

VICTORIA UNIVERSITY OF WELLINGTON
Te Whare Wananga o te Upoko o te Ika a Maui



School of Engineering and Computer Science
Te Kura Mātai Pūkaha, Pūrorohiko

PO Box 600
Wellington
New Zealand

Tel: +64 4 463 5341
Fax: +64 4 463 5045
Internet: office@ecs.vuw.ac.nz

Community Exchange Project

Jareth Gaskin

Supervisor(s): Jennifer Ferreira

Submitted in partial fulfilment of the requirements for
Bachelor of Engineering with Honours

Abstract

In response to the social and financial troubles that come in the wake of moving to a new location, the concept of the app Thyme was developed. This application is a hybrid of resource sharing ideology's such as tool libraries and gamified points logic to promote the sharing and using of community resources. Fundamentally, it aims to overcome the initial challenges of meeting up with new people in the local community by simplifying the communication processes, along with removing obstacles that stand in the way.

Table of Contents

1. Introduction	1
1.1. Goals.....	1
1.2. Contributions	1
1.3. Layout.....	2
2. Background and Related Work	3
2.2. Risks of social media.....	3
2.3. Existing work	4
3. Design.....	5
3.1. Design constraints	5
1.1. Finalised requirements	5
3.2. Design Architecture.....	7
3.2.1. Requests page	7
3.2.2. Messaging page	8
3.2.3. Community page	9
3.2.4. Profile page	10
3.3. Design alternatives	12
4. Implementation.....	14
4.1. Framework choice	14
4.2. Architecture	14
4.3. Platforms	15
4.4. Remote database.....	15
4.5. UI/UX.....	16
5. Evaluation.....	17
5.1. User study.....	17
5.2. Conduct	17
5.2.1. data collections and ethics	17
5.3. Results	18
5.3.1. Measureable results	18
5.3.2. qualitative results	18
5.3.3. Points of interest.....	19

5.4. Improvements made	19
5.4.1. Achievements list	19
5.4.2. Tiered leader board.....	19
6. Conclusions and Future Work.....	20
6.1. Summary	20
6.2. Future work	20

1. Introduction

There is a growing movement of a 'sharing economy' [1] that have made large advances towards improving sustainability and reducing costs when it comes to common resources. By promoting access over ownership, this collaborative economy encourages participants to seek available resources from others over purchasing new material. This system has been evolved in cases to include its own currency and extend from only sharing resource, to providing skills and talents for the benefits of others. Along with cost savings, there is consistent evidence across literature that express the benefit to mental health and well-being [2]. Social bonds were strengthened across the communities that participated, and larger networks of friendly groups began to arise. People began to feel more confident having open conversations with strangers, and less isolated by those in a close vicinity to the area that they lived. However, especially in smaller communities that adopt this system, it creates and ungoverned currency that comes with an array of risks that can quickly result in an unsustainable system.

In this project, a user-friendly interface is created to handle many of the problems that would cause discord, and help create a sustainable environment for exchanging skills, knowledge, and resources. Furthermore, it aims to bring communities closer by creating open forms of communication. Taking these features into consideration, a mobile app was developed to handle these capabilities and provide better support for new communities to join.

Being built on established systems like Time-Banking, and tool libraries, this app is not made unique by concept alone. Thyme is a skill and resource sharing application that incentivizes participation through gamification reward mechanics. Presenting it in the form of a mobile app increases its availability, therefore making it more inclusive. The focus is to make it easier to establish a working system when entering a new community than it is to do so using a tool that already exist.

The need for this communication platform arises from the troubles faced by people that move into an unfamiliar environment. If it is moving from one neighbourhood to another, or moving to a whole new country, it can be hard to become a part of a new community. Therefore, the first point of interest that is approached is giving equal opportunity for those who are a part of the community, irrespective of if the person has been a part of the community for a while, or someone who has recently joined. Thyme's multiple communication channels allow people to start new relationships and connect with those around them.

The next element that is addressed by this solution is the issue of individual wealth. By creating a shared economy (sharing tools, skills, and other resources), people can feel more included as part of the community, as well as saving expenses. Thyme gives users the ability to borrow and use items that they would have otherwise had to buy for themselves. In many cases it could be for something that only happens once a year, like power washing a driveway, or borrowing a car while theirs' is in for servicing. The importance of this app will heavily favour the social and mental well-being that results from a strong community, but it also provides convenience to those who need things in the moment, as is the reality of twenty-first century living.

1.1. Goals

The aim of the project was to develop a mobile application, platform independent, that has the following features:

- 1) allows new members of the community to be able to communicate with existing members.
- 2) Trade in goods and services with other users of the app
- 3) promote the sharing of resources to encourage users to freely give and take rather than buy new.

1.2. Contributions

The results of conducting this project should provide the following for future development in the given area.

1. Functioning prototype that will be used for visual appraisal and run prototype testing on.
2. A concise topic evaluation that can be used to further research in the specialised area.

1.3. Layout

The remainder of the report will entail:

- Background knowledge on the topics of digital currencies as well as active movements to apply shared economy logic to real world environments
- Design walks you through the design process and the choices that were made at each step of the project's development – including changes that were made based on the clients' personal tastes and revelations made through the use of the prototype's beta testing
- Implementation is an explanation of the technical aspects of the project
- Evaluation will present the results gathered through user testing of the prototype, and the changes that were implemented based on the feedback given.
- Conclusion and future work will be discussed in the closing section

2. Background and Related Work

This section covers the background information regarding exploration into shared economy as well as similar technologies in place. Examples of time and resource sharing activities are examined to provide insight into what makes them work. Long standing social media platforms are observed to make comparisons to the implementations expected in this new app.

2.1. Shared economy

Sharing economy can be simplified down to peer-to-peer sharing. It effectively takes out the governing body that controls monetary value in the community that a person may exist in. Examples of this include ridesharing and sharing knowledge [3], which can evolve into the creation of a LETS (Local Exchange Trading System) [4]. An approach to creating a stable shared economy that has succeeded in the past is time banking, where people trade their time for what others can provide. This time exchange is usually represented by a token (usually equivalent to an hour's work) that can then be traded for someone else's time.

There are many informative websites that explain the concept of time banking applications, as well as giving a way for people to join in. What many of them have in common though is that they use a third-party application to organize themselves. Facebook, emails, and platforms such as slack are used to organize meetups and do workshops that encourage people to join in. The way that Thyme aims to make itself stand out from other time banking systems is by making it an all-inclusive application. While the traditional workshop and meetup organizations can participate in raising awareness of the application, once somebody has acquired the app, they are able to join any part of the system without external help.

Past examples of shared economy that have been successful in developing into businesses include Airbnb, as well as Uber. The common factor is that they took a concept that has been around for a long time and used the new technologic development into the Internet of Things movement to make communicating with people who could be potential customers. These examples suggest it may be possible to take the tool library concepts and use the Internet of Things technologies to develop an inclusive environment. In this environment, people can freely trade in goods and services with greater confidence that they are in a safe environment, with greater guaranties of security than was previously provided in these systems.

2.2. Risks of social media

The social and mental welfare of the people that use time banking applications have been proven to increase, and this can also be seen across research into social media applications [5] social media is one of the fastest growing social networks and has maintained itself as the greatest communal networking system.

It is these factors that will be replicated in Thyme. Thyme will encourage people to be more open to their surrounding community, by asking people for help and offering what they can in return. While this is not going to be explicitly tested during the rounds of usability testing, it will be an essential part of bringing the communities together in future use.

With these benefits, there are also corresponding risks, on social media, there are many risks that people are opening themselves up to. The ones found that pose the most concern is emotion-related, sociological, and health-related. Having a direct channel established between people that are not in immediate vicinity to each other can lead to cyber bullying, and unattractive behaviors that would not take place in face-to-face situations. This is a common factor across many of the most popular social media sites that were observed. The most common method of combating these issues is active administration of the network taking control of the situation. They usually can blacklist and remove people who show these behaviors. In the app that will be created, the approach that best fits the

situation is the ability to react independently to such behaviors. Users of Thyme will have the opportunity to block people who are trying to contact them, denying their requests.

Unfortunately, these measures do not carry over to physical protection. With previous examples of shared economy, people are required to advertise their current location. Uber uses this to find the person they are picking up, and as for bed-and-breakfasts, the host allows people into their homes. Though there are measures that can be taken in the latter case [6], few of these protections will apply to Thyme's requirements in the effort to protect the users.

Alternatively, examples of safe behavior for meetups that would increase the safety of the users can be found from unexpected places such as dating apps [7]. Fundamentally, there are similar premises such as meeting up with strangers, and for the most part, this is a dangerous aspect similar in the two apps.

The security measures put in place by the developers of the dating apps include creating digital breadcrumbs, and invasive techniques that retrieve data from the person's phone to ensure that the profile is being used by someone with the correct intentions.

For the current prototype, this is not something that is able to be incorporated, as some of the techniques are beyond the bounds of what is expected to be tested. However, there are other techniques that rely on the users to put into place. These apps recommend that (at least for the first time meeting a stranger) it is important to let a third party know where you will be going, or that someone will be approaching your place. For the exchange of resources, it may be good practice to meet in mutual territory, somewhere that is familiar to both parties and has observers. This can mitigate many of the risks for everyone involved.

Making use of these already setup guidelines would be a good start for Thyme. Having an informative page that appears once, or even each time a new person is recognized through your interactions would be a safe way to help the end users of the app.

2.3. Existing work

The closest product to Thyme is the app Cashless, created as part of "The Together Project;" an idea that is seeking the same results that Thyme hopes to replicate [8]. Rather than following a scheme that exchanges work for a currency, which can be found in many of the other economic apps, The Together Project's team have their own system for rewarding the users.

Where the Thyme differs from Cashless is the result that takes place after the skills or resources have been exchanged. Cashless seeks to create a virtual map of "gratitude" to acknowledge those in the apps community who are helping each other. While visually appealing, the Client of this projects chose a different path of rewarding participation. In place of gratitude, Thyme will have its point system, with virtual rewards for reaching point milestones.

[9] Here we can see an example of the services that Thyme hopes to incorporate into its functionality, without the price tag that is attached. The Apps that have been listed here are examples of how others have taken a single problem that they have had and have made a solution. Their underlying "job seeking" classification is something that thyme wants to access, but from a goodwill earning perspective.

By being an open platform with less restraints on what is being listed, there becomes a more inclusive community where people feel like they will have something to offer. From this list, the app TaskRabbit [10] provided the best outlines for how the finished product will act. The verification process is one that we particularly would like to undertake to provide security for those that need services within the application. However, this will require someone who will have to act as verifier, and as the app grows, so will the team of verifiers. This is not appealing as it does not align with the

accessibility of Thyme, as the money to pay for an administrative team will have to come from somewhere within the app to make itself sustainable. TaskRabbit's solution for this is an up-front fee to become a verified “tasker” something that can dissuade others from initially joining the app.

This comes back to Thyme's lack of currency. Having a points system, where participation rewards the members with points but does not require any points to be removed from someone else's account. To people, this creates less worry when joining in, and asking for help when they need it since they don't feel pressured to save and have good money practices. When people have a currency system, they feel like they need to save money, so are less likely to help in mundane situations. In discordances with this, people feel like they need to help more, before they feel confident enough to ask for help. Finally, having a money system that directly correlates to their community leaderboard standing would result in people capitalizing in some areas of sale, leaving some people out, or making them feel daunted.

3. Design

3.1. Design constraints

As a project taken on under the guidance of a supervisor and in accordance with a client's wishes, there are baseline requirements that were required to create the envisioned app. The following specifications were to be a part of the app, but after those, there was freedom of choice for the continuation.

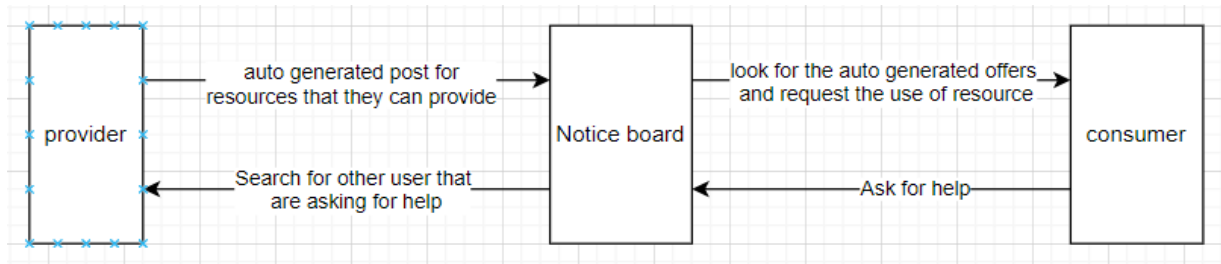
- The app was required to have a notice board feature. As the driving design of the app Thyme, the most work was put into having a noticeboard that people can efficiently navigate. It was important that it was organised in a tidy manner, and each post provided all the information that was required for a person to want to accept the job.
- Communication after resource/request acceptance was required to take place internally, to provide an all-in-one interface.
- Personal profiles. As with any app that has implementations of social communication, everyone required a profile that held onto the personal details as well as progress made in app. Beyond the basics (name, titles, bio) there were identifiers designed to help with person-to-person communications (star ratings, reviews, and leader board status).

1.1. Finalised requirements

The following requirements and persona examples were finalised for the MVP with discussion between supervisor, client, and developer.

Feature:	Example	Priority
Posting new requests and offered to the notice board.	Alice has recently taken up knitting, and she would like to know if there are others in the community that can lend her some stitching pattern books.	Must have
functionality to communicate with interested people who want to take up their offers	Alice has an active offer on the noticeboard. Bob has a need to use this resource and has selected to request the use of this resource. Alice can see that a new Messaging channel has opened for this offer and can now talk freely with Bob.	Must have
Open Communication between users	With ongoing jobs, and with tools that have been handed out, Bob wants to be able to communicate with those people he is sharing his	Must Have

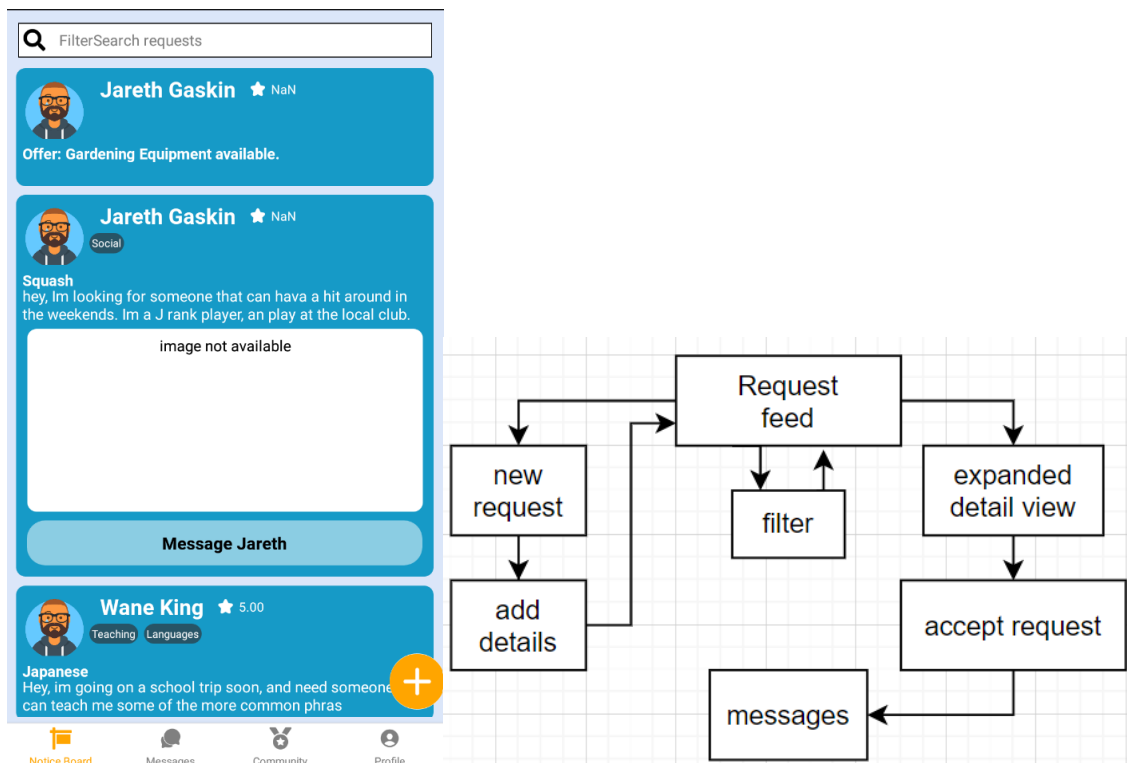
	resources with.	
track their 'community points'	Bob has been helping a lot of people in his community and would like to see how many community points he has currently earned.	Must have
Browse current needs that have been posted by community members and select jobs that you are able to help with. (Person with skills/resources to offer)	Alice has a camera; she is looking for people who have posted requests for photographers or people who simply need their photos taken.	Must Have
Browse current offers that have been posted by community members and select offers for skills/resources needed. (Person without skills/resources)	Bob has an overgrown garden; he is looking for any tools that other users have offered to help him with cleaning his yard. He can do this by scrolling through the offers, or he can search directly for gardening tools.	Must Have
Filter search result and general browsing for faster searching (using set categories, tags)	Bob and his friends would like to learn French; they are looking for individuals that can teach them some of the more common phrases. They open the search feature and clicks on 'Languages' category or tag with the label 'French'.	Must Have
Leaderboard	Bob has been hard at work offering his skills and resources to community members, in turn he has accrued a substantial number of community points. He wants a way to see how he ranks within the community against his family and friends and needs access to the global leaderboard of community member scores.	Should Have
post community events	Alice is organizing a barbecue fundraiser event and would like to advertise on the app to spread the word within the community. She created a new post in the community section.	Could Have
track / diary events that users are hosting.	Bob has created an event and needs an obvious way of seeing how people are interacting with the event. This can include the number of people going, and who can help.	Could Have
Reviewing users after a completed offer	Bob has used Alices camera and returned it in the same condition as he got it. He would like to leave a comment so that other people know that she is a lovely person.	Could Have
Reporting unethical users	Bob borrowed an electrical drill and returned it without a battery. The owner needs a way to let others know about David.	Could Have



3.2. Design Architecture

The highest level of the app tree consists of a tab-navigation system. This choice was made so that at any given time, independent of the depth that a user may have navigated on a given tree, they may switch to other tasks while keeping the state of the current page. When building the app, there was four key focuses that were divided into 4 main pages. As such, they will be discussed by page.

3.2.1. Requests page



The request page consists of a generic scrollable interface that takes up most of the screen. This scroll panel contains a list of posts that display users requests and offers. Aside from this, the request page offers a way to filter the feed and create a new request. this page has a total of two screen levels at maximum, meaning the depth that a user needs to navigate at most is through one button press to reach all aspects of the request feed. This is purposeful, making it far easier to learn for inexperienced users and making the program as intuitive as possible. For experience users, it makes the process of search or making new requests convenient.

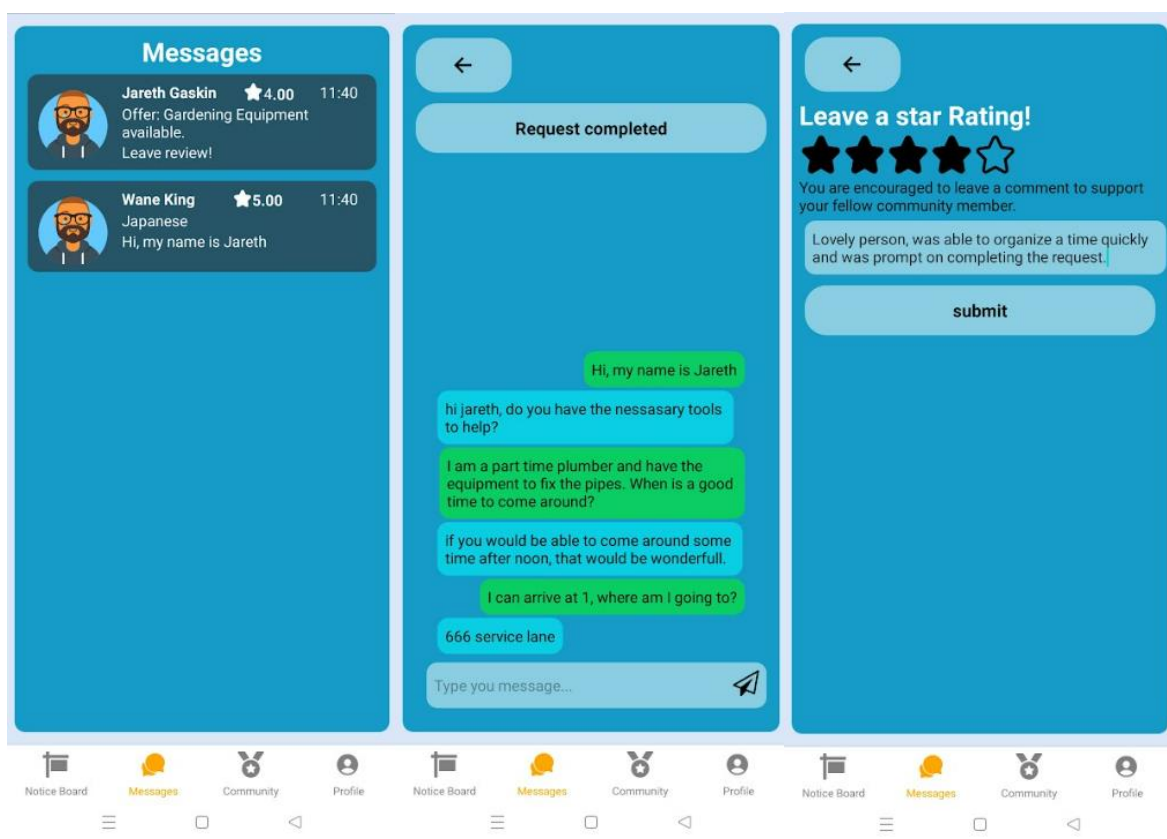
Example of resource trading can be found on many of the popular social media platforms. In the case of Facebook or Instagram, there exists pages where people sell or even give away products. Knowing this, the project to make an app where the core feature is based on something that is already being used in other media had to surpass the existing examples in some aspect. Failing this, the app would not be

considered by users over the alternatives that they already have in place and have a wider range of functionality.

With mobile design, there are decisions on how much information could also be hidden from initial glance, because with mobile design having an object that takes up too much space can result in lots of scrolling. With expanding communities, the large scale of information needed to be navigable. To provide this, a filtering feature was developed that made use of tags that were attached to each post. Further on, these tags will be referenced again when approaching achievements and expanding categories.

The final interface for a post is shown above in both expanded and retracted form. The expanded form gives the user permissions to accept the request and view any further details that may extend beyond the maximum word capacity of a retracted post. A large white space is also reserved to represent the ability to add photos in further iterations of the prototype. Choosing to have the posts as a collapsible component serves a dual purpose. It is a safeguard that prevents users from accidentally pressing on the accept buttons when scrolling through the feed. The second functionality of this is the reduction of content displayed on the screen to casual observers. When scrolling through the multitude of requests, users can quickly disregard posts by their tags (if they do not have the skills or resources to provide) and do not want to be scrolling past extended post descriptions.

3.2.2. Messaging page



After accepting a request, or in response to a request of your own being accepted, a new messaging channel is created on the messaging page. Once both users have accepted that the request has been completed, the channel will be closed, completing the cycle.

Since part of the ending of a complete cycle is giving feedback and a rating, the messaging page also handles the feedback giving process directly. After a chat has been close, it is replaced with a similar looking chat header that has details of the request completed and a typical star rating setup. Users of

the app are also encouraged to give feedback, as this helps to make the community have better communication, whether it is recognising good people for the help that they have provided or notifying others that there may have been problems with the request situation. As such, the rating and review process is a safety feature as much as it is about improving your personal score and rating.

Here there was freedom to choose the exact navigation paths and UI flow. There was only a general idea that it would look like a generic txt or messaging app where people could communicate while the job is active. This feature provides real time communications, having similar delays for message transaction as those found in other social media applications.

To complete an offer, there is a two-factor authentication required. The first person to believe that the request has been completed will press the request completed button. This notified the other person by way of informative text at the top of their screen, just above the button. From there, the other user will be able to accept that the job is complete, and they will both be transition to the review screen.

It is important to note, that these steps can be started from either user, regardless of if they accepted the request or created it. There is also no restriction on the time between when a proposed completion can be offered, and the corresponding acceptance must be made. If both parties accept, the review screen will be accessible. For asynchronous acceptance, the user that will be returning to the application to find their request completed will access this screen much the same way as the messenger, as seen in the first image.

Completing the review will present both users with their ratings, and add the points gain to their accounts. The users can view their reviews (posted and gain) through their profile page.

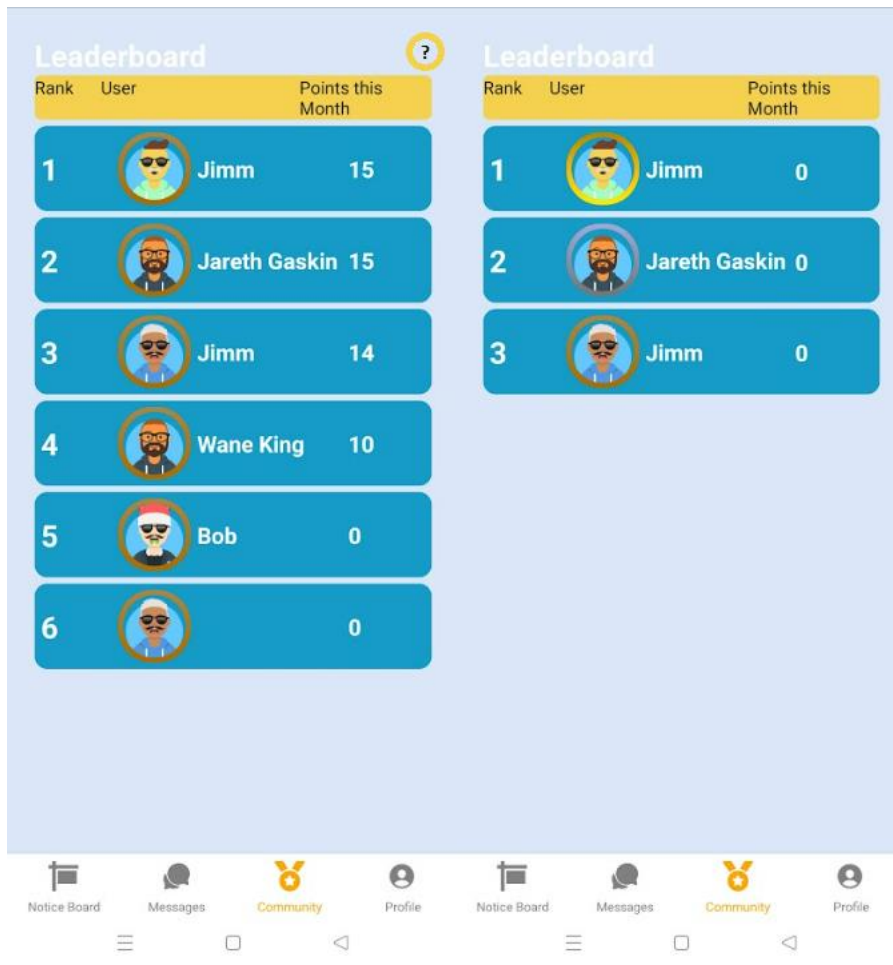
Finally, the communication channel is shut, removing it from a user's messaging page.

3.2.3. Community page

Established as the community page because it was initially where all on to multi-interactions would take place, it was devolved into a community leader board. Though it is a page that has a single active implementation, it remained separate with the possibility of re-attempting the community activities features.

The leader board's purpose is to encourage active participation in the community. With the position on the leader board being influence by not only the number of jobs being completed, but also the quality of the job (calculated by ratings given by the reviewer), people are encouraged to be courteous to each other and give an overall better service, rather than trying to monopolise quick and easy individual tasks.

Initially, the leader board was single tiered, with all members featuring on a single leader board. After user testing, it was deemed that adding different tiers would have a greater effect of creating continuous encouragement. It was noted that while it was required to have a periodic reset of the leader board (to make it possible for new members to the community to have a chance at featuring in the higher numbers rather than older members dominating) it would be discouraging to have users continuously starting from zero with no show of progress.



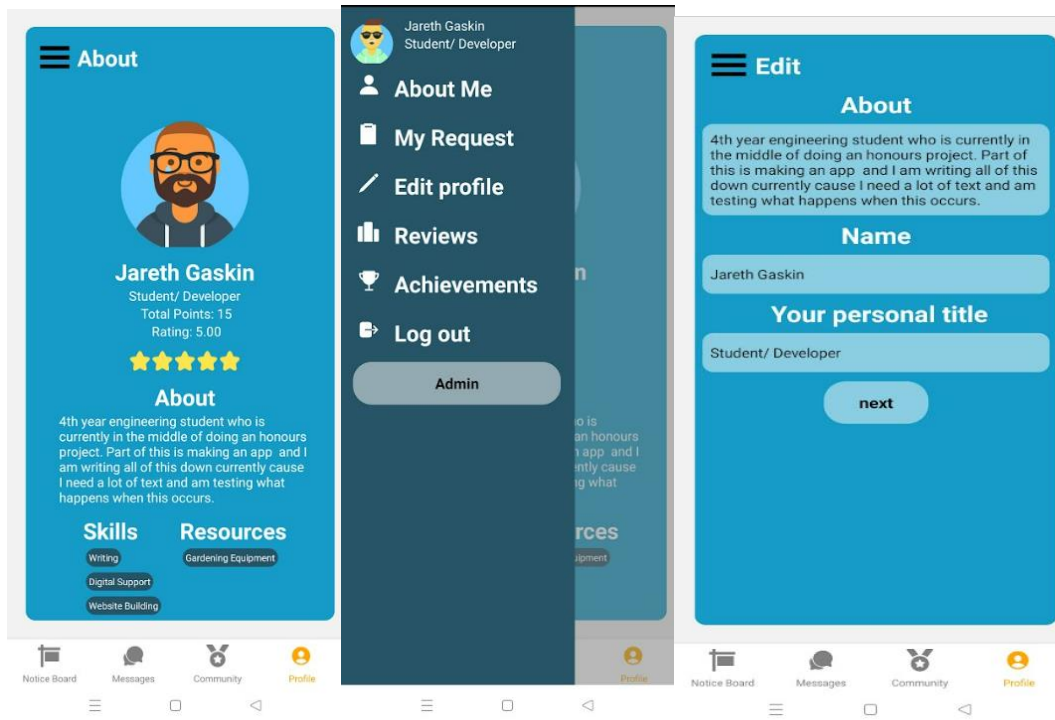
In this system, at the end of a cycle (monthly being an accepted length of time) the administration can do a soft reset that promotes those that exceed a certain boundary to the next tier. Those that have not been active will be dropped to the previous tier.

Being at a tier gives the user a corresponding decorative frame for the user profile image. This shows to the community, especially anyone that is interacting directly with your profile, that you are an active member of the community, and this is how well you are doing. Above-left gives a representation of the “bronze” tiers, which would be the tier that all users that were new to the app would start at. Though not on the first page, there is an information marker that lets the user know of the boundary they must reach before the reset to move up.

Above-right shows a representation of the three different tiers (bronze, silver, gold). It is important to note that with the tiered layout, they would not normally be shown together.

3.2.4. Profile page

As part of social networking, and individual page for self-presentation is required for keeping track of the individuals' activities and response. Here the user can also add small annotations that make people more or less likely to accept help from them.



The following is the information that is attached to the profile with freedom to be changed at a given moment:

- The visible name that will be presented to other users when interacting through the application
- Title/Descriptor a self-title that in future iterations would be a way for users that view your profile to help make decision on whether they think you are the most appropriate for the job.
- The about annotation also is an additional feature that has no direct use in the current iteration of the prototype. It is, however, inside this iteration as a place holder that will become influential in decision making in future aspect of safety/security when users will be given the opportunity to accept or deny responses to their offers.
- The profile avatar helps to add distinction between multiple chat headers in the messenger page, as well as requests in the main page. With current prototypes made from static images, the avatar images eventually should help to provide real world identification. In the consideration of speed, storage space, and user network data, avatars were chosen as the profile representative over images that users provide. This also has the added bonus of ensuring that avatars are people representative, rather than generic images that do not help other users (for example a picture of a cat).
- The user has access to a list of skills and resources they will be able to provide through the app. The choice to make a static list was to provide a measure of control over what can be offered in the app, as well as ensure correctness (spelling and avoid ambiguous labels) for proper communication. This does enforce certain rigidity to the list, disallowing people to add their own contributions. As such, the expansion of the app does heavily rely on the developer to make a large, inclusive list, or an administration that can continue to add to the list by accepting response directly from the users.

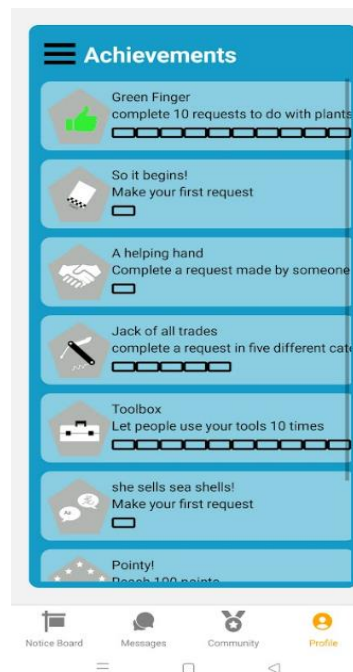
Other than the changeable information attached to the profile, there is a few bits of data that are influenced by the user's interactions but are not directly changeable. These are:

- The points system, which was initially thought to mirror the time-banking system. It would mimic a digital currency where points were given and received when processing requests.

However, this idea of losing points by asking for help was deemed to have too much of a negative impact against the encouraging nature of the application. Therefore, the decision was made to have a point-based system that only increased. At the point of decision, there was discussion on how this would reward the players, incentivising them to continue using the app to help others, rather than just the direct benefits received from others helping them. An approach was to have point milestones that rewarded the user with digital collectables or profile enhancing features. This idea finally gave way to the leader board approach, as the leader board provided all users with a more accurate measure of community members active participation than the rewards scheme would.

- The rating is a generic scale. A possible five-star rating can be awarded by any user of the app after a request is deemed complete. This rating will continue to be averaged out as more are added, giving the overall rating that is displayed to the public visibility. The rating is an instrument for informing communities of

Along with the profile data, there is one other feature that is accessed through the profile page, this is the achievements list. The achievements list is the second example of a gamified feature that promotes the use of the app's features. Additionally, it can serve as a guide for users to try new things, or to understand some of the more complex features of the app. From the sample displayed in the below image, single step achievements direct the user to make a request, and to complete it. This would prompt users to understand what the cycle of making a request is, and they can check back to verify that the process of making a request is complete. Alternatively, future implementation can cause it to appear after the action is complete, giving them immediate feedback.



3.3. Design alternatives

Given the specifications developed, there were multiple paths that could be taken to provide an app that focussed on promoting a particular requirement over another. Of the main requirements (promoting use, saving money, developing communities) the focus was put towards encouraging users

to explore the app, identifying encouraged use to be the most predominant feature. From this, the other two should naturally follow on. The more the app is used, the more communities will develop connections. Promoting diversity of requests through leader boards and achievements was the better way to spread the connections among more individuals.

Features that were considered to be added into the app include:

- Community events. In this feature, instead of accepting a request, which would remove it from the leader board, there was a standing request that multiple people could join. Here, events like social gathers could take place, but also things like teaching skills or pottery classes. Things that are a one-to-many interaction. This would have increased the community aspect, creating bonds at a far greater rate than one to one could offer. It also gives people the opportunity to meet people of similar interest. This feature was displaced in the development order in favour of producing a leader board. A community-based event has a different interaction with the points system, and at the time of conceiving, the app was at the stage where points were going to be a transferable asset, like time banking. Therefore, with points now being accumulated, but not spent, the leader board provided a better community interaction.
- An alternative initial design that was going to be used for the notice board was one that segregated the resource offers from the skill offers. This separation was first implemented as a way to reduce the number of posts that the user would need to scroll through, and as an initial filter it made sense. Once the automatic generation of resource offers were implemented, in that when a resource is listed on a profile as available it would be appearing on the notice board as a post, the separation became unneeded. Indeed, as the filtering of both resources and requests used the tag-based system, which was further reduced by making the tags searchable in the filter system, there was no productive reason to keep them separate.
- Finally, the biggest change to the initial brief was the shift away from a currency system that used spending and earning concepts. This change dramatically changed the priorities of the app away from security and token validation technologies and shifted to features that were more observable to the user of the app (UI/UX). The main arguments were oriented against the spending of tokens earned in the app, because this has been observed to result in hoarding of currency, and less likely to spend. One of the features that made Time-Banking systems was the fact that in many cases people did not have to earn a lot of Time-Tokens before they could start requesting people's help. This logic is carried over to this prototype app, demonstrated by only increasing the points earned, and users not having to spend points to ask for help.

4. Implementation

4.1. Framework choice

With most of the concept design already drawn up, the choice to use a framework was made based the need to jump right into it. Having a framework rather than working from scratch meant that the time between starting and having a visual representation built on a mobile device was reduced to a matter of hours rather than days of work.

The choice to use the React Framework, using react-native libraries, was made because it provided an enormous range of digital support, as well as the tools to deploy remotely, and make iterations quickly.

The React framework is constructed of components that work on both platforms. This convenience allowed the developer (who uses android) to work easily on the construction of the app, and the client (who used apple) to be able to view the app without having to make two different interfaces to show the same prototype.

React is a JavaScript library specifically designed to make interfaces. It incorporates HTML tag structure with JavaScript properties to create powerful state components. It uses an XML-like syntax called JSX, to construct components from various models and incorporate JS directly into the same file as the rest of the code.

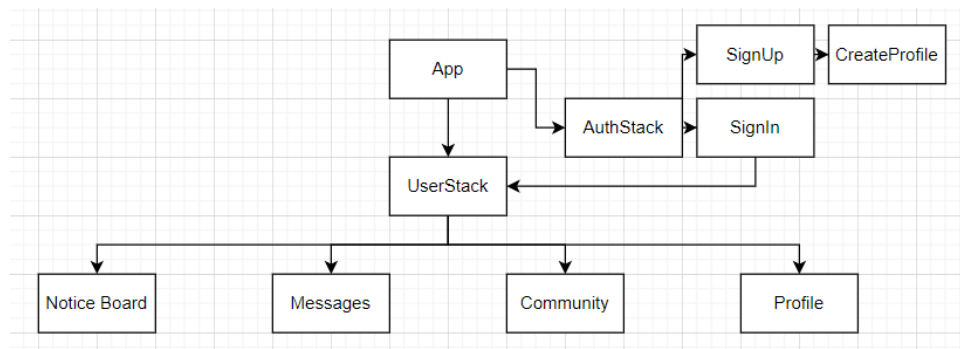
React directly interacts with the DOM (Document object module) that runs on the native device. Elements in React are plain objects and are cheap to create. React DOM matches the React elements for rendering, meaning that the application itself only needs a stable snapshot of the application object to render.

4.2. Architecture

The prototype application uses three types of navigations and uses functions to assign which objects are being represented on the screen, rather than using the react component library.

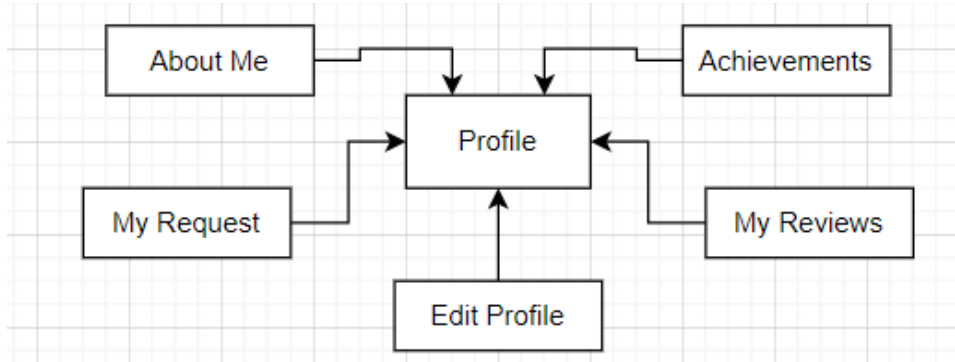
At the top level, on first entering the app, a stack navigation is used to control the flow of the app. This was chosen as each screen only requires a linear flow of back and forward. Each click moved to a new page, only navigating back by popping the screen of the stack. With no page leading to a loop, this navigation choice was appropriate.

With each user of the app requiring an account, the login process requires an email and password configuration to check for the existence of an account on the (Firebase) servers. If the user has previously logged in before with correct credentials, the path from app directly to User stack is achieved by using a locally persisted authentication key.



Subsequently, the app will reach a state where the user is logged in. this discards the user authentication stack, making that state no longer hold memory. From there, the rest of the app is accessed through combination of a tab navigator (the four main pages) and nested navigators;

In the cases of Noticeboard and messenger, there is again a new stack navigator, that allows them to retrieve and display new components on the screen.



The profile page slightly differences in its architecture, it makes use of the state logic to represent the component requested in the same place. Each time it retrieves the component, it reloads its state. This difference was made due to each of these components not relying on the context of any of the others. Having no direct influence on each other means that they do not need to be stored, ultimately saving space and power.

4.3. Platforms

React Expo is a product that has two key features that have been used consistently throughout the app.

- Expo CLI: a tool for creating projects, viewing logs, publishing, in general just helping with the development process. This tool provided an online server that hosted a copy of the published application, allowing it to be accessed from anywhere with the proper credentials.
- Expo Go: an app that can be found on the apple and android stores. This app connects to the Expo servers and allows apps to be run in a safe environment. For the developer, it means that an app can be uploaded remotely to a server and the user can test the latest update safely in a controlled manner. It is also important to note that it does not require app store credentials or complicated testing methods. This is vital as final users, and therefore the representative tester, will be from a wider range of age and technical skills.

4.4. Remote database

Both Firebase and Firestore was used in this project. Firestore was used to hold an individual snapshot of the user's profile because the profiles are read infrequently, being loaded at most only once when the user opens the app. This is optimal because Firestore tracks its usage per upload rather than by data, and in future iterations, the profiles data will become much larger. Firestore was not appropriate for features such as messenger or posts, where data as added and removed frequently, leading to limits being reached extremely fast after more people became a part of the network.

This problem was solved by firebase, the real time counterpart of firestore. This service is measured by data transfer rather than per use, making it a better match for the features that in some instances required little more than a switch of a Boolean to signal that a person thought a job was complete.

With the adding and removal of jobs happening quickly, the persistence of the database, as well as the synchronicity of the app became the hardest challenge to maintain when developing. Since users could be in any state of the app when new data is added, or because multiple people could theoretically

request the same data at the same time, there were measures that needed to be taken to ensure correctness.

Another challenge that was presented was the structure of the database. Though only a prototype, features were refined to use the smallest amount of data required. Each new line of data added to a message or post exponentially increased the amount of storage required and the amount of data transferred. This consideration extends to the environmental impacts and well as the immediate impacts to the user and their data consumption.

Google firebase provided several features to help with how the app connected to the network. One such feature is the Listener, which once attached, would update whenever there was a change to the database, allowing for synchronicity. This also meant that care had to be taken to prevent small updates from accidentally causing substantial amounts of data to be resent repeatedly, or for whole trees being downloaded because of a single change at a leaf node.

The response to this problem is data flattening, an aspect of data storage that needs to be considered. It is hard to find the balance between storage use, data used, and energy consumption. Using the example of the messaging page, it would be unwise to download all the messaging history when simply trying to get the most recent message for all contacts to display in a lobby. In this scenario it is obvious that there must be a separate collection to hold all the “last messages.” This is where the duplicate data comes from, as it would then either require a copy of the latest message to exist in two places, or for more computation power to be used in requesting all messages but then the last message separately. Understanding this, a separation of data must be made that is balanced, focussing reducing the impact that would harm the app the most. Ultimately, the worst of the impacts need to be moved away from the users.

4.5. UI/UX

Graphic choices of colour and layout were provided by the client, as they had previously drawn up mock-ups of the MVP. Assets provided by open-source website, or hand made in case of achievements, were incorporated into the app.

The exemplar images were designed using a wireframe soft wear to draw vector images of the main screens. These were used as reference when the screens were mimicked using react libraries. The colour schemes were changed multiple times to provide better visual stimulus when differentiating various aspects of the app. Once the first demo prototype was running, the colour scheme could be iterated quickly to provide a better result.

5. Evaluation

5.1. User study

A user study was conducted to appraise the prototype that was developed over the span of the yearlong project. The following describes the test undertaken, and the results found. It concludes with the description of how the results influenced the further development of the app, and the apps validation.

5.2. Conduct

Two evaluations were made during this study. The first was a baseline to determine the direction that the app should proceed in after the MVP had been met. These tests were design to get a broad range of user inputs, including their ideas on making the app better. This was done by gathering student from the university who willingly accepted being tested on.

The aim of the first round of testing was to attempt bug finding, have second opinions of the app, as well as observe the iterations that were made with the app. The external source of feedback from someone that did not know the premise of the constructed app provide vital insights into overlooked features.

The second round of testing had a more diverse range of testers. This was a practical choice because once thyme is launched, it would need to be usable by a diverse range of people. The second round of testing was undertaken to assess the added features, validating the correctness of the ideas. It also sampled from a greater age range to provide a more accurate test against the environment that the app was expected to be used in.

Before each test was run with a new person, a pre-determined set of data was added to the app. This data represented several other anonymous users, linked to generated personas, to make the app act as operational. The Testers were notified that this was the case, and that any response that they app gave them was pre generated to simulate actions, with the purpose of creating as immersive test as possible.

The users were tested one at a time without any communication or contact with any of the other testers in the sample. After each use of the prototype, the new users registered account that was created as part of the test was immediately removed, as well as any data that the tester had entered into the system, keeping the tests control and the tester protected from data leak.

For both sets of tests, at MVP and extended stage, the tester was asked to follow a series of tasks in sequential order. They were instructed to read the step, and then follow the instructions as they would think it should happen. In most cases, the user was in the presence of the developer, and so in situations that they were confused or stuck, they were able to directly explain the issue, which was noted down and the changed before the next user. This occurred more often in the first test, where examples of lack of information led to confusion, or buttons that took time to load not showing any visual input, making the user press multiple time and creating multiple calls to the servers leading to data duplication.

The testers were asked to give short answer replies to help with the development of the prototype and finished off the test with a user evaluation that gave a measurable result for graph comparisons.

5.2.1. data collections and ethics

Here we discuss the appropriate measures that were taken to ensure codes of ethics were followed and how people were approached.

As aforementioned, there were appropriate measures taken to ensure that the users' data was collected, handled, and deleted with proper measures taken towards privacy, but they were also made aware of such. For the first round of data collection and testing, students at the university were given the opportunity to test the app. While they were informed of its nature and given assurance that their private data would be handled only by the tester, it was semi-informal and required no document signing as they were under the universities ethics policies.

Upon attempting the second round of testing, the testers were provided with information sheets, as well as copies of consent forms and the option to receive records of the testing that took place, as well as any data recorded.

5.3. Results

The following results were gathered from the test through the use of a digital form, and then analysed using google sheets to produce the following charts.

5.3.1. Measurable results

Results from the usability scale and how those results influence the progress of the application. There were clear responses to the question that corresponded to ease of use. The system was well directed and after a few adjustments, the labels for buttons and icons became more intuitive. Text boxes that require input were also adapted repeatedly to have better placeholders that prompted responses. As a direct result of the first testers, this did not continue to be a problem that was picked up by the latter.

The trend in the software usability questions provided clear insight into how the app is appraise by the generations. While the majority of the questions that comprised of “how easy is it to use” were similar through both ages, the ones that referred to its importance as an app had a large divide, the older generation choosing the opposite to the younger.

It is a common conception that the younger generation are more open to adapting to new software as it is something that they have grown up with and are competent in exploring it. However, in this regard, many made the statement that they would prefer to use app that they are used to, mentioning Facebook as their common choice for a marketplace.

5.3.2. qualitative results

This section discusses the short answers responses and the verbal statements that were given in response to testing the app. These responses were either incorporated into the app if the new idea was appropriate, or disregards if they created conflicting ideas or where not to the taste of the app.

Again, it was quickly apparent that the age ranges, and therefore the experience with new technologies, directly influence the results.

For the younger generations (20 – 30) the mostly found that the prototype itself felt flat, lacking fancy animation and shading graphics that are commonly found in subtle ways inside newer applications. They gamified features where appreciated, but the concept was not well founded. When asked directly if they would use this app over how they currently would seek help, some said that they could not come up with a scenario that they would use the app. Others said that they would use something like this as it was to “small” and not as connected as the mainstream apps.

Another thing that was quickly pointed out by an inquisitive student was that it would not be hard to find a way to “farm points” in a manner that would quickly make the leader board aspect redundant as a comparative feature.

For the older generation (60+), the gamified logic was lost on them, though the achievement list was noted as “silly”. From there experience, they could see that this app would make a difference in some scenarios, but only provided that there were enough people using the app that someone could be available at the time.

These results provide a concerning gap between how the app was presented to encourage interest, and those that would understand the uses for this app.

5.3.3. Points of interest

Using three different devices, a trio of software engineers (inclusive of the developer) attempted to bug test some of the features of the app, and found that there were some key errors

- first, that a request could be picked up by too many people at the same time if they attempted to retrieve it at the same moment, this would result in a messaging chat window that would react to all three devices but would not require all three to accept the completion of the request.
- The points giving system did depend on a large amount of trust in the community that they were a part of. Since the points are given after the acceptance of the request, and there is no means for the developer to know if the job has been completed, there is nothing to stop members from repeatedly opening and closing requests.

5.4. Improvements made

After the results of the first testing, which was done at the stage where everything was first implemented, and the questioning was focussed on the idea of the app and the initial graphic setup. These were used to fix mistakes such as punctuation, input errors (buttons not working) and widespread use.

The additional features were then put forward into the final prototype, which resulted in the following additions.

5.4.1. Achievements list

As a further improvement on the gamified logic, it was suggested that an achievements list would provide a way to add interest to the app.

After analysing the potential of such a feature, it has considered to have the ability to incite the completions in some people. For others, it can serve to encourage people to try new aspects of the app, and act as instructions for those that do not understand how the app function. This is like a new Games tutorial, which provides the first couple of steps that are key to using a new app. From there, the user should have sufficient knowledge to begin exploring the app for themselves. Taking inspirations from games, the achievement sample that was added into the app was made to be funny, in a comedic way. Using phrases such as “green finger” as a title rather than saying “do something with gardening tools”

Though not implemented, an overlay like a notification bar or news feed bar could have been implemented to hold the next achievement or the achievement closest to being done. This little reminder would encourage people to take their last step toward doing a new task.

5.4.2. Tiered leader board

Along with the achievements, the tiered leader board was a concept presented after the minimum viable product was developed. The concept being born of gamified logic as well, based on representation in games such as ***** It has the effect of promoting competitive edges in those that actively use the app, as well as providing a goal for new beginners. The tiering also helps to make it so that people are not overwhelmed immediately by large scores of those that actively participate in the use of the app rather than being part of it passively.

6. Conclusions and Future Work

The following section summarises the results of the evaluation, the design implemented and how the requirements have been met. Along with this, the immediate continuation of the app's development is address and the direction that has been decided based on feedback form the user testing, as well as the features that the development of the app has made way for.

6.1. Summary

In this project, an application was developed with the goal to provide a community exchange platform. This application was to be designed to work on a mobile platform for ease of use, and not be restricted to a single operating system.

The outcome of such an app was to create a channel through which people could create strong connections in their local community. This also extended to new members to the area, helping them to gain connections quickly and add to the surrounding local economy with skills and resources that they were willing to provide.

The next objective was to provide a means to trade in goods and services with other users of the app. This was met by provide a clear navigable interface for advertising the resources that a user has to offer, as selected by them. It also gave an interface for those that needed help to make requests to the local community.

Finally, the app worked to promote the sharing of resources and skills over seeking new material or larger corporate businesses. This was address using gamified features that incentivised the use of the app. The app provided a means to use community members resources with no monetary cost, and a channel for keeping track of resources that have been shared. The addition of a community leader board and in-app achievements encouraged the continued use of the app by the users.

6.2. Future work

Though the UI leaves much to be expected, the functionality of the app is sound. Each system flow is completed enabling the user to completed requests and message each other as expected.

Should the app be continued past its current state, it would be expected that the usability of the app become more refined in its processing. Examples that immediately stand out is that of the is the handling of persistent state. The proper software patterns need to be applied for the asynchronous initiation of database listeners, and the persistence of localised data when the app is close or turned off.

Though mentioned briefly in the design work, the community page does have room for much improvement. The participation of one-to-many events is a natural lead on to the one to one, making it a feature that would increase the inclusion of the app, bring in more users. This would require momentous changes to the points and review system, changing the foundation of how the app provides status to different users.

Though decided against in this prototype, it may be reconsidered in the future about developing the points into a currency system. For many, it would feel more rewarding to know that their work is resolving into something that will remain usable in the community, even if the apps goals are to focus more on the community feel, and that sharing is promoted rather than paying to use things.

For the older generations, this feels more natural and negates the feelings that the app is “volunteering work”

The database that was used to construct the real time interactions has a hierarchy tree structure. It would take very little effort to add another tier to that hierarchy that represented groups (i.e., local towns or even suburb's) and had an addition variable in the profile accounts that linked people to the suburbs. This would allow people to create independent groups within the app and provide a more controlled situation. This way forward could also have community leaders that organised and maintained order within the app. A new generation of the app Thyme that proceeded in this direction would result in a more dynamic app, that can be adopted by closed community groups as well as large open ones.

7. Acknowledgments

I would like to thank the client Luke Ross for providing an exciting project to work on for my Honors Year. I would like to thank my supervisor, Jennifer Ferreira, for the support and organisation that has been presented throughout the year. Finally, I would like to thank the testers, who will remain anonymous, for their contributions to the study.

[1] Wu, Jie, Steven Si, and Haifeng Yan. “Reducing Poverty through the Shared Economy: Creating Inclusive Entrepreneurship Around Institutional Voids in China.” *Asian business & management* 21.2 (2020): 155–183. Web.

[2] Mushtaq, R. and Bruneau, C., 2022. Microfinance, financial inclusion and ICT: Implications for poverty and inequality.

[3] Martucci, B., 2022. *What Is the Sharing Economy - Example Companies, Definition, Pros & Cons.* [online] Moneycrashers.com. Available at: <<https://www.moneycrashers.com/sharing-economy/>> [Accessed 8 October 2022].

[4] Livingeconomies.nz. 2018. *Living Economies:: LETS*. [online] Available at: <<https://livingeconomies.nz/solutions/lets>> [Accessed 28 May 2022].

[5] Akram, W. and Kumar, R., 2017. A study on positive and negative effects of social media on society. *International Journal of Computer Sciences and Engineering*, 5(10), pp.351-354.].

[6] CBIZ Innkeepers Specialty Insurance. 2022. *11 Mitigation Tips for Your Bed & Breakfast.* [online] Available at: <<https://innkeepers.cbiz.com/blog/artmid/28542/articleid/93/11-mitigation-tips-for-your-bed-breakfast>> [Accessed 4 October 2022].

[7] Steve Huckle, Rituparna Bhattacharya, Martin White, Natalia Beloff, ,2016. “Internet of Things, Blockchain and Shared Economy Applications”. Available at: [online]<<https://www.sciencedirect.com/science/article/pii/S1877050916322190>> [Accessed 14 October 2022].

[8] Thomas, M., Ageyev, D., Hegel, P. and Ma, J., 2022. *Together Project*. [online] Togetherproject.nz. Available at: <<https://togetherproject.nz/index>> [Accessed 05 May 2022].

- [9] Dragusheva, V., 2021. *7 sharing economy apps to help you earn money in 2021*. [online] Karshare.com. Available at: <<https://karshare.com/blog/7-sharing-economy-apps-to-help-you-earn-money-in-2021>> [Accessed 26 May 2022].
- 10] TaskRabbit. n.d. *TaskRabbit connects you to safe and reliable help in your neighbourhood*. [online] Available at: <<https://www.taskrabbit.co.uk/become-a-tasker>> [Accessed 18 2022].