



School of Information Technologies
Faculty of Engineering & IT

ASSIGNMENT/PROJECT COVERSHEET - GROUP ASSESSMENT

Unit of Study: COMP 5216

Assignment name: COMP 5216 Final

Tutorial time: _____ / **Tutor name:** _____ /

DECLARATION

We the undersigned declare that we have read and understood the [University of Sydney Academic Dishonesty and Plagiarism in Coursework Policy](#), an, and except where specifically acknowledged, the work contained in this assignment/project is our own work, and has not been copied from other sources or been previously submitted for award or assessment.

We understand that failure to comply with the *Academic Dishonesty and Plagiarism in Coursework Policy* can lead to severe penalties as outlined under Chapter 8 of the *University of Sydney By-Law 1999* (as amended). These penalties may be imposed in cases where any significant portion of my submitted work has been copied without proper acknowledgement from other sources, including published works, the internet, existing programs, the work of other students, or work previously submitted for other awards or assessments.

We realise that we may be asked to identify those portions of the work contributed by each of us and required to demonstrate our individual knowledge of the relevant material by answering oral questions or by undertaking supplementary work, either written or in the laboratory, in order to arrive at the final assessment mark.

| Project team members | | | | |
|----------------------|------------|--------------|----------------|---------------|
| Student name | Student ID | Participated | Agree to share | Signature |
| 1. Siqi Wu | 490536885 | Yes | Yes | Siqi Wu |
| 2. Yiqing Yang | 470131402 | Yes | Yes | Yiqing Yang |
| 3. Xiaowei Zhang | 500659788 | Yes | Yes | Xiaowei Zhang |
| 4. Xiaoyu Huang | 500217441 | Yes | Yes | Xiaoyu Huang |
| 5. Yuan Li | 480180681 | Yes | Yes | Yuan Li |
| 6. | | Yes / No | Yes / No | |
| 7. | | Yes / No | Yes / No | |
| 8. | | Yes / No | Yes / No | |
| 9. | | Yes / No | Yes / No | |
| 10. | | Yes / No | Yes / No | |

1. Introduction

- a. Currently, there are a large number of orphans waiting for help. However, the lack of a way for orphanages and donors to communicate makes the donation not convenient, and the relationship between them weak.
- b. The goal of our project is to build a bridge between orphanages and donors so that more orphans could get donations. To achieve, we developed a platform which integrates many functions for both donor and orphanage. For orphanages, they would be able to propose their requirements on the homepage and share the daily life of orphans in their forum to attract more people to donate. For donors, they could view a list of orphanages and their current requirements of donation. Besides, to build trust between donors and orphanages, the donor can track each donation and can track the source and usage. Besides, donors could see how their donation is used in the forum or chat with the orphanage.

2. Validation

a. Validation of GUI

As for the GUI testing for our application, our group firstly performs the manual tests on our 1st version because they allow human observation which is helpful for us to provide a user-friendly GUI.

GUI tests by the development team

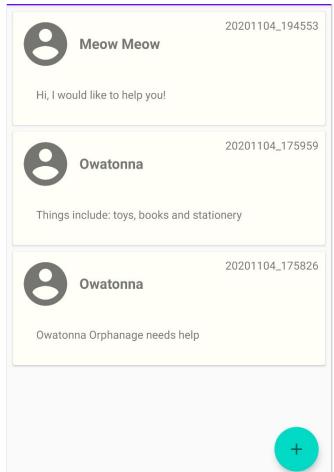
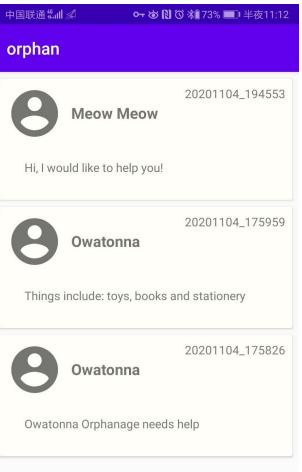
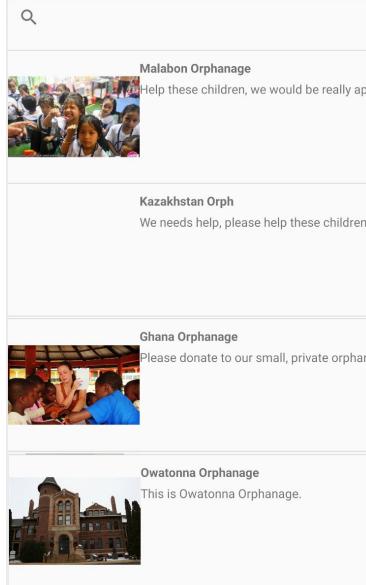
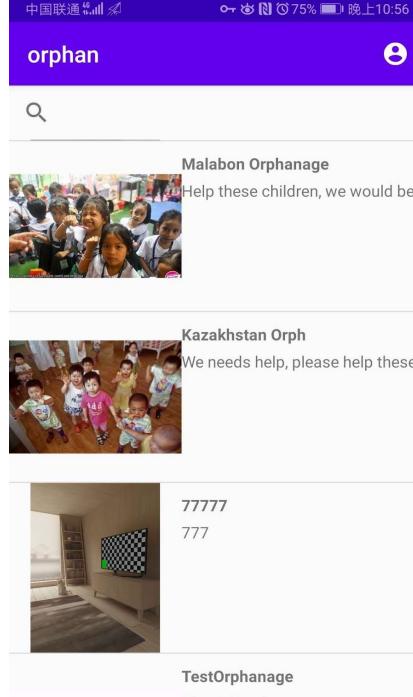
As the development team, we perform the GUI tests of the 1st version of our application on different devices, ranging from the emulators and the physical devices. We have made tests on the main parts of our application to check if the GUI components are placed correctly on different devices. Although due to the limit of the number of available Android devices our group has, we still perform the GUI tests on two different devices, including an emulator of Pixel XL API 30 and a physical device of Huawei Mate 10.

From the test results, the GUIs are slightly different among different devices. At the same time, there is still some information for financial GUI missing when testing on the physical device, which needs future improvement in the later version.

The following are the test results and comparisons of different tests.

| Emulators | Physical |
|---|---|
| Pixel XL API 30(resolution of 1440x2560:560dpi) | Huawei Mate 10(resolution of 2560x1550 pixels) |
| The financial GUI for the user and the orphanage: | The financial GUI for the user and the orphanage: |

| <p>Sports Happy</p> <p>Children are in need of sports equipment.</p> <table border="1"> <thead> <tr> <th>29</th> <th>\$5672</th> <th>\$5000</th> </tr> <tr> <th>Donors</th> <th>Current</th> <th>Goal</th> </tr> </thead> </table> <p>Usage</p> <p>We will use these money to buy sports equipment to children. Thank everyone!</p> | 29 | \$5672 | \$5000 | Donors | Current | Goal | <p>Dorm Mending</p> <p>The dorm for children is pretty old, and we lack the fare for mending. We are in need to provide them a better dorm.</p> <table border="1"> <thead> <tr> <th>0</th> <th>\$0</th> <th>\$30000</th> </tr> <tr> <th>Donors</th> <th>Current</th> <th>Goal</th> </tr> </thead> </table> <p>CLOSE</p> <p>Books</p> <p>The children like books! If you would like to donate, we would be really appreciate.</p> <table border="1"> <thead> <tr> <th>0</th> <th>\$0</th> <th>\$5000</th> </tr> <tr> <th>Donors</th> <th>Current</th> <th>Goal</th> </tr> </thead> </table> <p>CLOSE</p> <p>Books</p> <p>The children like books! If you would like to donate, we would be really appreciate.</p> <table border="1"> <thead> <tr> <th>0</th> <th>\$0</th> <th>\$5000</th> </tr> <tr> <th>Donors</th> <th>Current</th> <th>Goal</th> </tr> </thead> </table> <p>CLOSE</p> | 0 | \$0 | \$30000 | Donors | Current | Goal | 0 | \$0 | \$5000 | Donors | Current | Goal | 0 | \$0 | \$5000 | Donors | Current | Goal |
|---|--|---------|--------|--------|---------|------|---|---|-----|---------|--------|---------|------|---|-----|--------|--------|---------|------|---|-----|--------|--------|---------|------|
| 29 | \$5672 | \$5000 | | | | | | | | | | | | | | | | | | | | | | | |
| Donors | Current | Goal | | | | | | | | | | | | | | | | | | | | | | | |
| 0 | \$0 | \$30000 | | | | | | | | | | | | | | | | | | | | | | | |
| Donors | Current | Goal | | | | | | | | | | | | | | | | | | | | | | | |
| 0 | \$0 | \$5000 | | | | | | | | | | | | | | | | | | | | | | | |
| Donors | Current | Goal | | | | | | | | | | | | | | | | | | | | | | | |
| 0 | \$0 | \$5000 | | | | | | | | | | | | | | | | | | | | | | | |
| Donors | Current | Goal | | | | | | | | | | | | | | | | | | | | | | | |
| <p>The donate function in the financial fragment:</p> <p>Total Left \$-672 left</p> <p>Payment </p> <p>Amount \$10 \$30 \$50 \$100 \$Other</p> <p>Leave Message</p> <p>Leave Messages</p> <p>CANCEL DONATE NOW</p> | <p>The donate function in the financial fragment:</p> <p>Total Left \$-682 left</p> <p>Payment </p> <p>Amount \$10 \$30 \$50 \$100 \$Other</p> <p>Leave Message</p> <p>Leave Messages</p> <p>CANCEL DONATE NOW</p> | | | | | | | | | | | | | | | | | | | | | | | | |
| <p>The wishlist GUI for the user and the orphanage:</p> <p>FINANCIAL WISHLIST DONOR LIST FINANCIAL WISHLIST DONOR LIST</p> <p>Money</p> <p>Money is essential for the orphanage. Help us!</p> <p>UPDATE CLOSE</p> <p>DELIVER YOUR GIFT</p> <p>Books</p> <p>The children's favorite.</p> <p>UPDATE CLOSE</p> <p>DELIVER YOUR GIFT</p> <p>Book</p> <p>Children's favourite.</p> <p>UPDATE CLOSE</p> <p>DELIVER YOUR GIFT</p> <p>Book</p> <p>Children's favourite.</p> <p>ADD</p> | <p>FINANCIAL WISHLIST DONOR LIST FINANCIAL WISHLIST DONOR LIST</p> <p>orphan</p> <p>orphan</p> <p>Money</p> <p>Money is essential for the orphanage.</p> <p>UPDATE CLOSE</p> <p>DELIVER YOUR GIFT</p> <p>Books</p> <p>The children's favorite.</p> <p>UPDATE CLOSE</p> <p>DELIVER YOUR GIFT</p> <p>Books</p> <p>The children's favorite.</p> <p>ADD</p> | | | | | | | | | | | | | | | | | | | | | | | | |
| <p>The user Profile:</p> <p>MY GOAL</p> <p>MY DONATIONS</p> | <p>The user Profile:</p> <p>MY GOAL</p> <p>MY DONATIONS</p> | | | | | | | | | | | | | | | | | | | | | | | | |

| | |
|---|---|
| | |
| The specific orphanage's forum: | The specific orphanage's forum: |
|  <p>The screenshot shows a mobile application interface for a specific orphanage's forum. It displays two posts: one from user 'Meow Meow' and one from user 'Owatonna'. Both posts include a profile picture, the user's name, the post date, and a message asking for help. A green '+' button is visible at the bottom right.</p> |  <p>The screenshot shows a mobile application interface for a specific orphanage's forum. It displays two posts: one from user 'Meow Meow' and one from user 'Owatonna'. Both posts include a profile picture, the user's name, the post date, and a message asking for help. A green '+' button is visible at the bottom right.</p> |
| The list of orphanages: | The list of orphanages: |
|  <p>The screenshot shows a mobile application interface displaying a list of orphanages. It includes four entries: 'Malabon Orphanage' (with a photo of children), 'Kazakhstan Orph' (with a photo of children and a message asking for help), 'Ghana Orphanage' (with a photo of children), and 'Owatonna Orphanage' (with a photo of the building).</p> |  <p>The screenshot shows a mobile application interface displaying a list of orphanages. It includes three entries: 'Malabon Orphanage' (with a photo of children and a message asking for help), 'Kazakhstan Orph' (with a photo of children and a message asking for help), and a test account 'TestOrphanage' (with a photo of a room and the name 'CheckCheck').</p> |

GUI tests by potential users

A user experience testing is conducted, which focuses on usability. The subject is a potential donor outside the team. During the testing, the issue was given a list of tasks and asked to finish them without help. A team member was responsible for observing the subject's reaction and the way of interaction. At the end of the testing, the below evaluation list is for open questions.

Based on the observation and the evaluation list, the product provides high usability, which is easy to understand and interact. For all tasks, the subject was able to find the button quickly, and the users can do most tasks without making any mistake. There was only one time that the function felt confused and sought help on a developing page, whose functions have not fully finished yet. Also, the product has a high level of error prevention, the system is able to stop users before any error occurs, and users could have a clear understanding of the error based on the alert message.

However, there are still two issues. First, the subject pointed out the inconsistent orphanges' image size and suggested us to resize and uniform the size. The second issue is highlighted by the observer, which is the inconsistent Back button on the donation page. All other pages can return to the previous page using the default system back button, but it does not work on the donation page, where the user has to click on the built-in back button.

| Orph Donation Evaluation List | |
|---|--|
| Questions | Comments/Suggestions Collected |
| 1. What do you think of the contents or information provided in the Orphan Donation? Is it complicated to understand or hard to follow? | The prototype attempts to match the real world well, and the contents are easy to understand. And the order of pages is easily understood as well. However, there is no backorder of the page seems yet because some pages did not have back buttons. |
| 2. What do you think of the consistency of the Orphan Donation? Are there any components using different words but perform the same function? | All information is necessary, useful, and relevant. There is no misleading information and vague elements to use. |
| 3. Will it be hard for you to use the Orphan Donation without any instructions? Will it be hard for you to learn it? | Easy to follow and the connection between the different pages are easily understood, and all of the pages' linkage can be performed by simply clicking the buttons. Besides, the application could provide some instructions and tutorials for the first time. |
| 4. What do you think of the latency time of the app? Do you need to wait for a long time? | The latency time is acceptable. The device can load each image within one second. It would be better if the app can display the image in the orphanges list in order. |
| 5. Are there any instructed error messages to notify you? Are they enough? | Appropriate error messages are prompted and are easy to follow. |
| 6. Does the application give you enough freedom to use, which means you can leave the current state easily? Is the application requiring too many | Most parts are easy to use and acceptable. This app does not require too many permissions from users, which is good. |

| | |
|--|---|
| permissions from you? | |
| 7. What do you think of the whole design of the application? Does it have any redundant elements or information? | Some pages did not have similar or same styles like the button style and background colour while the contents are correct. |
| 8. Any other suggestions for us to improve the Orph Donation? | <p>The functions are flexible and efficient, while more complicated functions are expected to complete, such as payment method, or tracking details of donated objects.</p> <p>There is no such instruction page at first to guide users. Maybe the development team can make a prompt instruction to let users know what we can do within the app, and what we need to do first, like login.</p> |

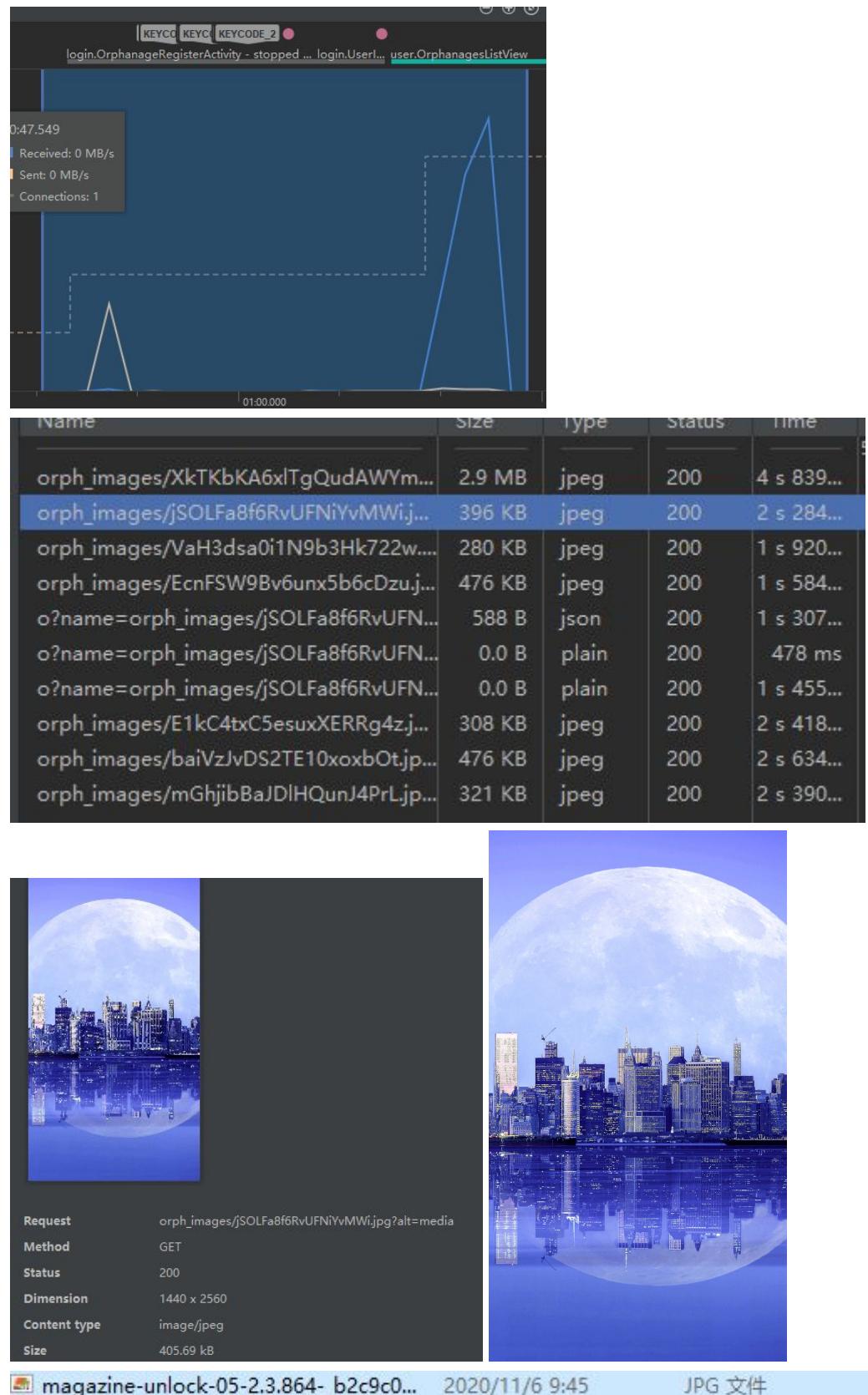
b. Validation of bandwidth/energy saving mechanisms

Orphanages profile and image

In the product, orphanages are allowed to upload a photo of the orphanage when they register. Besides, after logging in, donors can also see a list of orphanages, along with their photo and profile. Therefore, a bandwidth strategy is applied here to save energy and bandwidth.

First, the photo of orphanages would be compressed with 75% compression and converted to jpg, which can still provide high-quality pictures on the mobile phone while saving a certain amount of bandwidth.

Besides, before the application fetches the list of all orphanages, it would check the current network connection status. If it is using WiFi, the application will download the information about new orphanages and update the local cache if necessary. However, if it is using cellular, the application would only read from the local cache and display the saved orphanages. Since the real-time nature of the orphanage's information is not vital in this part, the application will not download any data here if it is using cellular.



Financial Donation Part for both orphanage and normal donors

The financial donation part for both orphanage users and normal donors needs to be updated every time so that they can check with the current status of the proposal donation request.

Thus, we decided to save bandwidth and energy from user experiences and for easier later data access in this component.

For both orphanages and normal donors, we limit the network connectivity type to them. If they want to check the current financial donation of the selected orphanage, the app needs to store all of the information needs in a more secure database. FirebaseStore and they can only get access to the data when having a network connection.

Orphanage

Furthermore, for the orphanage, we also limit the network connection type. Only when they are connecting to a more stable WIFI network, they can propose a new financial donation request.

Normal Users

We do not put a limit of network connection type to them because normal users can make donations whenever and wherever they want. However, they still need a necessary network connection because the data needs to be stored in CloudStore as well.

Also, in this part, we have made another solution to save bandwidth and energy. Apart from uploading the data instantly after the user has made a financial success, our app also keeps a copy of user donations as a history list. By doing this, once the user wants to check the specific number of total financial donations he has made, the data can be easily extracted from the device rather than downloaded from the FirebaseStore, which will save some bandwidth and energy. What's more, since we only keep the current login users' donation data with a particular format and limited field to the Application Cache, it will not occupy much space on the device, which means the computation performance will seldom be affected.

For choosing the internal storage, the Application Cache folder is the best choice. Other applications cannot get access to the data which will protect the data security, and the files would be automatically removed once the users uninstall the application, which will save the memory of the device.

```
Test Wu:WU0uzTNZRM60TULd5vmi:null:50:20201106_071055  
Test Wu:WU0uzTNZRM60TULd5vmi:orphanage town:150:20201106_080803
```

Wish lists for orphanages and donors

For both orphanages and donors, the variable userId would be sent from the login activity or be stored in StoragePreference. The userId is already in the StoragePreference, and there is no need to fetch the userId from firebase redundantly, thus saving the redundant bandwidth.

The wish list fragment does not implement many bandwidth-saving strategies for the orphanage so that the users would get notified instantly. There are two buttons available for each wish list item in the view of the orphanage: add/update and close, every time the orphanage needs to add/update an item or close an item, it has to show the instant modified result in the app.

Forum

The forum is useful to enhance communication between orphanages and donors. To enable both sides to receive timely responses, we made this forum get realtime updates with the database by using FirebaseUI. There is one advantage of real-time data: once the user enters the forum page, it will download all forum messages from the service. Firestore will automatically save data in the cache so that it will only transfer data when there is any update in the service. Since it has been in the cache, even if you disconnect the network, the content in each forum still displays.

On the other hand, considering the size of the content in the forum will get more as the responses increase that people may not be interested in the historical forum content. Therefore, we set a limitation that the forum RecycleView only displays a maximum of 20 messages, and it is also a strategy to reduce bandwidth usage. The left first peak wave in figure 1 is the first time the user enters the forum page that loads 20 responses from the server. The third peak wave is when the back to the forum, and it only transfers a few data to keep the database connection without downloading anything. The last peak wave is when the user added a new response in the forum, and the database will detect the update and download it to local.

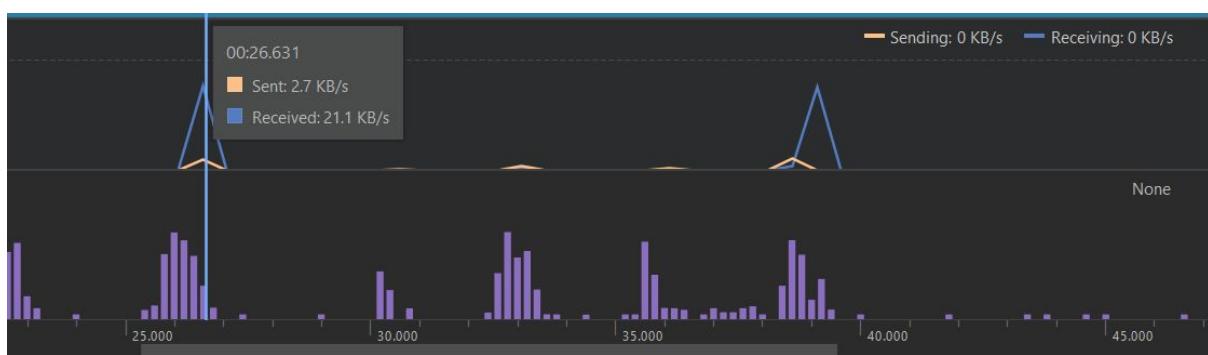


Figure 1 Forum network activity

User profile

The user profile is useful to display each donor's current financial donation and also make them track if the orphanage received their gift or not. Since Firebase supports offline data persistence which means that when you download data from the database, it will automatically cache a copy of your download data, therefore, it can display database content even if your device is offline. In the donation gift track page, we used a Firebase in-build method ‘.addSnapshotListener’ and ‘MetadataChanges’ that the system will fetch data from the catch first and it only download data when there is an update in the database (Orphanage updated the gift arrive information on the server). The function has also been tested in the network profile, and the first wave peak is the user first entered the donation tracking page. After that, no matter what times you reload the page, it won't have any network connection. Also, the second peak wave is when we update ‘arrived’ Boolean value from false to true in Firestore, it will download updated data immediately.

Every time the user successfully donates money, the system will create a cache file to store the user's financial donation information. Therefore, we read the cache file to get the user's current financial donation amount without using any network.

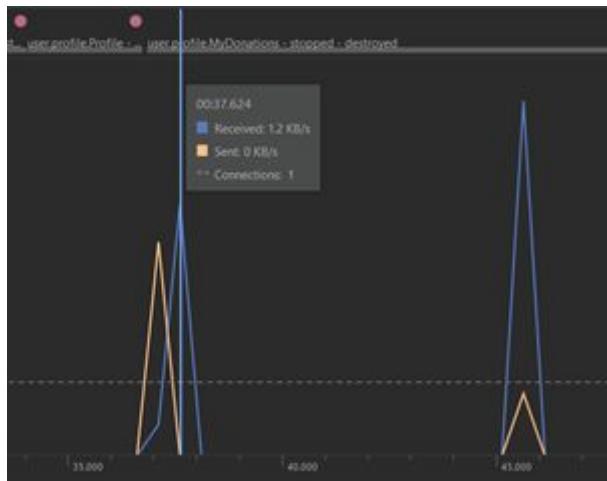


Figure 2 Donation list network tracking

3. Challenges and Setbacks

a. Setbacks - achieved

The credibility of the application

Users may not trust the security and reliability of our applications, and therefore, refuse to provide personal information. To solve that, we migrate the authentication and database to Google Firebase, so that the privacy and secrets are passed to Google, which is a trusted company, instead of being uploaded to our server or database.

The login interface is offered by Firebase AuthUI, so that users could see the standard Google login interface and the app would suggest using the Google App account, which could improve the credibility of the application.

In the future, when the application is in the real world, we will provide a user privacy agreement, explain and promise data security, and further seek cooperation with animal protection organisations or the government.

PayPal - partially achieved

The final product integrated PayPal SDK to send payment. However, some issues make us can't present the real scenarios about donor pay to an orphanage. Refer to figure 3, it needs to manually create buyer accounts in the PayPal dashboard to test payment so that we can't bind a PayPal account to each user in this phase. Also, each payment can only be sent to the owner of the app through a client ID which provides you when you apply a new app instance in the dashboard.

[See about the Sandbox Testing Service](#)

Sandbox Accounts:

Total Accounts: 6

[Create bulk accounts](#)

[Create account](#)

| <input type="checkbox"/> Account name | Type | Country | Date created | Manage accounts |
|--|----------|---------|--------------|---------------------|
| <input type="checkbox"/> orph@gmail.com | Personal | AU | 07 Nov 2020 | ... |
| <input type="checkbox"/> comp5216@gmail.com | Personal | AU | 07 Nov 2020 | ... |
| <input type="checkbox"/> sb-eu9dx3642680@business.example.com | Business | AU | 04 Nov 2020 | ... |
| <input type="checkbox"/> sb-a47qnk3554973@personal.example.com | Personal | AU | 04 Nov 2020 | ... |
| <input type="checkbox"/> zxw971118-facilitator@gmail.com | Business | AU | 05 May 2019 | ... |
| <input type="checkbox"/> zxw971118-buyer@gmail.com DEFAULT | Personal | AU | 05 May 2019 | ... |

Figure 3. Screenshot of sandbox account pool

b. Setbacks - has not achieved

Non-Technical - Cooperation with the government

Since our application is in the 1st version and it still needs to be improved, so it is not ready to have cooperation with the government or any official professional organisations to help us authenticate the orphanage's information in our application.

Nevertheless, once the phase of developing the application is in the next level, in which achieves the high-level of security and professions, it will be possible to have cooperations with some organisations and local governments to help us authenticate the orphanages, which can promise the reliability of the information.

Technical - Payment APIs

Reasons for taking different paths

As mentioned in the proposal, we have tried many different APIs for in-app payment. However, there are various reasons that stop us from finishing this “commercial” function.

Square

Square also provides the official tutorials for Android Developers. However, there are different kinds of problems that happen to us. One of the prerequisites is that we need to enable payment processing with Square account, which requests us to apply for an ABN number and official Business Information.

Prerequisites and assumptions

To build with In-App Payments SDK on Android, the following must be true:

- Your application `minSdkVersion` is API 21 (Lollipop, 5.0) or later.
- You have added In-App-Payments SDK to your Android project. For more information, see [Install In-App-Payments SDK](#).
- You have a Square account enabled for payment processing. If you have not enabled payment processing on your account (or you are not sure), visit squareup.com/activate.

Tell us about your business.

Enter a few details so we can provide specific information about how Square can work for your business.

The form consists of several input fields:

- What type of business is it? (dropdown: Please select...)
- Business Category (dropdown: Please select...)
- Estimated annual revenue (dropdown: Select estimated revenue...)
- How do you plan to use Square? (Optional) (dropdown: Select usage type...)
- ABN (text input: Enter your ABN)
Enter your ABN number, optional for individual use.
- Trading Name (text input: Trading Name)
The business name that will show up on your receipts.
- Business Type (dropdown: Please select...)
- Business Name (text input: Business Name)

At the bottom left is a note: "By continuing, I agree the information is correct." At the bottom right is a blue "CONTINUE" button.

Google Pay

There are many easy followed tutorials, and we have tried to apply them to our application. However, if we want to embed this API in our application successfully, we need to register a formal commercial account for which we need to pay. Thus, it is an experimental procedure for us and prevents us from finishing the actual donation functionality.

Stripe

The Stripe API supports a range of payment methods and is widely applied in many other famous apps such as Google, Amazon, Slack and Zoom¹(Stripe, 2020). It also owns a big scale of open-source community which provides many example demos for different functions²(Stripe, 2020). However, to enable other businesses to accept payments directly, we need to activate the account but an ABN is necessary³(Stripe, 2020). Despite having imported the package to our project, we could not finish the function that allows the donors to directly pay to the bank accounts or wallets connected to the Stripe developer account.

4. Next Steps

a. What misses currently

- i. Authentication - everyone can register an orphanage without any certification
- ii. Payment still in the test phase
- iii. Forum firebase message - The forum only allows to display a maximum of 20 messages that users may miss some critical discussions.
- iv. Delivery tracking - In the current version, users can only know if their gifts arrived or not. It is better to implement a delivery tracking system to enhance user experience so that users see the delivery progress in detail.

b. What needs in the future

Authentication

First, we must understand what procedures people need to go through to establish an orphanage and where to go. Then cooperate with relevant departments to obtain the registration information of the local orphanages. Finally, develop a system to identify orphanage related certifications submitted by users.

Payment still in the test phase

According to the PayPal document, payment to multiple accounts only supports existing integrations which means we may not be able to use it to iterate in our project. The current solution is to update our accounts to live mode and contact the PayPal support team. If it still doesn't work, we need to find a technician who is proficient in back-end server establishment to help us develop the user donation function on the Braintree platform.

The alternative way is to manually collect the donor's money and transfer it to the corresponding orphanage account.

Forum firebase message

Since we limited the database transfer for each forum to 20 to save bandwidth, we need to implement a reminder mechanism to notify other users if an orphanage's manager sends any message in the forum so that they won't miss any important announcements. The Firebase cloud message can send a targeted notification to users' phones.

Delivery tracking

Integrate a delivery tracking API so that users know the delivery progress.

There were many GUI improvement feedback from user tests. Except for the technical side, we plan to redesign the user interface to increase the usability of this app.

5. References

[1] Stripe.com. 2020. *Online Payment Processing For Internet Businesses - Stripe.*, viewed 5 November 2020, <https://stripe.com/au>

[2] GitHub. 2020. *Stripe Samples.*, viewed 5 November 2020, <https://github.com/stripe-samples>

[3] Stripe.com. 2020. *Enable Other Businesses To Accept Payments Directly*, viewed 5 November 2020, <https://stripe.com/docs/connect/enable-payment-acceptance-guide>

6. Appendix

a. Documentation

b. Manual

Connect to the internet all the time whatever you are using the emulator or physical device(important)

Open Android Studio (Our system and software configuration)

Android Studio 4.1

Build #AI-201.8743.12.41.6858069, built on September 24, 2020

Runtime version: 1.8.0_242-release-1644-b01 amd64

VM: OpenJDK 64-Bit Server VM by JetBrains s.r.o

Windows 10 10.0

GC: ParNew, ConcurrentMarkSweep

Memory: 1237M

Cores: 8

Registry: ide.new.welcome.screen.force=true

Non-Bundled Plugins: com.thoughtworks.gauge, org.intellij.plugins.markdown, com.intellij.marketplace, org.jetbrains.kotlin

SDK Version → Android 11.0(R), API Level 30

Virtual device setting

Pixel XL API 30(resolution of 1440x2560:560dpi)

System Image:

Release Name R, API Level 30, ABI X86, Target Android 11.0(Google Play)

File → New → Import Project

Update Gradle to the latest version

Change SDK location to your configuration(invalidate and restart, Android Studio would auto-change your sdk location)

Start Emulator

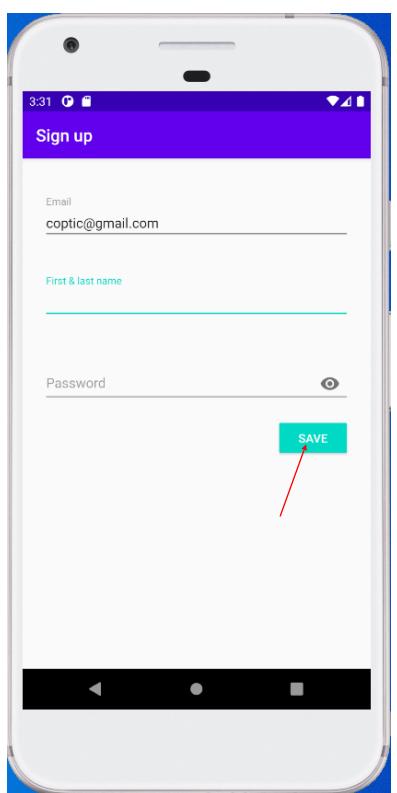
1. Appendix

a. Documentation

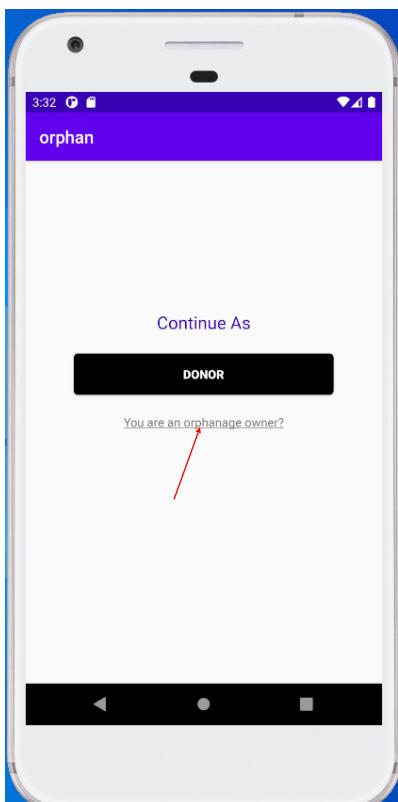
User Manual for Orphanage Owner



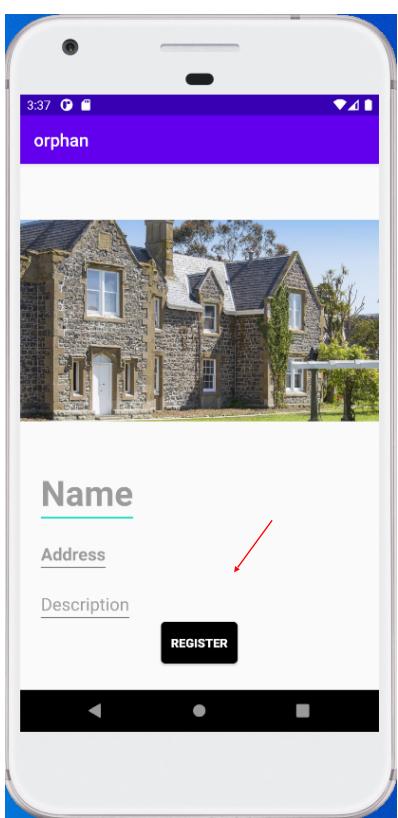
This is the first screen when we first open the application, you need to fill in your email address to sign in or sign up. Our system will automatically check your email in our database, if you are a new user you will be led to the sign-up page. If you are already an existing user, you need to type in your password and login. Click NEXT button after you type in your email.



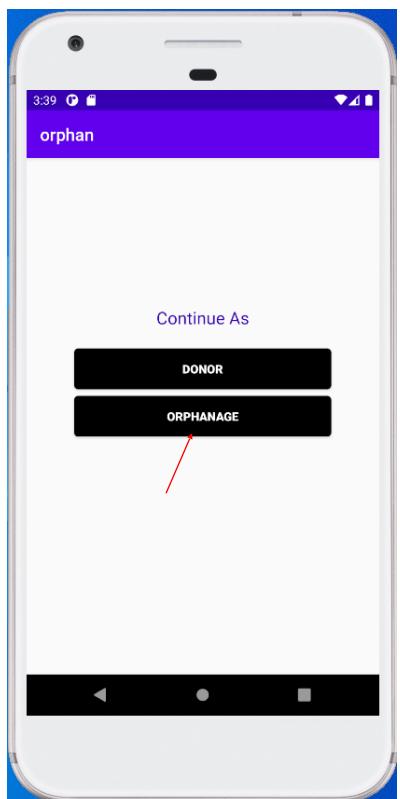
This is the page for new user sign up. You need to fill in your name and set your password. Then click SAVE button to go next.



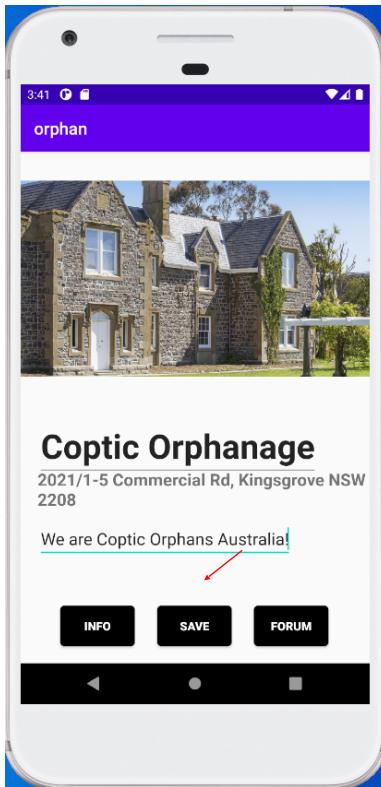
Then we will choose as a donor or an orphanage. If you choose orphanage, you need to fill in the related information. Click "You are an orphanage owner?" to go as an orphanage user.



Then we will go to the orphanage register page. You need to fill in the related information of your orphanage. And you need to upload an image of your orphanage. Here I have already uploaded an image. Click REGISTER button to go next.



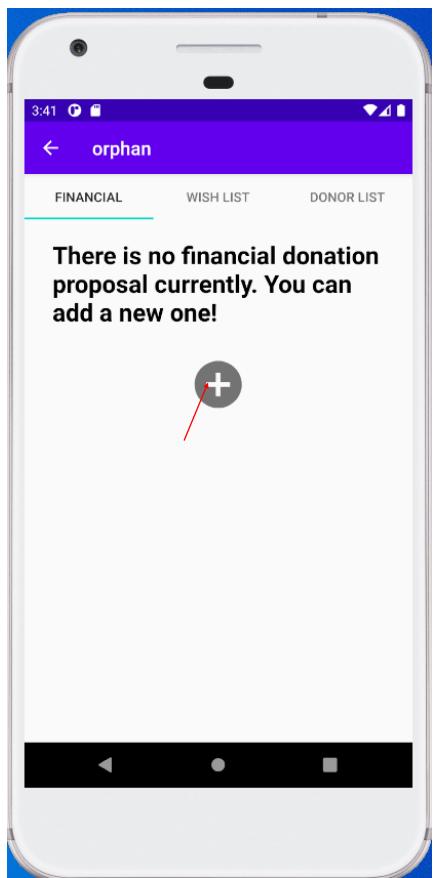
After finishing the orphanage registration, we will back to the role selecting page. And you can see the orphanage choice becomes a button. Now you can click the ORPHANAGE button and go as an orphanage user.



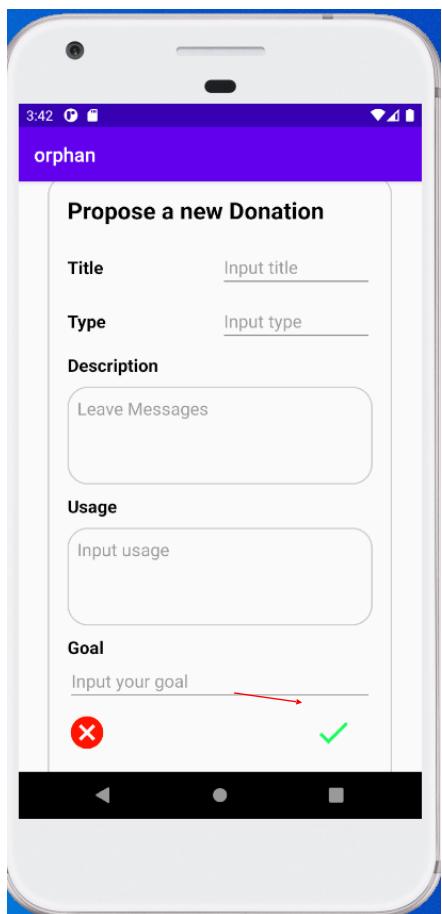
We will go to the orphanage page. Click the little pencil icon to edit your information. And you can click SAVE button to save your changes.



Click the INFO button to check and edit all the information of your orphanage.



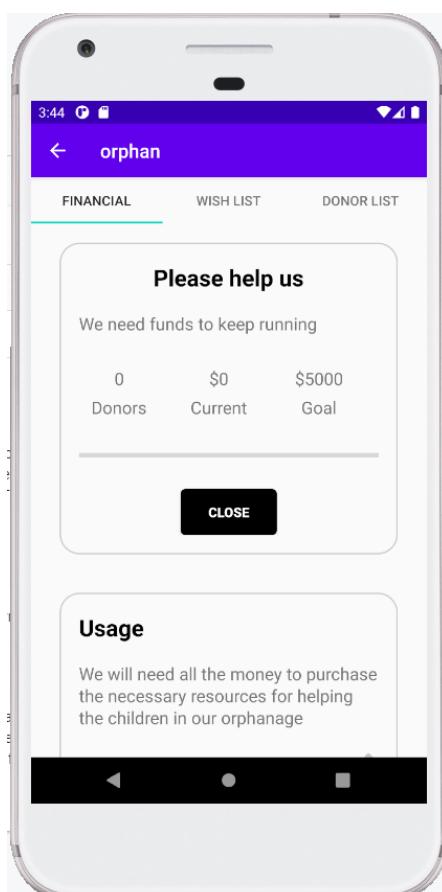
Here are 3 components we have for each orphanage. They are Financial, Wishlist and Donor List. First, we will go through the Financial. This is where you can create your financial proposal to allow donors to donate money to you. Donors can see your post on their page, and they cannot donate money to you unless you create a financial proposal. Click the plus button to create a new financial proposal.



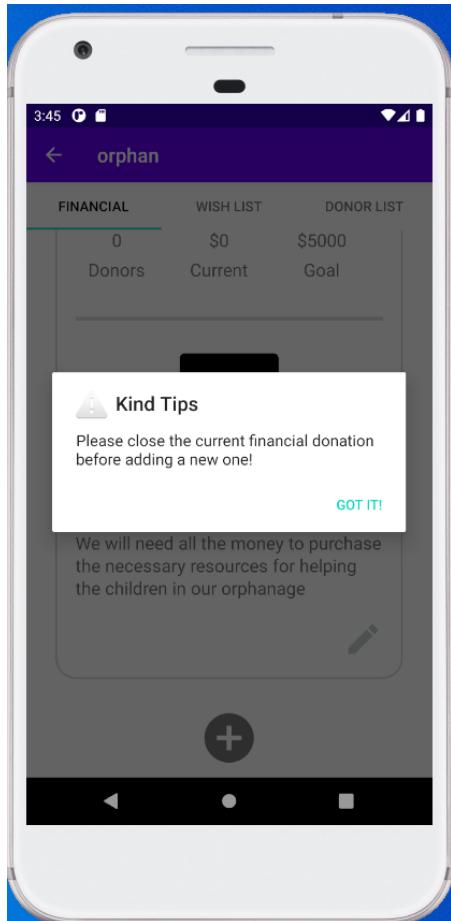
This is the financial donation proposal page. You can edit your title, type, description, usage, and goal here.

Goal is the money you want for this donation. For example, you want to have a 5000 dollars donation from the donors, you type in 5000 in the Goal.

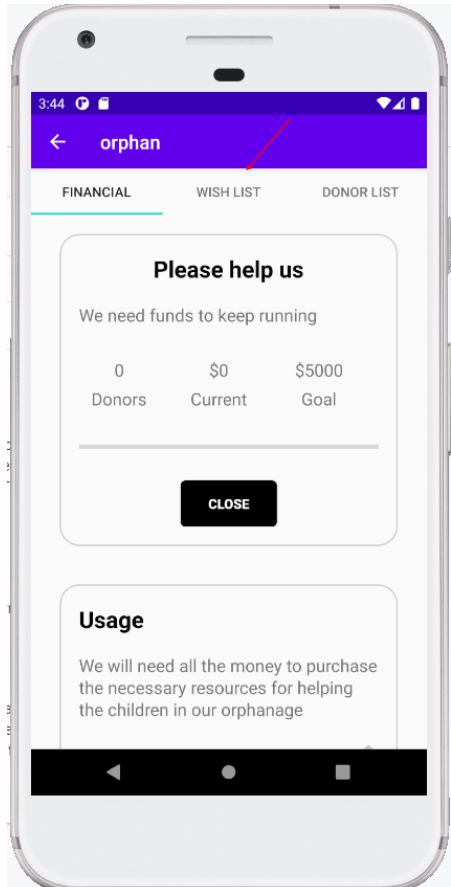
Click the red cross button to cancel this creation and click the green tick button to finish creating this new proposal.



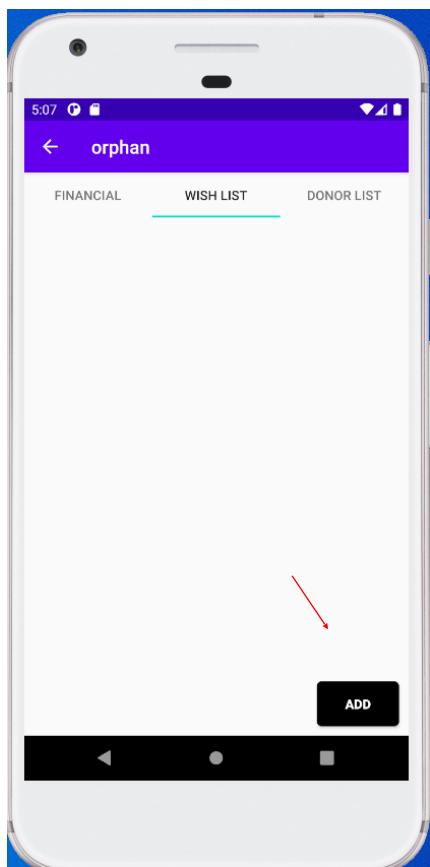
Now we can see the financial proposal we created in the last step. You can see how many donors have donate for this post, the current donation amount and your set goal. You can click close button to terminate this proposal at any time.



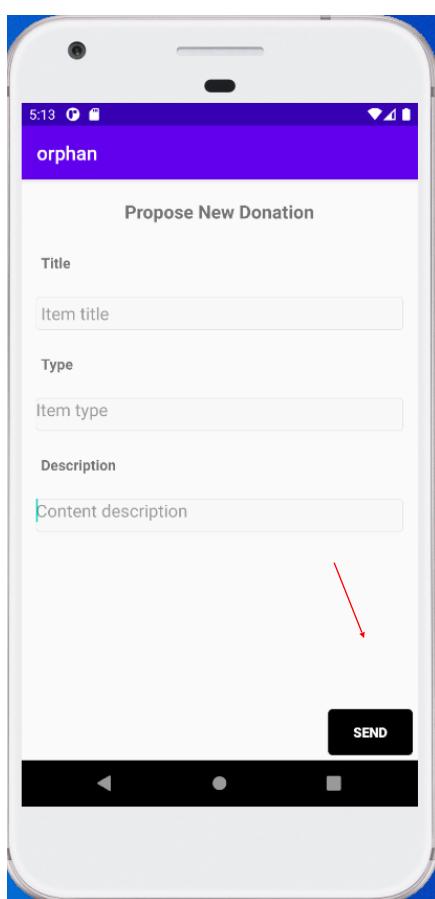
Each orphanage can only have one financial proposal at a time. If you try to create another one while you already have one ongoing proposal, you will get an error message.



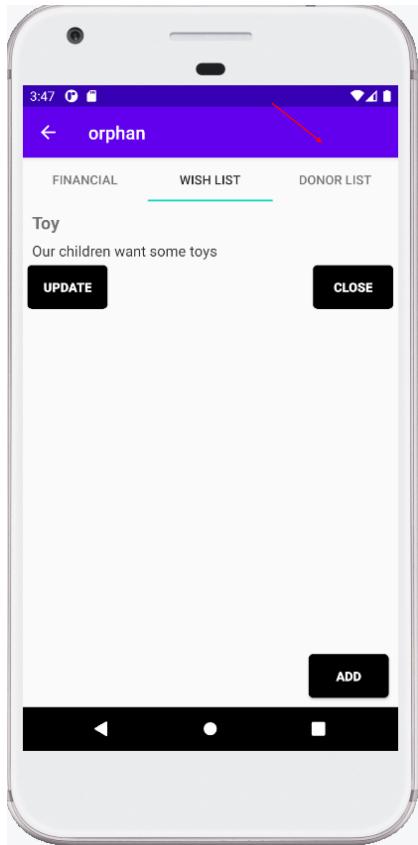
Now click the WISH LIST button on the navigation bar to go to the wish list page.



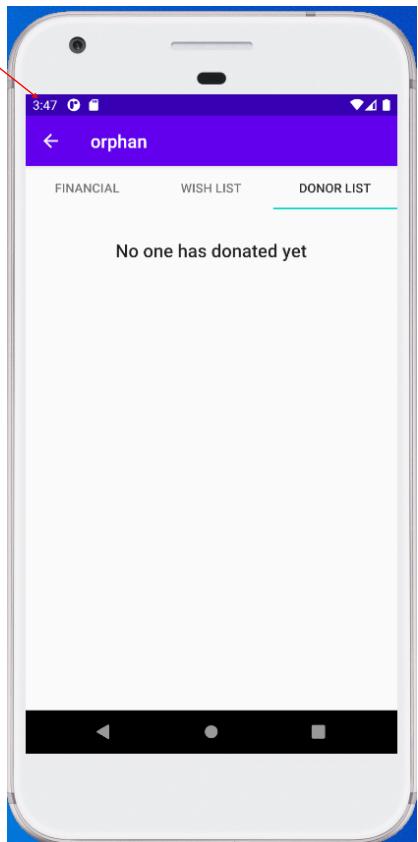
This is the wish list page. Wish list is the non-money resources you want donors to donate to you. Click the ADD button to create a new wish list.



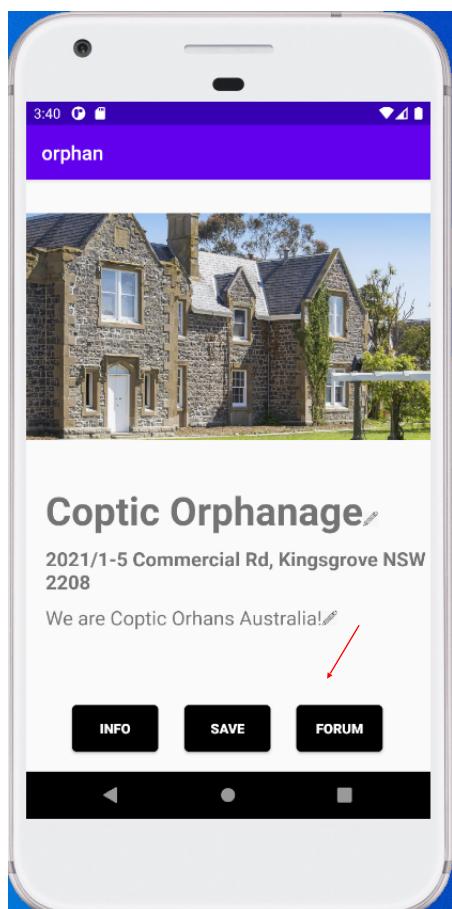
In this page, you can edit your details of what you want for this wish list. You can ask for some donation such as toys, medicine, etc. for the wish list. Click the SEND button to finish creating the new wish list.



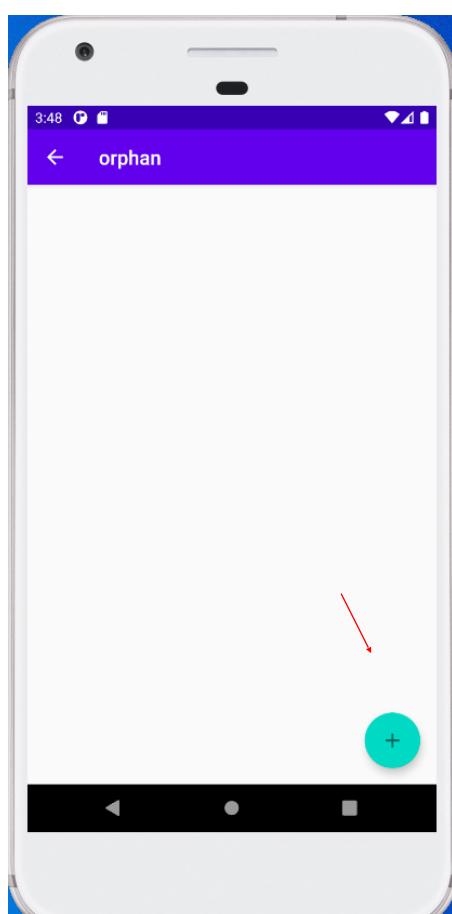
Now we can see the wish list we just created. We can click the UPDATE button to edit the information of this wish list and click the CLOSE button to terminate this wish list. Then you can click the DONOR LIST button on the navigation bar to go to the donor list page.



In this page, you can see all the donation history from donors. It will display the donor name, donation date and how much they donated. Currently, there is no one donated yet, so it will show "No one has donated yet". Now we can click the back button on the top left corner to go back to the home page of the orphanage.

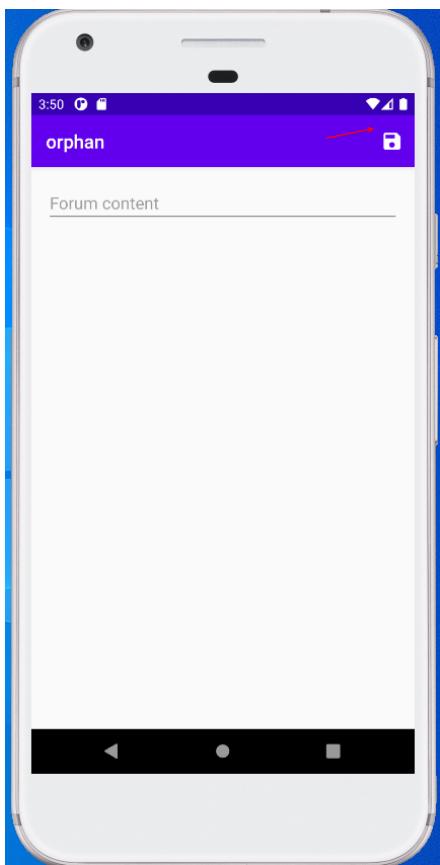


At the home page, you can click the FORUM button to go to the forum page.

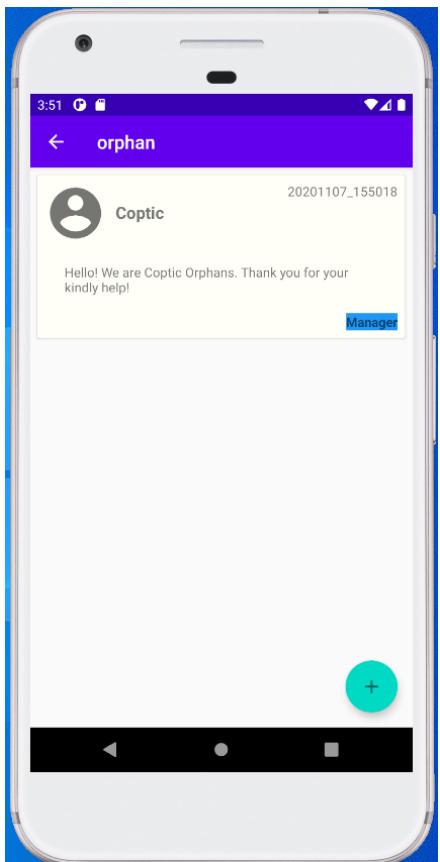


This is the forum page. Orphanage user and donor user could both posts here and all the posts are visible both sides. Click the plus button to create a new post.

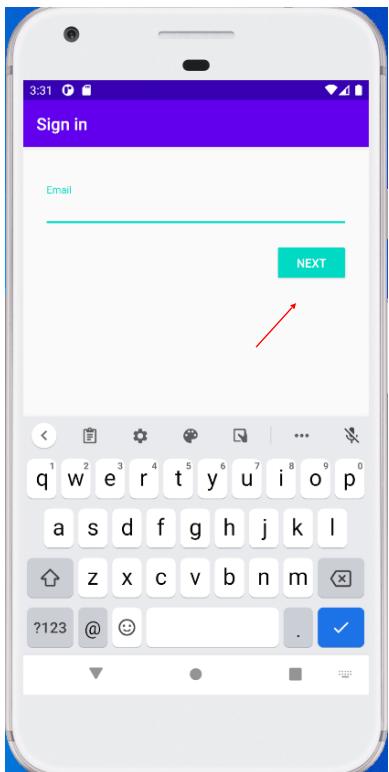
Fill in the content you want to say in the post. Then click the save button on the top right corner.



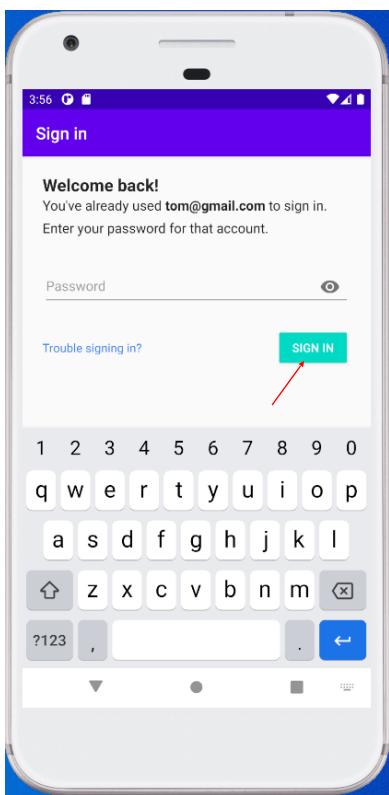
Now we can see the post in the forum page. This is also visible at the donor side. As we are the orphanage owners, there will be a manager mark at the bottom right corner.



User Manual for Donor User

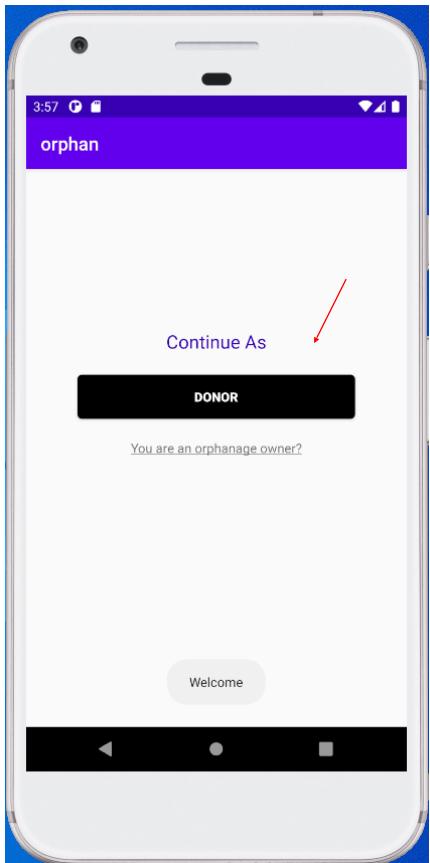


Now let's try to go as a donor. First, we need to login with the email address. This time I will use an existing account to demonstrate. Click NEXT button to continue.

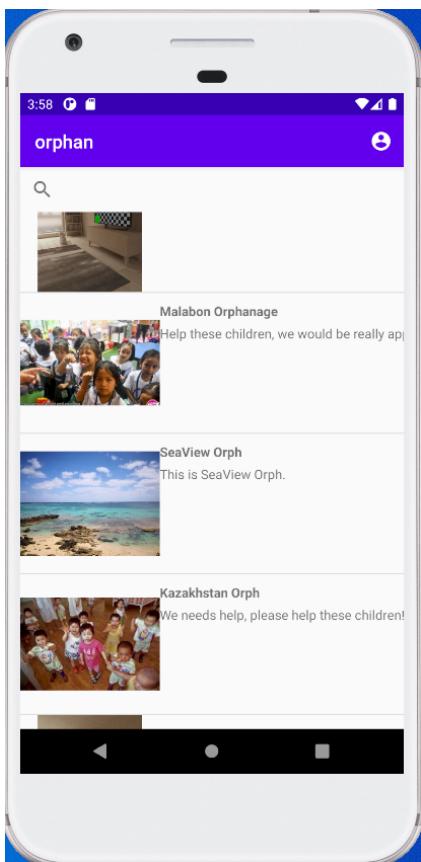


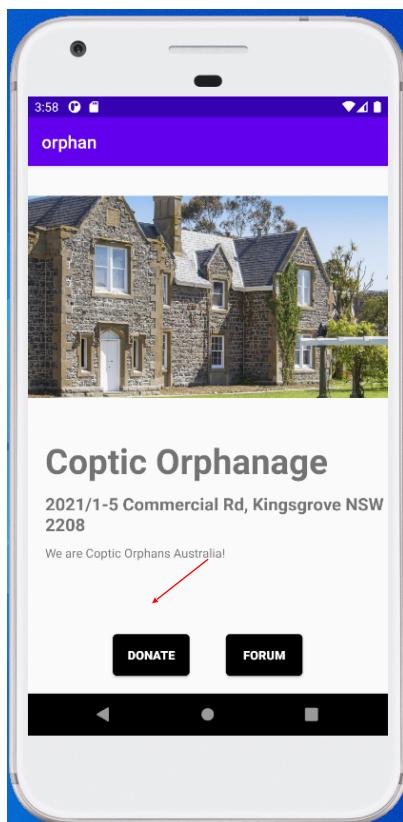
This time we directly jump to the screen let you enter your password because the system finds this account that exists in the database. We don't need to register anymore. Enter the password and click SIGN IN button to go next.

Click the DONOR button to continue as a donor user.

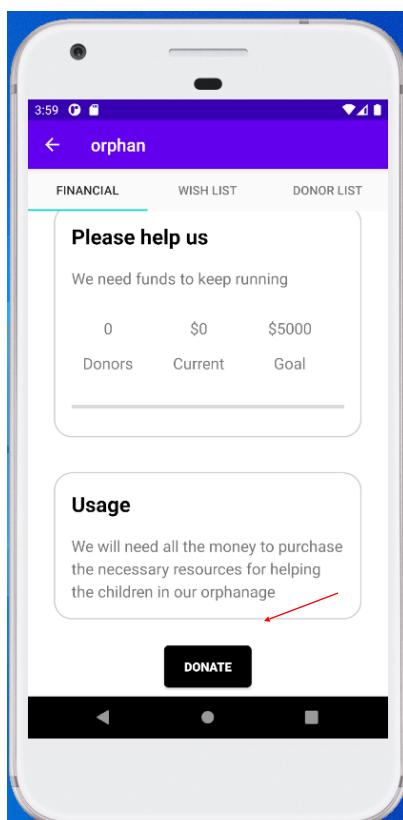


This is the first page for the donor user. It lists out all the orphanages that have been registered in our application. You can scroll down to check each of them. Click any of them then you will go to the page for that specific orphanage. Now let's select the orphanage we just created in the orphanage user manual step.

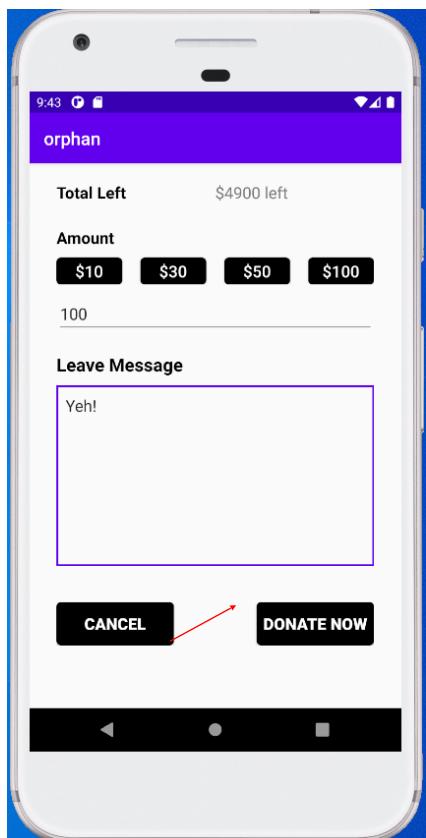




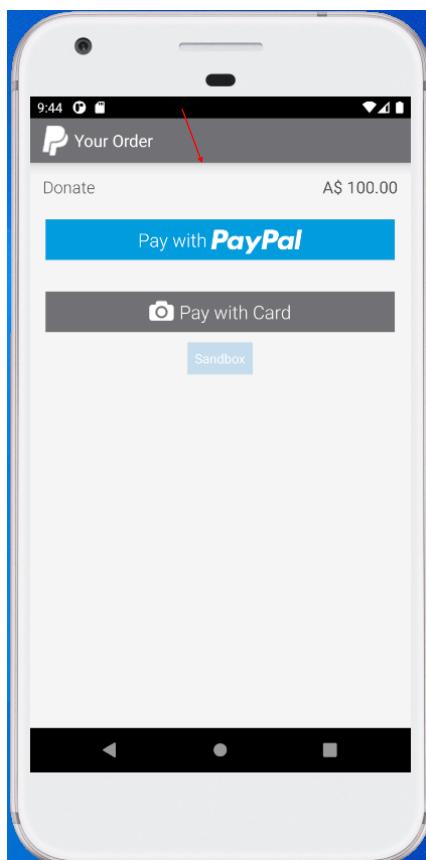
This is the Coptic Orphanage we created before. We can see its information. Click donate to check the detailed donation information of this orphanage.



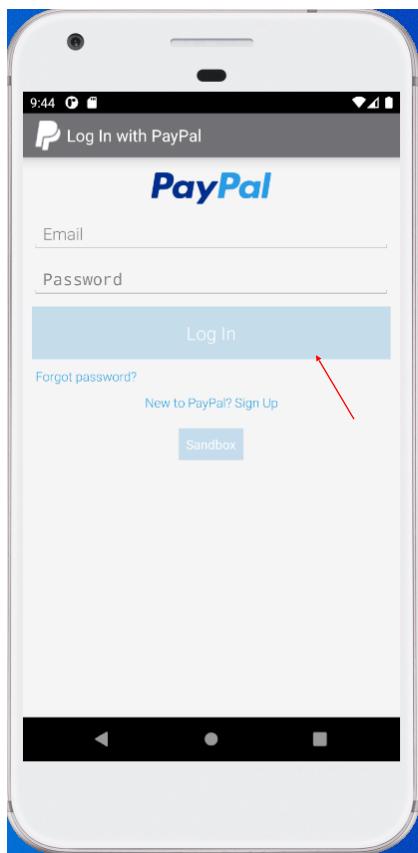
We can see that there are three same components at the donor user side which are FINANCIAL, WISH LIST and DONOR LIST. In the financial page, we can see the financial proposal we created before with the 5000 goals. Now let's try to donate some money for this proposal. Click DONATE button to start your financial donation.



This is the donation page. At the top, you can see the total left amount of this proposal (I already donate \$100 for testing, so it only got 4900 left here). You can choose the amount you want to donate. Right now, we only have a fixed amount for the user, in the future customized amount will be available. Then select an amount, fill in the message and click DONATE NOW button to go to the payment page.

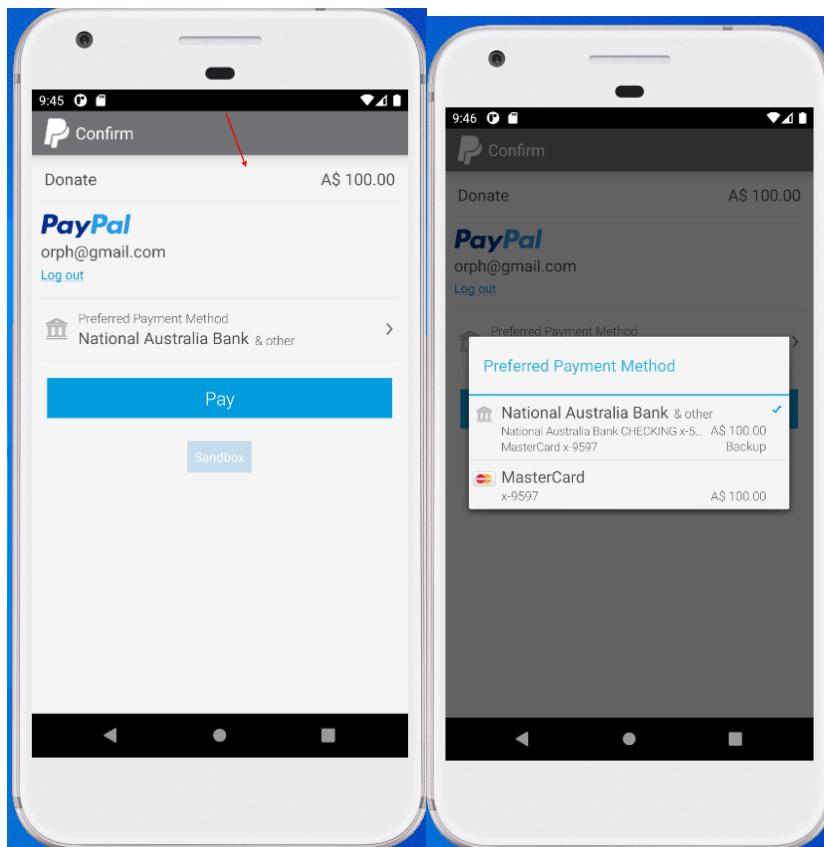


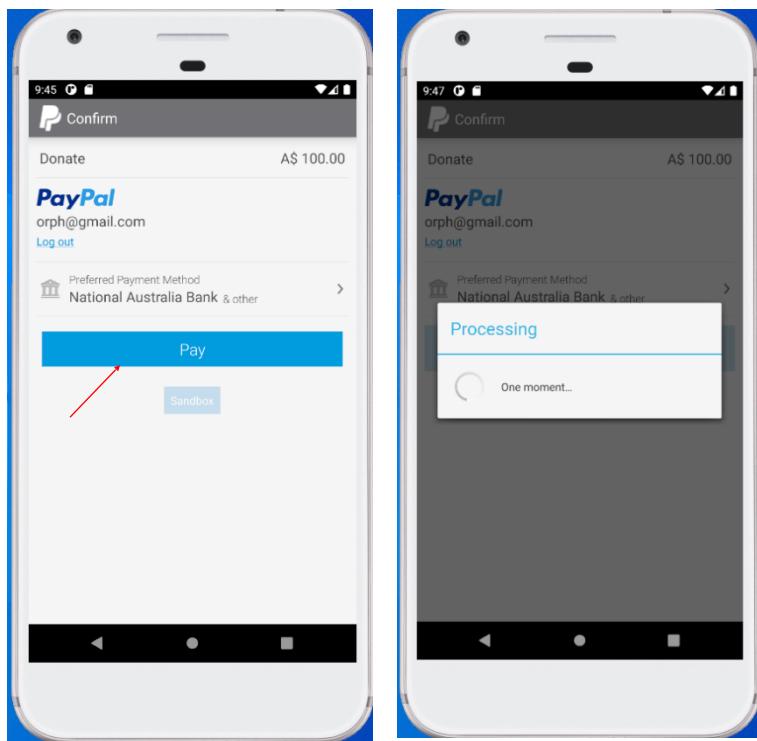
We can use PayPal to finish the payment or a credit card. Payment with Credit cards needs to use the camera scan function which I could not demonstrate on the emulator so I will only demonstrate the PayPal payment here. Click Pay with PayPal to continue the payment.



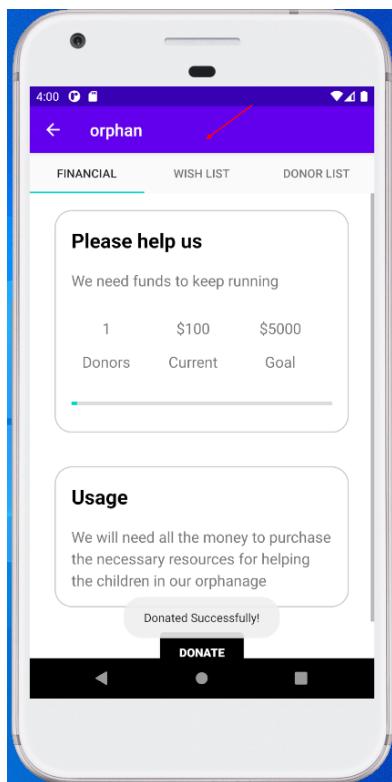
Enter your PayPal account detail (Here we are using the PayPal sandbox account for testing purpose. The account email address is orph@gmail.com. The password is 12345678). Then click Log In button to go next.

This is our PayPal account page. We can see the donation amount at the top. And we can select our payment card.

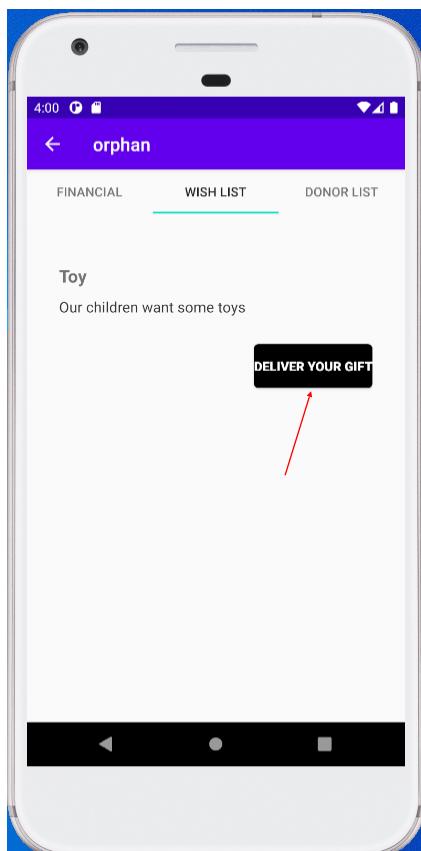




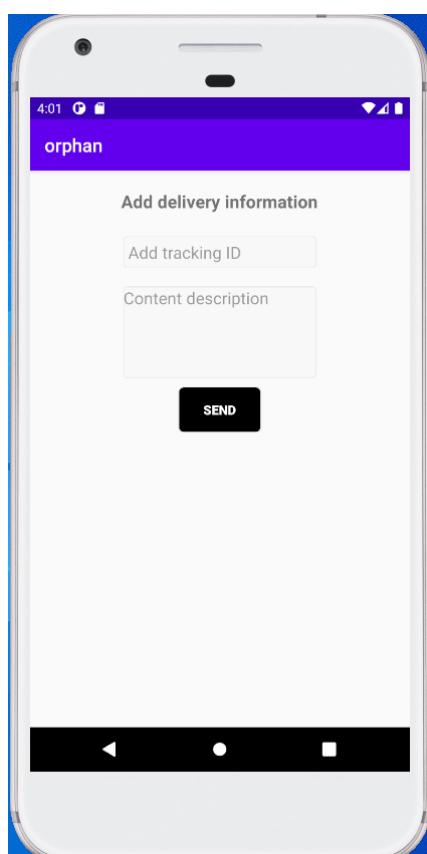
Then click PAY button to finish this payment. We need to wait for a few seconds to finish the processing.



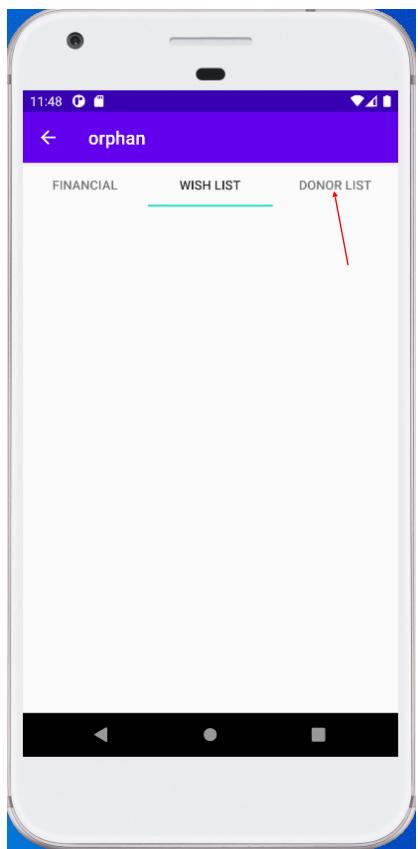
Here we go. We donate for \$100, and a message will pop up at the bottom tells you that you have donated successfully. We can see the Donors becomes 1, and Current becomes \$100 because we donate for \$100. You can have multiple donations as you like. Just click the DONATE button and repeat the previous step to make another donation. Next, let's go to the wish list page by clicking the WISH LIST button at the top navigation bar.



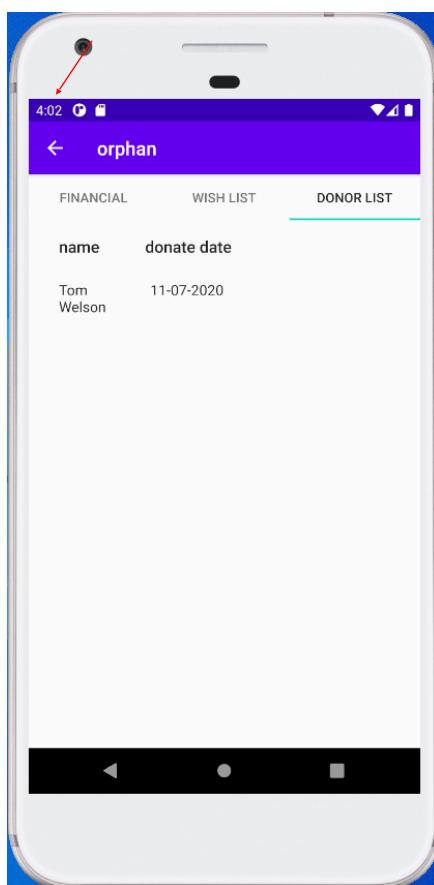
Here we can see the wish list we just created. We can click the DELIVER YOUR GIFT button to deliver some non-money resources or gifts to the orphanage.



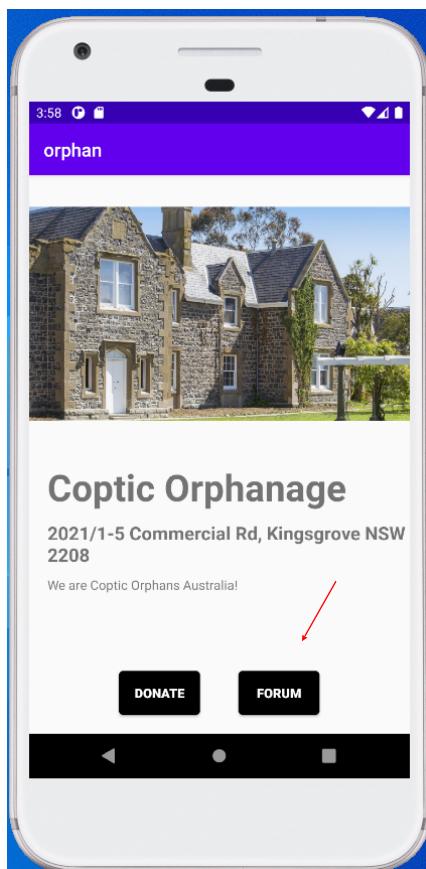
Notice: This part hasn't been fully functional yet. So, there will be no actual delivery happens in our process. It will be completed in future development. The basic idea is that you can deliver the resources to the orphanage and enter your tracking ID and content description here so the orphanage will know you have been delivered something to them, and they can track for it according to the tracking ID. Click SEND if you have delivered something to the orphanage and fill in your tracking ID and description.



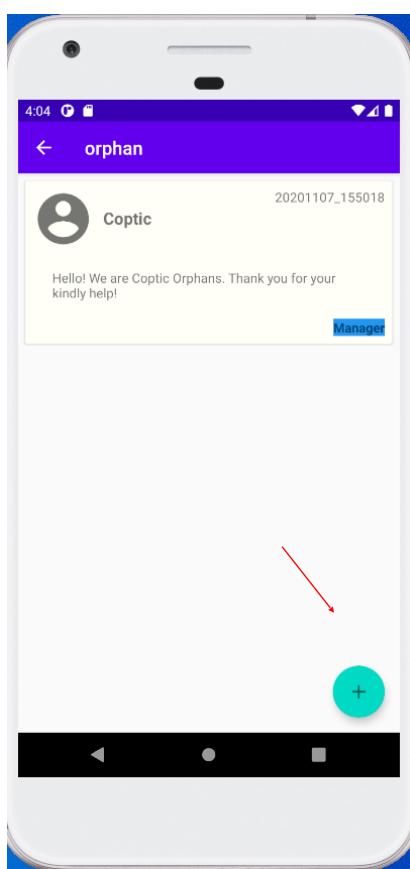
Once a donor has sent the delivery to a wish list. This wish list will be hidden. The orphanage can track for it and close this wish list once they receive the delivery. Or they can update it if the donor has delivered the wrong resources as well. Now we can click the DONOR LIST button on the top navigation bar to check the donor list.



This is the donor list page for the donor side. It will display all the donation history of this specific orphanage. The difference is that donor cannot see the actual amount for each donation because this list is visible for every donor, and we don't want to let them think that we are trying to force them to compete with each other. Click the back button on the top left corner to go back to the orphanage list page.

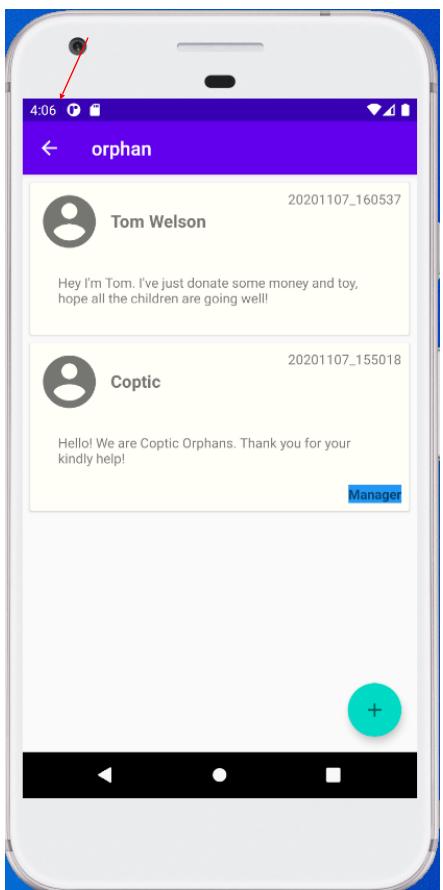


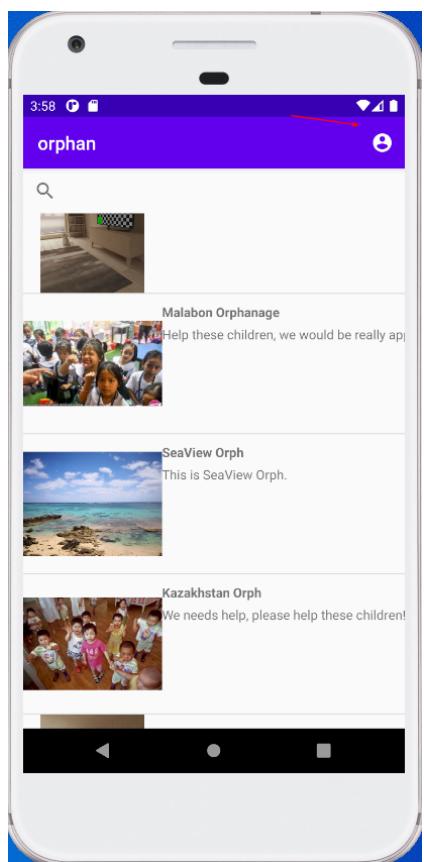
Now let's select this orphanage from the list again and check the forum function from the donor side. Click FORUM button to go to the forum page.



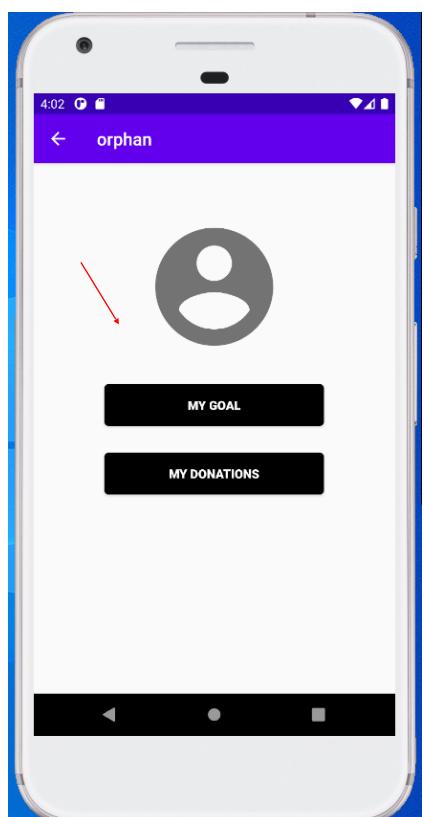
We can see the post from the orphanage we just created. And the donor can also create a post as the orphanage do. The process is exact the same. Click the plus button at the bottom right corner to create a new post.

Fill in the content you want to post and click the save button on the top right corner.

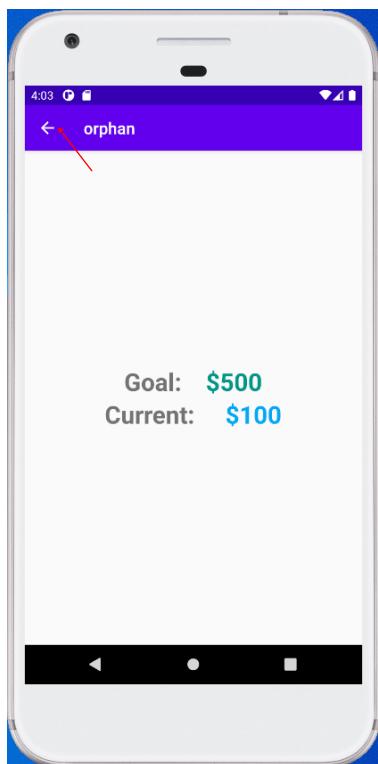




At the orphanage list page, we can see there is a button on the top right corner. We can click it to go to the user profile page.

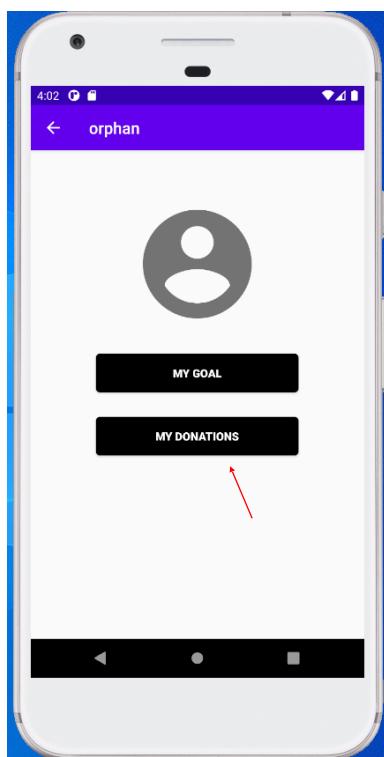


This is the user profile page. Click MY GOAL button to go next.

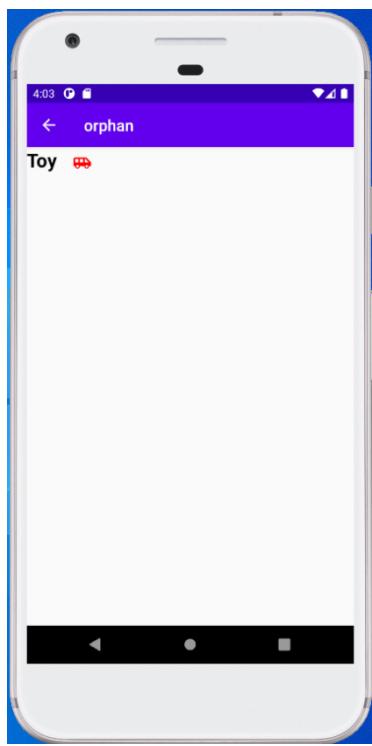


In this page, we can see how much money we have already donated in total. The goal is currently fixed, and it will be editable in the future so the donor can set their monthly goal.

Click the back button to go back.



Click the MY DONATIONS button to go next.



In this page, we can see the non-money resources we have sent to the orphanage. Currently, we have just sent some toys to the orphanage.

That's it for our current version's functions. We will develop and perfect more functions in the future to bring a better user experience to the user.