wallpapercave.com/goldfish-wallpapers |

References

- [1] “Solid Waste Management - Total Waste Generated-Data.gov.sg (2000-2016)”. [Online]. Available: <https://data.gov.sg/dataset/solid-waste-management-total-waste-generated>. [Accessed: 22-Aug-2020].
- “Waste Recycling Statistic (2016-2018) - NEA Government”. [Online]. Available: <https://www.nea.gov.sg/docs/default-source/default-document-library/waste-recycling-stats-2016-to-2018.pdf>. [Accessed: 22-Aug-2020].
- [2] “ENVIRONMENTAL KEY STATISTICS”. [Online]. Available: <https://www.mse.gov.sg/resources/key-environmental-statistics.pdf>. [Accessed: 22-Aug-2020].
- [3] K. Silpa, Y. Lisa C., P. Bhada-Tata, and F. V. Woerden. “ What a Waste 2.0: A Global Snapshot of Solid Waste Management to 2050. Urban Development”. 2018. [Online]. Available: <https://openknowledge.worldbank.org/handle/10986/30317>. [Accessed: 22-Aug-2020].
- [4] C. Ong, L. Fearnley, and S. B. Chia “The diversity of divestment in Singapore: Junk commodities, charity gifts, and recycling bins. Environment and Planning E: Nature and Space” 2020. [Online] Available: <https://journals.sagepub.com/doi/abs/10.1177/2514848620924564> [Accessed: 23-Mar-2021].
- [5] “Annual Crime Brief 2020 Infographic”. Available: <https://www.police.gov.sg/media-room/statistics>. [Accessed: 24-Aug-2020].
- [6] J. Clausen, B. Blättel-Mink, L. Erdmann, and C. Henseling. “Contribution of online trading of used goods to resource efficiency: An empirical study of eBay users.” 2010. [Online] Available: <https://www.mdpi.com/2071-1050/2/6/1810> [Accessed: 23-Mar-2021].
- [7] P. R. De Andrade, A. B. Albuquerque, O. F. Frota, R. V. Silveira, and F. A. da Silva, (2015). “Cross platform app: a comparative study.” 2015. [Online] Available: <https://arxiv.org/abs/1503.03511> [Accessed: 23-Mar-2021].
- [8] S. Amatya, and A. Kurti. “Cross-platform mobile development: challenges and opportunities.” 2013. [Online] Available: https://link.springer.com/chapter/10.1007/978-3-319-01466-1_21 [Accessed: 23-Mar-2021].
- [9] T. F. Bernardes, and M. Y. Miyake, "Cross-platform Mobile Development Approaches: A Systematic Review," 2016. Available: <https://ieeexplore.ieee.org/abstract/document/7483531> [Accessed: 23-Mar-2021].
- [10] D. C. Brabham. “Crowdsourcing as a model for problem-solving: An introduction and cases.” 2008. [Online] Available: <https://journals.sagepub.com/doi/abs/10.1177/1354856507084420> [Accessed: 23-Mar-2021].
- [11] Y. M. Wang, Y. S. Wang, and Y. Y. Wang, “Exploring the determinants of university students’ contribution intention on crowdsourcing platforms: a value maximization perspective, Interactive Learning Environments”, 2021. [Online] Available: <https://www.tandfonline.com/remotexs.ntu.edu.sg/doi/full/10.1080/10494820.2021.1890619>. [Accessed: 03-Mar-2021].

- [12] M. Farahidayah, and A. Hazleen, "State of Mobile Crowdsourcing Applications: A Review", 2015. [Online] Available:
https://www.researchgate.net/publication/281277007_State_of_Mobile_Crowdsourcing_Applications_A_Review [Accessed: 23-Mar-2021].
- [13] J. Phuttharak, and S. W. Loke, "A Review of Mobile Crowdsourcing Architectures and Challenges: Toward Crowd-Empowered Internet-of-Things," 2018. [Online] Available:
<https://ieeexplore.ieee.org/document/8566148> [Accessed: 23-Mar-2021].
- [14] K. Wazny, "Applications of crowdsourcing in health: an overview", 2018. [Online] Available:
<https://www.ncbi.nlm.nih.gov/pmc/articles/PMC5840433/> [Accessed: 23-Mar-2021].
- [15] S. Vermicelli, L. Cricelli, and M. Grimaldi, "How can crowdsourcing help tackle the COVID-19 pandemic? An explorative overview of innovative collaborative practices", 2020. [Online] Available:
<https://onlinelibrary.wiley.com/doi/full/10.1111/radm.12443>. [Accessed: 03-Mar-2021].
- [16] Y. Yang, M. Sherman, and J. Lindqvist, "Disaster mitigation by crowdsourcing hazard documentation," 2014. [Online] Available:
<https://ieeexplore.ieee.org.remotexts.ntu.edu.sg/document/6970266> [Accessed: 23-Mar-2021].
- [17] E. Çolak, and F. Sunar. "The importance of ground-truth and crowdsourcing data for the statistical and spatial analyses of the NASA FIRMS active fires in the Mediterranean Turkish forests. Remote Sensing Applications: Society and Environment," 2020. [Online] Available:
<https://www.sciencedirect.com/science/article/abs/pii/S2352938520300999> [Accessed: 23-Mar-2021].
- [18] M. Modaresnezhad, L. Lyer, P. Palvia, and V. Taras, "Information Technology (IT) enabled crowdsourcing: A conceptual framework", 2020 [Online] Available:
<https://www.sciencedirect.com/science/article/pii/S0306457319302882>. [Accessed: 03-Mar-2021].
- [19] K. B. Sumra, and W. Bing. "Crowdsourcing in local public administration: the importance of online platforms. International Journal of Public Administration in the Digital Age" 2016. [Online] Available:
<https://www.igi-global.com/article/crowdsourcing-in-local-public-administration/161613> [Accessed: 23-Mar-2021].
- [20] F. J. Garrigos-Simon, Y. Narangajavana, and J. L. Galdón-Salvador. "Crowdsourcing as a competitive advantage for new business models." 2014. [Online] Available:
https://link.springer.com/chapter/10.1007/978-1-4614-8184-3_3 [Accessed: 23-Mar-2021].
- [22] "Flutter's single-code base feature" [Online]. Available: <https://flutter.dev/>. [Accessed: 27-Aug-2020].
- [21] "About DART?" [Online]. Available: <https://dart.dev/overview> [Accessed: 27-Aug-2020].
- [23] "Flutter's hot reload feature" [Online]. Available: <https://flutter.dev/docs/development/tools/hot-reload> [Accessed: 27-Aug-2020].
- [24] "About Flutter's Web" (Recent change Beta to Release since Flutter 2.0) [Online]. Available: <https://flutter.dev/web> [Accessed: 03-Mar-2021].

- [25] “Flutter 2.0 added interactive scrollbars” [Online]. Available: <https://developers.googleblog.com/2021/03/announcing-flutter-2.html> [Accessed: 03-Mar-2021].
- [26] K. Lane (3 June 2012). "Rise of Mobile Backend as a Service (MBaaS) API Stacks" 2012. [Online]. Available: <https://apievangelist.com/2012/06/03/rise-of-mobile-backend-as-a-service-mbaas-api-stacks/> [Accessed: 29-Aug-2020].
- [27] S. Robort, and L. Mohamed. “Developing Cloud-Based Mobile Apps Using Google Firebase.” 2019. [Online] Available: <https://dl.acm.org/doi/abs/10.5555/3381555.3381558> [Accessed: 29-Aug-2020].
- [28] I. Ulukaya “A Cross-platform Firebase Sample App Featuring Best Practices” 2019. [Online]. Available: <https://firebase.googleblog.com/2017/09/a-cross-platform-firebase-sample-app.html> [Accessed: 29-Aug-2020].
- [29] L. Moroney “The Firebase Realtime Database. In The Definitive Guide to Firebase.” 2017. [Online]. Available: https://link.springer.com/chapter/10.1007/978-1-4842-2943-9_3 [Accessed: 29-Aug-2020].
- [30] “Google pricing plan.” [Online]. Available: <https://firebase.google.com/pricing> [Accessed: 29-Aug-2020].
- [31] “Firebase’s Firestore service limitation” [Online]. Available: <https://firebase.google.com/docs/firestore/quotas> [Accessed: 29-Aug-2020].
- [32] “Firebase’s API Libraries” (Required: Google account) [Online]. Available: <https://console.cloud.google.com/apis/> [Accessed: 29-Aug-2020].
- [33] “setStates() API functions details” [Online]. Available: [https://api.flutter.dev/flutter/widgets/State setState.html](https://api.flutter.dev/flutter/widgets/State	setState.html) [Accessed: 29-Aug-2020].
- [34] “About BLOC and its features” [Online]. Available: <https://bloclibrary.dev/#/> [Accessed: 29-Aug-2020].
- [35] “About Flutter’s Widgets” [Online]. Available: <https://flutter.dev/docs/development/ui/widgets-intro> [Accessed: 29-Aug-2020].
- [36] “Flutter’s Navigation and Routing Mechanism” [Online]. Available: <https://flutter.dev/docs/development/ui/navigation> [Accessed: 29-Aug-2020].
- [37] “BLOC Architrecture” [Online]. Available: <https://bloclibrary.dev/#/architecture> [Accessed: 29-Aug-2020].
- [38] “BLOC packages and it’s feature maintained by Felix Angelov” [Online]. Available: https://pub.dev/packages/flutter_bloc [Accessed: 29-Aug-2020].
- [39] “What is CRUD?” [Online]. Available: <https://www.codecademy.com/articles/what-is-crud> [Accessed: 29-Aug-2020].

- [40] G. Erich “Design patterns: elements of reusable object-oriented software” 1995. [Online]. Available: <https://archive.org/details/designpatternsel00gamm/page/127> [Accessed: 29-Aug-2020].
- [41] “Firebase’s product or services” [Online]. Available: <https://firebase.google.com/products-build> [Accessed: 29-Aug-2020].
- [42] “ACID Properties DBMS” [Online]. Available: <https://www.geeksforgeeks.org/acid-properties-in-dbms/>. [Accessed: 29-Aug-2020].
- [43] B. Shneiderman “Designing the user interface strategies for effective human-computer interaction” 1987. [Online] Available: <https://doi.org/10.1145/25065.950626> [Accessed: 29-Aug-2020].
- [44] “Figma’s software details”. [Online]. Available: <https://www.figma.com/> [Accessed: 29-Aug-2020].