



Table of Contents

| | | |
|----------|-----------------------------------|-----------|
| 1 | INTRODUCTION..... | 5 |
| 1.1 | Project overview..... | 5 |
| 1.2 | Aims | 5 |
| 2 | BACKGROUND..... | 6 |
| 2.1 | Market Research..... | 6 |
| 2.1.1 | Upwork..... | 6 |
| 2.1.2 | Fiverr | 7 |
| 2.2 | Research Conclusions..... | 7 |
| 2.2 | Requirements | 7 |
| 2.2.1 | Minimum Viable requirements | 7 |
| 2.2.2 | Non-functional requirements | 8 |
| 3 | DEVELOPMENT PLAN..... | 9 |
| 3.1 | Agile methodologies | 9 |
| 3.2 | Control system (GitHub)..... | 9 |
| 3.3 | Language and Architecture..... | 10 |
| 4 | DESIGN | 11 |
| 4.1 | Lo-Fi design..... | 11 |
| 4.2 | Hi-Fi Designs | 11 |
| 4.3 | User Interface Design (UI) | 11 |
| 4.4 | Database Design | 12 |
| 4.5 | Firebase Design..... | 12 |
| 5 | IMPLEMENTATION | 12 |
| 5.1..... | Features | |
| | | 13 |
| 5.1.1 | Login/Signup..... | 13 |

| | |
|---|-----------|
| 5.1.2 Profile | 13 |
| 5.1.3 Reviews..... | 13 |
| 5.1.4 Location | 13 |
| 5.1.5 Search | 14 |
| 5.1.6 Chat System..... | 14 |
| 5.1.7 Post..... | 14 |
| 6 QUALITY ASSURANCE | 14 |
| 6.1 Manual Testing | 15 |
| 6.2 Unit Testing | 15 |
| 6.3 Back-end Testing..... | 15 |
| 6.4 User Interface (UI) testing | 15 |
| 6.5 User Acceptance testing (UAT) | 16 |
| 7 EVALUATION | 16 |
| 7.1 Challenges/Limitations | 17 |
| 8 CONCLUSION | 17 |
| 8.1 Project overview | 17 |
| 8.2 Final product | 18 |
| 9 FUTURE IMPLEMENTATION | 18 |
| 9.1 Deployment plan | 18 |
| 9.2 Map Feature | 19 |
| 9.3 Save Job Functionality Feature. | 19 |
| 9.4 Notifications | 19 |
| 9.5 Promoting Functionality..... | 19 |
| 9.6 Payment Processing..... | 19 |
| 9.7 Gamification Feature | 20 |
| 10 ACKNOWLEDGMENTS | 20 |

| | |
|-----------------------|----|
| 11 BIBLIOGRAPHY:..... | 20 |
|-----------------------|----|

Abstract

Finding quick and instant funds and finance as a university student can be daunting at times as student life can get unpredictable. That is why the vision of GoLocal is creating a service marketplace that connects both service providers and potential customers i.e., university students in a smooth and user-friendly matter. This application aims to ease the process of how service providers can earn extra cash and assist other users on finding a service provider to fulfil their home chores and needs. How this works is to connect services providers with workers using an API to discover jobs in the local proximity, users also have the option of increasing and decreasing the proximity aid in finding services.

This report presents the development and the evaluation of the GoLocal application. Throughout the report, will cover the resources and tools that helped us create to bring the app to life, this includes background research we did, the plan of development we used to implement the app, the various limitations we faced, the testing we did, and our analysis of the test results. We also explored potential future enhancements for the app.

1 Introduction

1.1 Project overview.

GoLocal is a gig economy marketplace, aimed at students, that allows users to post tasks (jobs) that they need someone else's help to complete. Other users can then see these posted jobs, either on a user-specific home page or by searching for jobs that they may have skills/specialise in. These jobs can then be applied to and completed for a previously agreed price. Once an application for a job is accepted, a chat will open between the user that posted the job and the user that applied to the job, they can then discuss and organise a time for the job to take place. Finally, they can mark the job as completed and the chat will close and allow both users to leave a review for the user that completed the job.

1.2 Aims

The story of GoLocal was inspired by our personal need and struggle to easily find source of income as well as finding a reliable high skilled professional around the area to fulfil some tasks. We aimed to create a community-driven platform that would support both professionals and task providers. Hence, the name “GoLocal”, it represents the concept of

supportive community promoting sense of sociality, accessibility and convenience. Based on our original vision, preliminary research, and the discussion with our supervisor we set the main plan for the application by prioritising the main functionalities.

The aim of this project is to connect service providers with potential customers quickly and efficiently. Specifically, GoLocal provides students the opportunity to earn money using their skills. It also aims to build a safe and trusted user community by enabling rating and review system.

2 Background

We identified a gap in the market which is a platform that caters speciality to the local community, allowing local engagement. We found that many existing gig economy platforms were centred on large scale jobs and services and lacking a community engagement.

For research and inspiration purposes we carried out some background research into other markets and existing solutions to see how GoLocal can be used to solve current problems and needs along with other systems and apps on the market.

2.1 Market Research

We started by conducting research on similar gig economy platforms. Our initial observation was that all of these platforms followed a two-sided marketplace model, with a client (who submits the job/task and offers a price) and workers (who complete the tasks). This model was common across all the platforms we analysed.

2.1.1 Upwork

Upwork is a “freelance” website for professionals. This means that each account must be approved, by submitting details/qualifications and experience you have before you can be approved. Only then can you apply to jobs. It also finds jobs recommended for the worker based on their skills. This is a key example of the sort of way our system could be designed, it really shows the needs of both the sides of the app (the clients and the workers) to work well together for success. Upwork also allows the client to see all the applications from workers that they have received for each job they post, this means that they can choose the best suited workers. See the Similar Systems 2 Document in the corpus for more in-depth analysis on Upwork.

2.1.2 Fiverr

Fiverr also works with clients and workers, but in the case of Fiverr, the workers go onto the platform and advertise their services/skills and the clients come along and pay the worker for a task they need to be done. While Fiverr is a very good example of a service marketplace, it is not the best platform for us to compare our idea to. This is as it focuses on the specialist advertising their server and the client coming to look for the specific service. See the Similar Systems 2 Document in the corpus for more details on Fiverr.

2.2 Research Conclusions

After conducting research on both Fiverr and Upwork, because they are both service marketplaces, they both follow similar concepts of what we want to do. GoLocal will be more similar to Upwork. This is because Upwork focuses on the client posting the job they need to be completed, in the case of GoLocal this is what we want to do as every job will be different. This also means that specialists will be able to find the jobs best suited to them, by being able to choose which jobs they can apply to. Although both these service marketplaces are similar, they are not in the same niche as GoLocal. This is as Upwork is more for digital freelance work and Fiverr is for small digital projects that people request, GoLocal is all about the in-person jobs. This means we will focus more on connecting people based on location. This means that there is a clear gap in the market that GoLocal can fill.

2.2 Requirements

2.2.1 Minimum Viable requirements

We created some user stories to see what features we would need to implement. From these user stories, as a group, we were able to come with a list of minimum viable requirements (MVPs) for GoLocal. Please refer to the “Project Requirements Plan” document in the corpus for more detail on the requirements.

- The users should be able to create accounts with details:
- The users should be able to login to the app to view the jobs available.
- Users must be able to post jobs to the app.
- Users must be able to view posted jobs.
- There must be a way for users to apply for jobs.

- Users that posted a job must be able to see, accept and reject the applications.
- Users must be able to see the applications they have sent.

Along with these MVPs, we also have some requirements that are not essential for GoLocal to be functional, but will be necessary to improve the quality of life for users:

- Reviews system – users able to leave reviews and view reviews.
- Filter jobs to make browsing them easier e.g., by date posted or distance away from the user searching.
- Edit profile – allows users to edit details they entered on signup, such as email or phone number.
- See total number of jobs a user has completed.
- Have a recommended jobs part on the home page.
- A chat system so that both parties involved in a job can communicate – this chat system should only open when a user has accepted another user application.
- Allow users to search for specific jobs.
- Only allow a user to view the address of a job once it has been accepted.
- The app to be compatible with IOS and android.

2.2.2 Non-functional requirements

Along with the functional requirements, that we can gain from the user stories, there are also some non-functional requirements. These will ensure that the app is easy to use and well designed.

i. Efficiency

It is important that the app runs well and fast. User's might not necessarily have time to wait for pages to load while navigating through the app, so we need to make it as quick as possible.

ii. Usability

The app must be easy to use. The everyday user should be able to navigate through the app and quickly gain an understanding of how to use the app. It should also be intuitive and a clear and concise language should be used.

iii. Privacy

The app must be secure as we are storing sensitive data for users, such as addresses, emails and phone numbers. It is crucial to ensure that the app's database is secure and protected.

3 Development plan

3.1 Agile methodologies

As a group, we decided to opt for Agile SCRUM methodology for the development of our project. This methodology focuses on breaking the project down into different stages, "iterations", where each stage is worked on in sprints, normally for 2-4 weeks, until the stage is completed. We discussed each "iteration" being a different main feature of the app.

Another key part of Agile methodology is the SCRUM meetings. We decided to have one or two SCRUM meetings a week, as a guideline, to keep up with the progress that everyone in the group was making with each iteration and we kept meeting reports of these meetings along with the weekly meetings with our supervisor.

We decided to use Trello, a project management tool, for the development of our project. We entered our requirements that we developed from our user stories. Then we could break the project down and split it up into different sprints to start the development of GoLocal. These requirements can be found in the "requirements" section above.

3.2 Control system (GitHub)

As a group, we chose to use Git version control to help us monitor and develop our project. Git allows users to store their code in a centralised repository, this repository can be split up into "branches" allowing each member of the group to work on their part of the code separately, with a reduced risk of human error (through parts of the code being accidentally deleted or changed while members of the group are working on the project simultaneously).

We needed to create a central repository, so that we could store our code using git. The best option here was to use Git's own repository: GitHub. GitHub, along with its desktop app

GitHub desktop, allowed us to store all of our code in a central location, with each group member pulling the latest version of the code before they start coding and subsequently pushing their updates when they finish. This is important to the version control of our project as it allows us to monitor every change that is made, with the change log as GitHub allows you to comment on every push that is made so you say which changes/bugs were made/fixed. Also, should something go wrong, we can revert the code to a previous version and try again.

3.3 Language and Architecture

There are many different frameworks and languages that can be used for the front end of our mobile development. As a group we all did our own research into these different frameworks. We discussed in a scrum meeting which framework we thought would be the best to use (see development document for full breakdown). Ultimately, we decided to use React Native to develop GoLocal. This is because it is a very well documented framework, with lots of libraries available for use and it works on a “write once, read anywhere” format. This means that we can write the code and run it on either IOS or android, which is perfect for what we need. Finally, it is new for everyone in the group, meaning we can learn lots of new skills while we create the app. See the Framework Research document in the corpus for a full breakdown on frameworks we looked into.

React native works well with some software called Expo. Expo is an open-source framework that allows apps to run natively on Android, iOS and the web. This allowed us to develop the app for all platforms at the same time.

For the database for this app, we decided to use a MySQL database. This was because as a group we are familiar with the MySQL so we can use our previous knowledge to create a good well-structured database.

For the chat system we used a Firebase Realtime database by Google. This allows us to let the chats update in real time and it is also compatible with React Native.

To link the Database and the front-end we decided on a PHP backend as it is compatible with both React Native and MySQL. This will allow us to easily send data back and forth from the app to the database.

4 Design

After narrowing down our minimum viable requirements (MVP) from our user stories creating our interface designs was very clear and straight forward. The approach we had toward the designs for the app was simple and minimalistic to enable a user friendly and easy to use feel.

4.1 Lo-Fi design

To start with, our Lo-fi designs were very simple and quick sketches made to a MS Word document. Even though our designs were very minimalistic at the start it still gave an idea what the sort of the features and components we were going to develop. Some of these designs included the signup and login screen as well as the homepage. In creating this Lo-Fi design, it enabled us to explore ideas and ways of how users will interact and navigate through the app. Regardless of the basic nature of our design we were able to gather some feedback which was later used in our final design and development of the app.

4.2 Hi-Fi Designs

Following up on our initial Lo-Fi designs we proceed to creating a very interactive and user-friendly Hi-Fi design for the app. These designs come about from feedback gathered from our Lo-Fi designs in conjunction with our minimum viable requirements (MVP). To make these designs more interactive for users we used "Canva," a graphic design platform that enabled us to create more detailed interface designs. By doing this, we were able to demonstration the layout of the app, how flows and navigates from one screen to another, as well as illustrating features that the lo-fi designs could not show. Ultimately, the Hi-Fi designs served as a blueprint for the final development and release of the app.

4.3 User Interface Design (UI)

We intended to create an application that is visually appealing, consistent, and easy to navigate. We tried to use the 60-30-10 rule to create a visual balance between elements on the screen and increase both accessibility and legibility.

After conducting thorough research to determine the best colour palette for the UI, we decided to go with a minimalist colour scheme that incorporates shades of black, white, and grey, along with a pop of yellow. This specific colour palate is a versatile choice that evokes different moods and tones. The colour yellow was used because it is typically associated with positivity and warmth that evoke positive emotions like friendliness, creativity, and sociability, which align with the overall message of the application. The colour black was

avoided, instead, dark grey hues were used when needed. Both white and grey are associated with simplicity to maintain a minimalist clean design.

4.4 Database Design

We created an Entity Relationship diagram to portray the MySQL portion of the database storage system. This help's accurately displays the fields, primary keys, foreign keys, and relationships we include in our MySQL database (Figure 1)

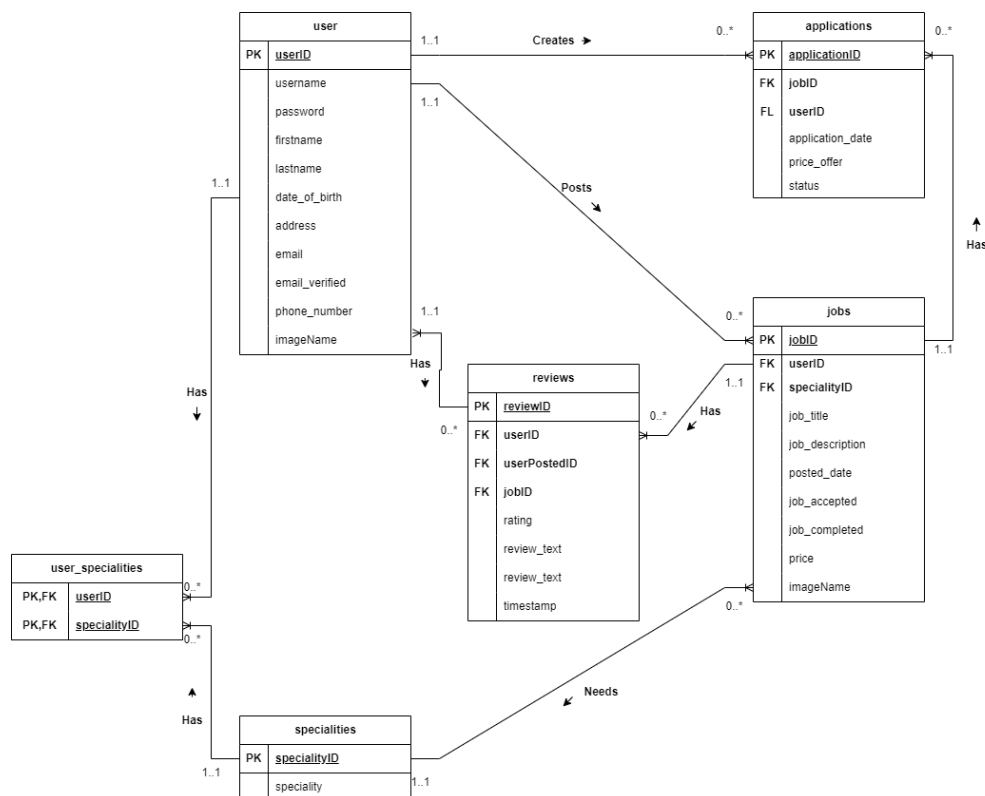


Figure 1:Final ER diagram

(All other drafts of the ER diagram can be seen in the Corpus.)

4.5 Firebase Design

We've made use of a Realtime database using Firebase to store our chat system data, which stores the data as JSON objects, with two main objects for users and chatrooms. An example structure of the firebase can be found in the Corpus.

5 Implementation

This project granted all group members the opportunity to widen their different technical and

non- technical skills. After comparative research on different frameworks, we decided to use react native cross-platform library to build our platform for different reasons i.e. rendering process, one-directional dataflow, and the large developer community.

5.1 Features

5.1.1 Login/Signup

The login page is the first page that you are greeted with once you open the app. It allows the user to enter their account details (if they have created one) or can redirect them to the “forgot password” or signup screens. The signup screen allows the user to enter their details and create an account. This they have filled in all the fields with valid data, then the data will be stored in our database and the account will be created.

5.1.2 Profile

The profile page displays the current logged in users’ information. It imports necessary dependencies and defines several state variables to store user profile information such as first name, last name, date of birth, address, email, phone number, image name, number of jobs completed, reviews, and score. Then it reads user ID data from *AsyncStorage*, fetches review data from an API. Finally, it renders the reviews written about the user by other users in a *FlatList*.

5.1.3 Reviews

Our reviews feature allows the user and other users to write and read for a particular job. To generate the app fetches reviews for the job from a server and displays them in a *FlatList* component. Users can also write their own review by providing a rating and review text. The user's ID is stored in local storage and used to identify the author of the review. The app uses various components from the React Native and React Native Paper libraries to display the reviews and input fields.

5.1.4 Location

On signup, we ask the user to enter their address. This address is then checked and validated using the google “*geocoding API*”, where it will only return if it has found a valid address at the input that we gave it. It will also return the address in a more standardized form which allows us to store all the addresses in the same form. The location is then used to sort jobs based on how close they are to the logged in user, by converting addresses to longitude and latitude coordinates and then comparing them.

5.1.5 Search

Our search system allows the user to enter a keyword or phrase of a job that they are looking to complete. This keyword is then used to query our database of jobs, where the keyword is checked for in the titles and descriptions of all the active jobs in our database. The results are returned in an array and displayed on the new “search” screen, making sure that jobs are not duplicated if they contain the same keyword more than once.

5.1.6 Chat System

The chat system serves as an area where users can interact with each other once an application for a job has been accepted. Users can discuss more specifics about the job in this space. It makes use of the Firebase Realtime Database and react native *useEffect* hooks, so the chat page between two users can work in real-time. It makes use of the GiftedChat library to view the messages in a ‘chatroom’ environment. The chat page also allows you to view the other users profile and reviews whenever you’d like, and it gives the host user (user who posted the job) more options, such as removing user from that job if it doesn’t work out and completing the job. Once the job has been completed, both users’ can leave a review for the other about their experience with both hosting/completing the job.

5.1.7 Post

The system allows users to post jobs that need completing. This includes adding a job title, description, and price, and uploading a photo of the job needing completion from the user’s camera roll. Users must also select a speciality category from a drop-down list that is specific for the job they are posting, so it can pop up on other users’ recommended jobs.

6 Quality assurance

We conducted Quality assurance as we employed a structured approach to ensure that GoLocal meets the requirements and user needs. This included both functional and non-functional requirements and user stories. These criteria are used to develop test cases and conduct test activities. We used a variety of testing techniques, including manual testing, unit testing, back-end testing, and user acceptance testing (UAT). After each iteration, we conducted code reviews where project members reviewed the code ensuring consistency and code quality to meet users’ requirements. Overall, this helped us address potential defects and catch problems in the early stages.

6.1 Manual Testing

Before every push to the groups GitHub, we tested our code manually to see if everything work how it should be. This included making sure all components load, variables are storing correctly and if the system is interacting with the database as expected. In addition to this, whenever we would pull new code and find a new error, we would note it down in an issue tracking document (Please see the Issue Tracking Document in the Corpus) and let the group know, so appropriate fixes could be done.

6.2 Unit Testing

We constructed unit tests using Jest for our programs. This included tests that check if the pages and the components within rendered properly, and for other functionalities within the page, such as text inputs resulting in the right responses. However, the unit tests were limited due to a multitude of reasons. No one in the group had every written unit tests before, which along with limited time, we found it difficult to write functional tests for all parts of our project. We also found it challenging to test for some external libraries we had imported. Please see the corpus for the Unit Testing documentation.

6.3 Back-end Testing

To test our backend PHP scripts, we went through each file and tested them individually. This consisted of giving the script test inputs, and monitoring what outputs it gives out and whether it makes appropriate changes within the database. Using this method, we could test very specifically for which inputs gave results that weren't expected. We had one file that was failing our tests, but when we tested it within the project, it was working as we expected, so we suspect there was an error in the file used for the tests. Please see the corpus for the Back-end Testing documentation.

6.4 User Interface (UI) testing

User interface testing (UI) is intended to focus on ensuring that the visual components work correctly to provide excellent user experience. The main objective of this test was to identify issues in the layout, visual design, and the overall appearance of the interface. The test cases covered fundamental aspects e.g., navigation, page content, and list contents.

This was conducted manually where after implementing the functionalities, the UI was employed and tested after each sprint. A UI testing was also deployed in final user acceptance testing.

6.5 User Acceptance testing (UAT)

This was the final stage of testing to validate that the final product meets the minimum viable requirements. This test was conducted on a number of students and some faculty members at the University of Kent in two phases.

The first phase was crowd testing followed by a questionnaire, which was conducted during the poster fair on several random users. Subsequently, the second phase was carried out on a number of students at the University of Kent. Both results are discussed in the User Acceptance Testing documentation in the Corpus.

Overall, it can be concluded that GoLocal meets the minimum viable requirements. Minor issues occurred; However, participants provided positive feedback on different aspects of the final product as shown in the appendices and the questionnaire included in the corpus.

7 Evaluation

Overall, the GoLocal app appears to be well-designed and functional. We succeeded to fulfil the minimum viable requirements where users can post and apply for a job. The home screen optimises user experience by personalising jobs based on speciality and location. The ability to sort and filter jobs by location is a valuable addition to the app as well. The chat system is well implemented allowing real-time engagement for users to discuss job details. The posting processes is straightforward and user friendly as well as applying and searching for jobs.

However, the app will benefit an improved system for managing jobs posted and applications with the ability to edit or delete the job. Another area of improvements is incorporating advanced search algorithms and implementing a recommendation system that generate suggested jobs based on user search history. Furthermore, the project would have also benefited with more testing for running the app on an IOS device, as most of the main testing was conducted on android devices/emulators. As of the final product, it runs very well on android devices, but its capabilities are more limited on IOS.

In addition, if we were to do the project again, we should have stuck more strictly to agile development. It was all our first times undertaking a large software development project like this, so at some points during the process we did more of a waterfall approach. If we stuck to agile more strictly, it would've led to a smoother development cycle, and saved us a lot more time towards the end of our project, were we did a lot of bug fixes and testing.

All in all, we believe that GoLocal has the potential to be a valuable platform for the community and a good source for students to deploy their skills and earn extra cash. With some minor improvements to both the functionality and the user experience, GoLocal can be widely used to find and complete local tasks.

7.1 Challenges/Limitations

One main challenge we faced as a group was learning about app development in general, more specifically, React Native, as no one in the group had any prior experience in this field. This being one of the reasons we decided on a project like this, also resulted in quite a slow start to the coding process. However, as the project went on, we became more comfortable with React Native and interacting with external libraries and API's. In addition to this, our backend was also quite challenging to begin with for the same reasons. Only one member of the group had a lot of experience with PHP, so writing our first initial scripts was difficult.

Furthermore, figuring out how to store out chat systems data proved to be quite difficult. Initially, we planned on using MySQL to store the chat system, however while drafting the database in an ER diagram, we couldn't figure out of to structure the tables properly, and that could be used easily in real-time. That's when we made the swap to using a Firebase Realtime Database, as it was a perfect solution for our problems.

Another challenge was Unit Testing for our project. Again, no one in the group had ever written unit test scripts before, and we left it quite late into the process. This resulted in a challenging time producing unit tests for all our code, as we found it very tricky testing certain libraries used, and relying more on other testing techniques, such as User Acceptance.

Moreover, a limitation the whole group faced while developing was dealing with emulators. As most of us didn't have android devices, we relied on an android emulator to test our app. This slowed down the development process, as everyone in the group had several experiences with having to fix bugs with the emulator before being able to continue coding.

8 Conclusion

8.1 Project overview

We aimed to create an application that implements a service marketplace in your local area. GoLocal achieves this by allowing all sorts of users to create accounts and view jobs based on many different factors relating to the jobs, the main factors being location and speciality. This

means that we have successfully fulfilled the initial project ideas and created an app that captures the concept we had in mind. GoLocal is also able to fill the gap in the market in the service marketplace industry, as not many competitors offer in person services, which makes it different from similar applications in the industry already.

8.2 Final product

At the start of your product, the initial idea was to create an application that fosters a service marketplace for connecting both service providers and potential customers, i.e., university students, in a smooth and user-friendly manner. Our final product has been able to achieve this to the best of its ability using the tools and resources available to us during the period of time given. As it stands now, Go Local is capable of meeting our target users' needs as they can create accounts via sign up, search, post, and apply for services; they can also view and write reviews once services have been completed. To top it off, the niche that distinguishes our app from others in the market is that our app enables users to find services posted by service providers in their local area based on the address they used to sign up, with the option of increasing and decreasing the proximity of search based on preference. Furthermore, you can have live conversations with other users on the app.

Given the chance to redo this project we believe a much different development approach will be taken. We would have fewer features that need a steep learning curve and take the time limitation into account. Overall, we are pleased with the project's outcomes as they represent a year of learning, mistakes, and converting those failures into lessons to accomplish a goal.

9 Future implementation

9.1 Deployment plan

Based on the positive feedback received during the UAT, we believe that there is great potential for GoLocal to be utilized through various channels in the future. e.g., Google Play and Apple Store. The distribution plan would target a specific niche i.e., university students.

Initially, we will conduct additional testing to ensure app quality. The testing will include functional testing and usability testing to check that the application is performing as expected. Furthermore, we will conduct further compatibility testing to ensure that the app works smoothly on all devices. Another testing to work on is performance testing, to ensure that GoLocal can handle heavy traffic and to avoid crashes under high usage. Once all testing has been completed, GoLocal will be published to Google play store, we will use a database

provider such as Firebase, after configuring the database we will generate a signed android package kit (APK) using android studio.

Additionally, a developer account will be created followed by creating the necessary metadata and the store listing as well as the graphical assets. An app store optimisation (ASO) will be executed to improve app visibility and increase the possibility to be discovered by optimising different elements. keywords, headings, and app description, this will ensure GoLocal remains visible overtime.

9.2 Map Feature

This feature will allow users to view tasks available within a specific area on the map. It can improve convenience and enhance user engagement by providing users with real-time information. This feature will increase the probability of a successful match.

9.3 Save Job Functionality Feature.

This feature would allow users to save the job and apply later that will improve user engagement and encourage users to spend more time on the app, increasing user retention.

9.4 Notifications

Another feature to implement in the next phase is for the app to push notifications. Users will get notifications from different scenarios e.g., chat messages, applications, and nearby tasks. Furthermore, an alert functionality can be implemented where users can customise the radius, category, keyword, and price for the job to be notified.

9.5 Promoting Functionality.

This will be a convenient feature that allows users to promote a job. For example, in a scenario where a job posted had no current application, the user can promote the job on the main screen for a fee for a certain period of time. This will increase job visibility, attracting more applicants. This can be customised to allow user to choose a limited duration for to promote the job.

9.6 Payment Processing

Currently, GoLocal does not support any payment method. Payments should be cash between the service provider and the user. A future improvement will be to implement a payment

gateway service such as PayPal that will also support credit card, debit card, and mobile payments.

This feature can allow users to pay for a service online and the service provider receives the payment later after the completion of the task. This will boost user experience offering security, convenience, flexibility, and efficiency.

9.7 Gamification Feature

This feature will encourage users to post more jobs or complete tasks, receiving positive feedback. It can optimise user engagement as user will be motivated to complete more tasks to unlock additional features.

10 Acknowledgments

We would like to extend our sincere gratitude to our project supervisor, Kemi Ademoye, for her invaluable guidance and support throughout the project. Kemi's insightful advice and feedback helped us to stay focused and motivated, and we are grateful for her dedication to our success.

We would also like to thank our fellow group members for their contributions to the project. Each member brought a unique perspective and set of skills, which made for a truly collaborative and enriching experience. We appreciate the time and effort that everyone put into this project, and we are proud of what we accomplished together.

Finally, we would like to acknowledge the participants who generously gave their time to complete our surveys and questionnaires. Your insights and feedback were essential to the success of our project, and we are grateful for your willingness to participate.

Overall, we are grateful to everyone who played a role in making this project a success. Thank you all for your support and contributions.

11 Bibliography:

- CareerFoundry. (2022). Color Theory And Color Palettes — A Complete Guide [2023]. [online] Available at: <https://careerfoundry.com/en/blog/ui-design/introduction-to-color-theory-and-color-palettes/#what-is-color-theory> [Accessed 2 Feb. 2023].
- Daļa, C., Zinātnes, P. and Humanities (2019). *DAUGAVPILS UNIVERSITĀTES 61. STARPTAUTISKĀS ZINĀTNISKĀS KONFERENCES PROCEEDINGS OF RAKSTU KRĀJUMS*

THE 61 DAUGAVPILS UNIVERSITĀTES 61. STARPTAUTISKĀS ZINĀTNISKĀS KONFERENCES RAKSTU KRĀJUMS PROCEEDINGS OF THE 61 st INTERNATIONAL SCIENTIFIC CONFERENCE OF DAUGAVPILS UNIVERSITY. [online] Available at: https://dukonference.lv/files/978-9984-14-901-1_61_konf_kraj_C_Hum%20zin.pdf#page=79.

- Perfecto by Perforce. (2021). *UI Testing: A Comprehensive Guide | Perfecto by Perforce*. [online] Available at: <https://www.perfecto.io/blog/ui-testing-comprehensive-guide#whatismobile> [Accessed 27 Mar. 2023].
- The Manifest (2018). *How to Publish an App on Google Play: A Step-by-Step Guide*. [online] Medium. Available at: https://medium.com/@the_manifest/how-to-publish-an-app-on-google-play-a-step-by-step-guide-80f9f533e370 [Accessed 20 Mar. 2023].
- Vijay (2023). *What is User Acceptance Testing (UAT): A Complete Guide*. [online] Software Testing Help. Available at: <https://www.softwaretestinghelp.com/what-is-user-acceptance-testing-uat/> [Accessed 7 Mar. 2023].
- Wu, W., 2018. React Native vs Flutter, Cross-platforms mobile application frameworks.