A picture containing text, clipart, vector graphics

Description automatically generated

|  |  |
| --- | --- |
| Project Report  Practical Project  Management & Professional  Development (ISYS20182)  *Team 1b*  Nottingham Trent University | Nicholas McCaig (N0787115)  Emerson Darwin (N0785398)  Michael Odejimi (N0794544)  Seb Price-Thomas (N0793768)  Bikram Gaire (N0784121)  9/5/2021 |

**Table of Contents**

[*Tables* 3](#_Toc71489299)

[Abstract 4](#_Toc71489300)

[Introduction/Project Background/Justification for the Idea 4](#_Toc71489301)

[1. Survey of Existing Solutions 5](#_Toc71489302)

[1.1 Research Into existing Solutions 5](#_Toc71489303)

[1.2 How our Application will differ from those currently on the market 7](#_Toc71489304)

[1.3. Project Management 7](#_Toc71489305)

[1.4 Team Members & Responsibilities 8](#_Toc71489306)

[1.5 Sources of information, resources required. 9](#_Toc71489307)

[2. New Ideas / Project Proposal 9](#_Toc71489308)

[2.1 Aims and Objectives 9](#_Toc71489309)

[2.2 Functional Requirements 10](#_Toc71489310)

[3.1 Why choose to use firebase? 10](#_Toc71489311)

[3.1.2 Firebase against alternatives 1](#_Toc71489312)

[3.1.2.1 Firebase Vs Parse 1](#_Toc71489313)

[3.1.2.2 Firebase vs Kuzzle 1](#_Toc71489314)

[3.1.2.3 Firebase vs Heroku 1](#_Toc71489315)

[3.1.3 Why Flutter over its competitors 1](#_Toc71489316)

[3.4 Risk Assessment 2](#_Toc71489317)

[5. Design and development 4](#_Toc71489318)

[Design methodology - Agile 4](#_Toc71489319)

[4.1 Concept Map 4](#_Toc71489320)

[5.2 Check Code 5](#_Toc71489321)

[5.2.1 Clocking in device. 5](#_Toc71489322)

[5.2.2 Check code display device. 5](#_Toc71489323)

[4.3.3 Firebase Implementation 6](#_Toc71489324)

[4.3.1 Firestore 6](#_Toc71489325)

[4.4 Flutter Implementation 7](#_Toc71489326)

[4.5 Data Flow Diagram 7](#_Toc71489327)

[4.6 Use Case Diagram 8](#_Toc71489328)

[4.7 UI design 9](#_Toc71489329)

[4.8.1 Color 9](#_Toc71489330)

[4.9 Timesheet Application 10](#_Toc71489331)

[6. Testing / Evaluation 10](#_Toc71489332)

[5.1. Test Objectives 10](#_Toc71489333)

[5.1.1. Functional Testing 10](#_Toc71489334)

[5.1.2. Software functional testing 10](#_Toc71489335)

[5.2. Database functional test 11](#_Toc71489336)

[5.2. Component Test 11](#_Toc71489337)

[5.2.1. User friendly 11](#_Toc71489338)

[5.2.3. Scanning the QR code 12](#_Toc71489339)

[5.2.4. Adding the staff 12](#_Toc71489340)

[5.2.5. Calculating the wages 12](#_Toc71489341)

[5.2.6. Easily navigation 12](#_Toc71489342)

[5.2.7. Adding the shift 12](#_Toc71489343)

[5.4. Testing against requirements/objectives 13](#_Toc71489344)

[5.4.2. Requirements 14](#_Toc71489345)

[7. Conclusion / Evaluation 14](#_Toc71489346)

[6.1. Lessons learned. 15](#_Toc71489347)

[6.2. Further work 15](#_Toc71489348)

[7. Professional, Social, ethical, and legal issues 16](#_Toc71489349)

[7.1 Legal Issues: 16](#_Toc71489350)

[7.2 Professionalism issues: 17](#_Toc71489351)

[7.3 Ethical Issues: 17](#_Toc71489352)

[6.4. Social Issues: 17](#_Toc71489353)

[References 18](#_Toc71489354)

[Appendix 19](#_Toc71489355)

[A. Full size concept map 19](#_Toc71489356)

[B. Full size use case diagram 20](#_Toc71489357)

[C. Full size user flow diagram 21](#_Toc71489358)

[D. Full size Firestore design diagram 22](#_Toc71489359)

[E. Trello board 23](#_Toc71489360)

[F. Meeting Agenda 23](#_Toc71489361)

[G Git log 31](#_Toc71489362)

[H. Software Testing 35](#_Toc71489363)

[H. Database Testing 38](#_Toc71489364)

[I. Report 41](#_Toc71489365)

[I.9.1. Introduction 41](#_Toc71489366)

[J.10. Questionnaire 43](#_Toc71489367)

[K – Use case Diagrams 44](#_Toc71489368)

[L – Trello boared agile development 49](#_Toc71489369)

#### Figures

[Figure 5.1 - Agile methodology used in this project (Muslihat, 2018) 4](https://myntuac.sharepoint.com/sites/course716389-Team1b/Shared%20Documents/Team%201b/Report.docx#_Toc71487078)

[Figure 5.2 - Concept Map 4](#_Toc71487079)

[Figure 4.5.3 - Example Check Code 5](https://myntuac.sharepoint.com/sites/course716389-Team1b/Shared%20Documents/Team%201b/Report.docx#_Toc71487080)

[Figure 4.5.4 - Check Code Flow Diagram 5](https://myntuac.sharepoint.com/sites/course716389-Team1b/Shared%20Documents/Team%201b/Report.docx#_Toc71487081)

[Figure 3.5.5 - Firestore design 7](https://myntuac.sharepoint.com/sites/course716389-Team1b/Shared%20Documents/Team%201b/Report.docx#_Toc71487082)

[Figure 4.5.6 - Example screen 8](https://myntuac.sharepoint.com/sites/course716389-Team1b/Shared%20Documents/Team%201b/Report.docx#_Toc71487083)

[Figure 4.5.7 – Data Flow Diagram 8](#_Toc71487084)

[Figure 4.5.8 – Use case diagram 9](#_Toc71487085)

[Figure 4.5.9 – UI user flow diagram 10](https://myntuac.sharepoint.com/sites/course716389-Team1b/Shared%20Documents/Team%201b/Report.docx#_Toc71487086)

[Figure 4.5.10 - Color palette 10](#_Toc71487087)

#### Tables

[Table 1 - Review of existing solutions 5](#_Toc71466886)

[Table 2 - Risk Assessment 3](#_Toc71466887)

[Table 3 - Software testing 36](#_Toc71466888)

# Abstract

This project is part of the personal project management module for second year students at Nottingham Trent University. The project is a mobile application using cloud services to store information about a company’s staff schedules, allowing the employees to clock in and out for their shifts while showing an employee’s schedule allowing a new hire to register for the application scanning a QR code on the management’s own device. The project utilizes the internet of things to display a code which can be used to verify the location of the member of staff’s location and prevent fraud.

The team completed extensive research to find a gap in existing products on the market to develop an application which prevents fraud, saves companies resources and time while continuing to respect the privacy of all employees. The COVID-19 pandemic exposed several flaws with existing products on the market and the team developed alternative solutions which avoids close contact and shared equipment helping reduce the spread of the novel coronavirus.

The development team used UML design to model the needs and meet the objectives of the application, using a NO-SQL database to store information and Firebase to manage the backend of their application while providing for future functionality.

The overall project was a success, the team was able to utilize cloud services to implement the minimum features in the application, careful planning, tool selection and development has created a fully functional application which can be expanded on in the future and is able to be adjusted to changing objectives from any potential client adjusting to their needs as needed.

The application has been designed with the BCS code of conduct in mind and all social, legal, and ethical issues were carefully considered when completing this project. The team was able to use new ideas and technology to avoid any potential issues and carefully considered the impact the application may have to those who use it.

# Introduction/Project Background/Justification for the Idea

The project is an app/webapp which will work as a digital timesheet for a business and its staff, the service will allow the business to assign shifts to users allow them to clock in and out on their phone or via an office computer. The system will have an authentication system, Geo-Fencing and 2 factor authentications.

This project is designed for small and medium sized businesses who want to use a cloud-based solution for staff hours tracking in more than one location such as a chain of retail locations.

Similar apps and websites include Clockify (COING Inc, 2020) and Deputy (Deputy, 2020). These solutions are low cost but may seem daunting to small and medium sized businesses, yet studies show when these businesses do not use automated services, they lose an average of 8% of payroll (GetPayroll.com, 2020).

Key benefits of the app include:

* Reduction in employee fraud and mistakes
* Geofencing to ensure the employee is at the correct location.
* Overtime tracking / Prevention
* Automatic break tracking
* Automatic Pay Calculation – Staff know exactly what they will be paid before the pay period.

The market for employee time tracking is large given that companies lose about $11 billion a year because of time theft (Djordjevic, 2019) while 48% of north American companies still use pen and paper-based systems, it is clear to see that businesses are still looking for low cost and simple solution which help reduces loss in a business while ensuring accurate pay to employees. By using new technologies, the group can create a solution which is simple and efficient.

Other products that offer similar services may use similar features to our idea, however our idea adds to this by using technologies such as geofencing and 2 factor authentication via a QR code system. We are also able to utilize cloud technology to allow anyone access their Rota as needed without having to contact the location to ask when they are on shift.

The cloud technology also allows users to be notified as soon as a proposed Rota is released using cloud functions and cloud messaging or when the Rota is amended. By having real time access to the data, the system will be able to monitor changes as they happen meaning a manager will not need to contact the effected staff members as issues occur, they can be contacted and approve changes instantly.

By utilizing the technologies available from firebase, we can incorporate these technologies into the cloud-based application with opportunities to use Machine learning to recognize text which could be used to import a physical schedule into the application in the event a manager chose to use a different application to create the Rota. We would be able to use the barcode scanning machine learning package to scan and create QR codes to be used as two factor authentication. We would also be able to incorporate face detection into a mobile device allowing contact free clocking in as a member of staff enters the building ensuring it is them alone clocking in, while being able to in a COVID secure way.

# Survey of Existing Solutions

## Research Into existing Solutions

After the team made a final decision on what solution we intended to make it became abundantly clear that while there were existing solutions for timesheets, none of them have seemed to fully grasp a seamless user interaction for both employee as well as employer. Many still requiring manual operation by either employee or employer to log hours and having clunky outdated UI’s. This is the gap in the market that we intend to enter by providing a product that would go above and beyond what currently exists within the market.

Table 1 - Review of existing solutions

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Application Name | Core Features | Strengths | Weaknesses | Sources |
| Paymo | * Live reports; Automatic real time generation of reports. * Web Timer; Built in functionality for browsers. * Desktop Widget; Functions in a desktop Widget. * Budget; Allows for goals to be set and automatically tracked. | * Clear UI design. * Functionality across devices. * Ability to continually monitor teams hours. * Ease of sharing reports with team and clients. | * Tracking of time is started and stopped by user; Could be abused. * No location used so users can run their time up regardless of where they are. * Does not appear to have levels of authentication e.g. 2 factor. | https://www.paymoapp.com/time-tracking/?gspk=dmlqYXlrdW1hcnNoaW5kZTYwMzc=&gsxid=TOz1cB2OAsfr |
| Montitask | * Automatic timesheets; Automatically tracks time from start to finish, also provides a manual option. * Calculate payouts; Creates summary reports for salaries and payouts. * Optional internet and app monitoring; Allows tracking of users through the day. | * Core functionality is well executed, Simplifying the time sheet process. * Only requires employees to download the app as managers can use the browser to view results. | * Employees could be unhappy with the apps invasive nature. * Lacks levels of security; no authentication. * Offline time entry can be abused for extra hours. * Report is not clear on what hours were worked, only how many. | https://www.monitask.com/en/Home/online-timesheets?utm\_source=softwaretestinghelp&utm\_medium=timesheet-apps&utm\_campaign=top-soft |
| TSheets | * Time tracking; Allows employees to clock in and out, take breaks and change job codes. * Scheduling; Allows for employees and managers to create and share schedules. * Reports; Automatically generates reports, including payroll plans and predicted time for jobs. | * Provides access across many devices and OSs. * The application is team oriented, provides many features to interact with other employees and share information. * Generates useful reports for managers. | * The time stamps work on a punch in and punch out system, reliant on trust. * Lack of authentication. | https://www.tsheets.com/uk |

## 1.2 How our Application will differ from those currently on the market

Our application will aim to fill the weaknesses that exist within current applications on the market, whilst utilizing their strengths. Through the combination of geo-fencing and two factor authentication we aim to deliver a reliable, and secure means to log hours for a time sheet, preventing any security concerns whilst also encouraging a greater amount of honesty by employees as their precise location will be logged along with the timestamp; for example, preventing them from adding an extra 5 minutes at the start and end of their shift whilst they still en route to work. Furthermore, with overtime and break tracking we can protect employees from being overworked and assist managers in ensuring they are compliant with current workplace regulations, protecting the company and the employee, allowing our application to serve as a safeguard. Lastly the tracking of time will be almost entirely automated, this will prevent the ability for employees to commit fraud, whilst also ensuring they are paid what they are owed, with live calculations of what they have currently made, and what their projected earnings will be.

By meeting these expectations, we will have created an application that fills a hole in the market, and create a more streamlined and efficient application, saving manpower and ultimately saving the company money more than any existing product currently on the market.

## 1.3 Project Management

Trello is being used to track, assign work, and monitor progress within the group while storing our code base on GitHub (the full git log is available in Appendix G and our other files on Microsoft teams.

The established ground rules are as follows:

* The group will meet weekly on a Friday during term time to discuss the project management and progress but will meet more often to work on technical aspects of the project.
* The group agrees to work to deadlines and inform others as soon as an issue arises.
* The group will update Trello with their progress.
* The group will reply to communication from other members within 48 hours or if unable due to illness or other issues informs the group.
* The group agrees to write a paragraph each week explaining what they have done or their reasons of being unable to.
* Work will be allocated via Trello regular comments on the assigned tasks can be used to inform others of their progress.
* Each Member must record an accurate record of the tasks done (via Trello), taking credit for work done by others will be reported to the supervisor.
* The group will follow a three-strike system when meeting deadlines and must inform the group before the work is due if they are unable to meet the deadline or they will be reported to the supervisor and will be recorded.
* Non-attendance to any meeting agreed by the group without informing the project manager beforehand will be considered a strike.

The Software developers will make weekly offline backups of the current codebase and documentation. The group will fairly allocate work so that no member is unfairly burdened by others which will be tracked on Trello while ensuring that each member plays to their strengths through active regular communication. If team members refuse to communicate or attend, decisions will be made on their behalf and they will not be allowed to change any decision made because of lack of communication.

## 1.4 Team Members & Responsibilities

Our group consists of 5 people which will consist of a Project Manager, Software Architect, two Software developers and a Software Tester. Although each team member will take a leading role in their retrospective areas of project all members will be responsible for all parts of the work as assigned while all members of the team will check each other’s work and offer advice and be available to help if a team member struggles.

Specific tasks are assigned to each member as below:

#### Project Manager - Nicholas McCaig

* Will set out the plan for the group including the necessary milestones.
* Create a Gantt chart.
* Administer the Trello with tasks as required, organize any meeting, as necessary.

#### Software Architect – Michael Odejimi, Seb Price- Thomas

* Will design and maintain the necessary diagrams with support of the group ensuring consistent quality and updating these, as necessary.
* Complete the presentation.
* Complete analysis of relevant social, legal and ethical issues.

#### Software developers – Nick McCaig, Emerson Darwin

* Will both maintain separate weekly offline backups of the codebase.
* Make clear requirements and objectives.
* Will take the lead in development of the application with help from all members.
* Will be responsible to recording the system demonstration.

#### Software Tester – Bikram Gaire

* Will maintain and scrutinize tests and code as required to ensure high quality code via a test plan.
* Hold regular meeting discussing any issues found by any members and layout a plan for resolving those issues.
* Will create the appropriate documentation to be used within the team.

All members will work on all aspects of the work but will take the leading role on their aspects of the work required as laid out via Trello (An example of the Trello board is included in appendix E) . All members will be assigned technical and administrative tasks to ensure fairness and record any meeting taking place with a summary of the topics discussed and those in attendance. Our full agenda record is included in Appendix F.

## 1.5 Sources of information, resources required.

The team will be able to use an emulator on their windows or mac device to test the application without the need an android device. The team will be able to login to firebase as administrators to amend and add to their work as needed in a safe test environment before security rules are introduced.

# New Ideas / Project Proposal

## 2.1 Aims and Objectives

The aim of this project is to create a simple timesheet app which will allow a business to accurately inform users of their shifts when they login and allow the users to clock in and out via an app or via an office computer.

The web/mobile application will:

1. Log the times of users as they arrive and leave work.
2. Confirm that the user is in a designated work location to log the times.
3. Ensure that employees are having their breaks and that they are appropriate lengths.
4. Track any overtime done by employees.
5. Show users their pay for logged hours.

Why the objectives are achievable:

1. The ability to log times of the users is not a complicated task, that can be achieved through access of a global time online and then saved to a database with each users unique identifier attached.
2. Geo-tracking can be accessed with the devices built in GPS and then used to log the location of the user, access can be denied if permissions for GPS is not given.
3. Hours can be tracked and measured, along with breaks being at set times alerted via the application.
4. Like breaks and the tracking of hours, any time worked past a set number of hours can be considered overtime by the application.
5. A simple equation with the amount of hours worked and their pay rate can be used to present the user with an accurate pay for their hours logged, tax can even be considered.

## 2.2 Functional Requirements

#### 2.2.1 Must haves

1. Cloud functionality.
2. Accurate and reliable access of the time to prevent fraud (cannot be local).
3. Ability for pay rates to be set and changed.
4. Secure authentication of users; Unique credentials, based of 2 factor authentication.
5. Check and store users locations when attempting to update their timestamps.
6. Accurate calculation of users pay based off logged hours.
7. A universal scheme that is accessible and useable by multiple companies; If necessary mutable.

#### 2.2.2 Like to have.

1. Extra functionality for managers, ability to view staff hours and check locations.
2. Multiple OS support.
3. Animations.
4. Easy on the eyes, good colour schemes and layouts.

#### 2.2.3 Non-functional requirements

1. Accessibility: all users including those with impediments should be able to use the application.
2. Performance: The app should be quick, and function without delay.
3. Reliability: The app should not have downtime.
4. Scalability: The apps functionality should not decrease as more users access it.

## 3.1 Why firebase?

Firebase is a platform designed for software development that is engineered as a “Backend-as-a-service" solution for mobile and web-based applications, which includes services tailored for managing, building, and testing applications. This essentially removes the requirement for us to manage the backend for our database as there are APIs provided which can be essentially “plugged” into the application. There are also a variety of other services provided which led to a plethora of advantages.

The services provided by firebase are split into three main areas: App-building, Testing and Maintaining, and Business Instruments.

App Building:

* Realtime Database
* Cloud Firestore
* Cloud functions
* Cloud storage
* Hosting
* Authentication

Testing and Maintaining:

* Test Lab
* Crashlytics
* Performance Monitoring
* App Distribution

Business instruments

* Google Analytics
* Predictions
* Cloud Messaging
* Remote Confignote
* Dynamic Links
* A/B testing
* In-app Messaging

One of the first and major advantages we considered about firebase is its ability to function with offline access, as a user works on the project, the files will be saved to a local cache which will then be synchronized once a connection is restored. This is almost essential for maintaining a workflow with currents conditions as course mates are spread over a wide distance with varying access to the internet. And ensures that work will not be lost.

Furthermore, the services listed above will greatly assist us in creating a superior product as well as saving a large amount of time for us as we will not have to worry about creating or managing some of the more complicated areas within or application allowing us to focus on and narrow our attention to other tasks, ultimately leading to a more polished and functional application.

Yet another advantage to using firebase is the accessible UI and ease of integration within firebase, it allows for users with varying levels of programming knowledge to work on the project, this ensures that all members within the group will feel they are able to contribute and work on the project as its overall complexity is reduced.

Lastly Firebase provides large amounts of support for the application with a dedicated YouTube channel releasing guides and information surrounding the service, as well as a product page containing guides for various features, this is once again a key consideration when working with varying skill levels as it ensures that even the least skilled programmer is able to always have support in place and continue work on the product.

While not a perfect product there are some weaknesses that also had to be considered with Firebase, firstly is that the design of Firebase leads to the DB being one large JSON file which limits many complex queries as its incredibly difficult to perform a complex query on a massive JSON file, another disadvantage would be that there is a lot less support for applications on IOS than there is for Android devices.

All things considered we decided that despite a few disadvantages, that the advantages would largely outweigh them, as they would not particularly affect the functionality or design of our application whereas the advantages would greatly assist in the production of the application, and for this reason we settled on Firebase.

## 3.1.2 Firebase against alternatives

### 3.1.2.1 Firebase Vs Parse

Parse is open-source and due to this there is a large community of developers that supports Parse. This

allows for many developers that have released plugins and extensions to further push the

capabilities of Parse to one's need. Furthermore, Firebase has a Vendor Lock-In whereas Parse does not

make it a sometimes-crucial aspect in someone's decision to choose one or the other. However, for

smaller apps Firebase is free allowing smaller apps to use it. Parse offers a choice between self-hosting on your own or someone else's servers or Parse's own dedicated servers giving users the freedom to go for another server host if they wish. Meanwhile Firebase is a cloud hosting platform meaning all firebase users use this cloud and can upgrade the speed depending on their plan. Finally, due to Firebase working with Google's Cloud Platform it is easy to use and set up as well as allowing real time testing of android apps.

### 3.1.2.2 Firebase vs Kuzzle

Kuzzle is also open source whereas Firebase is closed source making it possible that the community that

follows Kuzzle may help other users if they need it. Furthermore, with Kuzzle being open-source it

supports other Multiprotocol tools such as WebSockets and other tools. Kuzzle also provides some

unique features that Firebase doesn't have such as the ability to use a dedicated admin console which

provides admin control of the data, manage user rights and more. In comparison Firebase doesn't have

an admin console to boast. Furthermore, Kuzzle has a JavaScript SDK complete with documentation so

anyone can learn it. However, there isn't a large number of languages that are supported for Kuzzle.

Kuzzle and Firebase share the similarity that are both real-time meaning the data is constantly updated

for everyone using the app.

### 3.1.2.3 Firebase vs Heroku

Heroku focuses solely on the server side meaning developers using Heroku don't even need to think

about the server as Heroku is dedicated to being a fast, flexible, effective and reliable platform. This not

only reduces stress for developers but gives more time to the app that is integrating with Heroku to be

developed. Furthermore, Heroku supports a large number of languages making it very attractive to

developers of all kinds. However, the main reason people choose Heroku is just because of how easy to

deploy it is and how much time it saves, this can be crucial in any stage of development as it gives the

developers more time whereas Firebase isn't as easy and may require some time allocated to it for full

integration into the app and platform.

## 3.1.3 Why Flutter over its competitors

Flutter is a mobile UI framework that was developed by google to be open source, with the intent of aiding developers to build mobile applications. As it was designed with app design as its focal point, it brings about a substantial number of benefits when adopted for a project such as ours.

One of the most immediate benefits provided by flutter is a fast development, which can reduce working hours, this is especially prevalent when considering its ability to use the same code base for IOS and Android apps, preventing you having to do the same work twice for a different operating system.

Flutter also utilizes a layered architecture. Which allows for you to have full customization with a fast-rendering time as it claims to give control over every pixel on the screen, allowing for a customization experience that has “no limits”. This will help us to create a strong UI without facing too many problems due to the freedom of design that is provided.

A more unique feature called “hot reload” can also be taken advantage of as it allows you to make a change within your code and in real time see the results in the app preview without a need to recompile. This allows you to debug and change designs/ features with ease, as you can see the changes you are making without the need to compile every single time.

While flutter is a strong framework to use it does not mean it's perfect and has a few problems we may encounter such as large file sizes, which tend to be associated with flutter apps, or Dart (the main language of Flutter) being quite an immature language. However, they are not impactful enough to incentivize use of a different framework.

While there are some larger frameworks available such as React Native, or Xamarin which carry some similar advantages to flutter there are some takeaways which made them less appealing to use. To begin with React Native is still in beta and as such has quite a few compatibilities and debugging issues as well as a lack of custom modules. With these considerations behind React Native its likely use of it would have slowed us down more than it benefitted us. It was a similar situation with Xamarin as it is not a mobile friendly framework, that carries large development times. It is simply more advantageous for us to use Flutter over its competitors.

## 3.4 Risk Assessment

Table 2 - Risk Assessment

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Risk | Probability | Impact | Description | Contingency Plan |
| Fall behind schedule | 1 | 5 | Team members get behind | Call a meeting and find out what is holding people back |
| Unexpected feature needed | 1 | 5 | A feature unknown to the team appears | Call a meeting and integrate said feature into the plan. |
| Group conflict | 1 | 5 | The group begins conflicting in certain situations. | Call a meeting and find a resolution that makes everyone happy. |
| Improper training | 1 | 5 | Members may struggle with certain aspects of the project | Call a meeting and aid member in their work and their training |
| Improper resources | 1 | 5 | The resources needed may not available | Call a meeting and work together to find the resource |
| Improper communication in team | 1 | 3 | No communication between members | Call a meeting and plan more meetings |
| Requirements keep changing | 1 | 5 | Requirement changes | Call a meeting to make changes to compliment requirements |
| Project too complex | 2 | 5 | Project is too complex | Call a meeting and try overcoming challenge |
| Software loss | 1 | 5 | Losses such as software and memory | Go to backups and evaluate damages |
| Member loss | 1 | 5 | Losing a member of the team | Call an emergency meeting temporarily split work and find another member to pick up |
| Illness | 5 | 5 | Falling ill is likely in these unprecedented times | Call an emergency meeting temporarily split work until member recovers |
| Engagement | 2 | 5 | Lack of engagement in the project | Arrange a 1v1 to try and re-engage them with the project |
| Work balance | 2 | 1 | The balance between work and other | Arrange a 1v1 to try and fix the balance |
| User is unable to log in and out | 2 | 5 | The user is not able to log out and log in on the App | Check user side if no issue check code and fix. |
| Break tracker fails | 2 | 4 | The app fails to keep track of breaks taken | Check break tracker code and resolve issue |
| Overtime tracker fails | 2 | 4 | The app fails to keep track of overtime that has been taken | Check overtime tracker code and resolve issue |
| Fails to display logged hours | 1 | 5 | The app fails to display the correct number of hours that have been logged | Check logged hour code and resolve issue |
| Cloud fails | 2 | 5 | The app fails to connect to the cloud | Check with cloud service if no issue check code that uses cloud service. |
| Secure authentications leaked | 1 | 5 | Users login credentials are leaked or lost | Require all users to create a new password and re secure authentications |
| Lose track of time | 1 | 5 | The app fails to track time properly | Check clock code and resolve time issues. |
| Calculation of logged hours and pay is incorrect | 1 | 5 | The app displays incorrect hours and pay that user is owed | Check hour, pay logger code, and resolve issue. |
| User unable to navigate with the UI | 3 | 2 | User struggles to use the UI and is unable to navigate properly | Identify what part of the UI the user struggles with and identify if changes can be made to accommodate the user. |
| Unable to change pay rate | 1 | 4 | Employers are unable to change wages. | Check wage changer code and resolve issue |

# Design and development

## Design methodology - Agile

Agile software development (Kent Beck, 2001) was selected to design our application. The program development team used Trello to determine the features that needed to be completed (Included in Appendix L) The tasks were divided into small tasks and all members were required to update the status of the tasks they completed daily and discussed in daily ‘Scrum meetings’ to be discussed in the technical meetings.

Figure 5.1 - Agile methodology used in this project (Muslihat, 2018)

## 4.1 Concept Map

Diagram

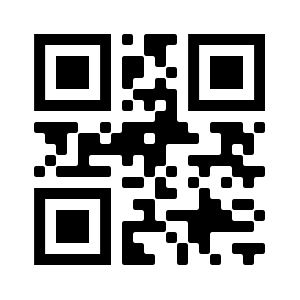
Description automatically generated

Figure 5.2 - Concept Map

(Full size in Appendix A)

Concept map was the first tasks that need to be undertaken in the primary stage- design and development stage. A concept map is a diagram either electronic or pen or paper, which links with different ideas of the project. By creating the concept map, it gives the developer the clear visualization of the plan. However, the concept map was broken into the 5 stages: Designing, Analysing, Implementation, Testing and Maintenance.

## 5.2 Check Code

‘Our app was based on the principle of allowing a member of staff to clock in/out for their shift while ensuring that member of staff was on the premises. To do this, we use a ‘Check code’.

A check code is the name given to a code which verifies the users’ presence in the workplace it is a randomly generated 5-digit number which can be typed or scanned by an employee and is required to clock in and out of a shift. An employee can check in at the stores “clocking in device” or via a managements mobile phone. This process is contact free which is important as users touching the same devices in workplaces has become a problem because of the COVID-19 pandemic (CDC, 2021)

Figure 5.3 - Example Check Code

## 5.2.1 Clocking in device.

A problem became clear during discussions with the research group, how would a user clock in when a manager was not available to show the QR code? The solution was to display a check Code on a device other than a phone. An E-ink display was selected as the best approach due to its low power consumption and ability to display information even during a power outage.

### Diagram Description automatically generated5.2.2 Check code display device.

A raspberry pi was used to develop the program to display the check code, because of its versatility and low power consumption. The flow diagram explains how the raspberry pi will function to generate the check code. The code here was developed in python.

#### 4.2.3 Setup and management of devices

A picture containing text, electronics, circuit

Description automatically generatedWhen setting up a new device a Key must be downloaded to the device and its settings configured to the correct store. Once device is responsible for generating the check code and uploading it to Firestore, other devices can be designated as ‘Secondary devices’ to be placed in other locations and will instead respond to an update to the database and show the latest code which will be synchronized. To change the device settings a SSH port is opened on the device to allow the user to remotely connect from another device without the need for a screen and keyboard, this was done to avoid the need for an IT professional to visit the location and instead remote into the network the device is connected to.

Figure 5.5 - Check Code Flow Diagram

A close-up of a motherboard

Description automatically generated with low confidence

Figure 5.4 - Check code device (with QR code)

Figure 5.6 - Check code device (without QR code)

## 4.3.3 Firebase Implementation

Firebase is a back end as a service tool which provides all the functionality needed for app development.

### 4.3.1 Firestore

Firestore is a NOSQL database (Google LLC, 2021) that uses a series of documents and collections which work like files and folders. Below is the design of the Firestore database.

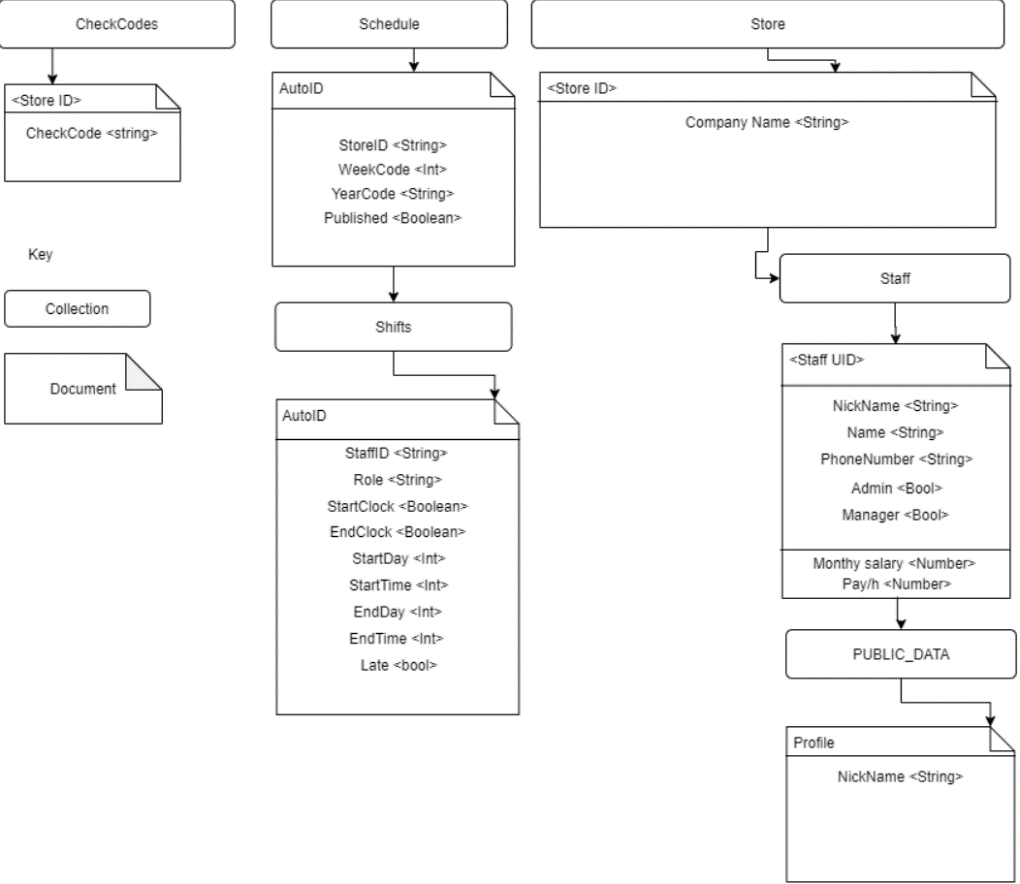
Firebase only allows a user to download a document, you cannot download a partial document therefore the PUBLIC DATA collection contains copied data from the STAFF collection, by copying the data into the PUBLIC\_DATA file in the collection we are able to keep private data secure. Data can be copied automatically using Firebase cloud functions.

Figure 3.5.7 - Firestore design

Full size in Appendix D

#### 4.3.2 Data - Stream or one time get?

We have two choices when retrieving data from Firestore, we could setup a stream builder with flutter which will automatically update as data is updated on the database, or we can complete a onetime get when the user loads the application. We chose to use a onetime get rather than a stream when retrieving data, this is due to the nature of our data, it is unlikely to update often (Only when a manager adds or modifies a shift) and therefore the more expensive process of streaming data is unnecessary.

## 4.4 Flutter Implementation

Flutter has been used for the front end of our application following material design guidelines laid out in the material design documentation (Google, 2020). Flutter uses dart, an open-source programming language supported by Google, it is optimized for UI creation, includes tools like hot reloading and is fast on all platforms and can be compiled into JavaScript for the web (Dart, 2021).

Figure 3.5 shows an example of how the application may look on an android device, we follow the UI design in figure 3.8 to ensure that our application has all the required functionality.

## 4.5 Data Flow Diagram

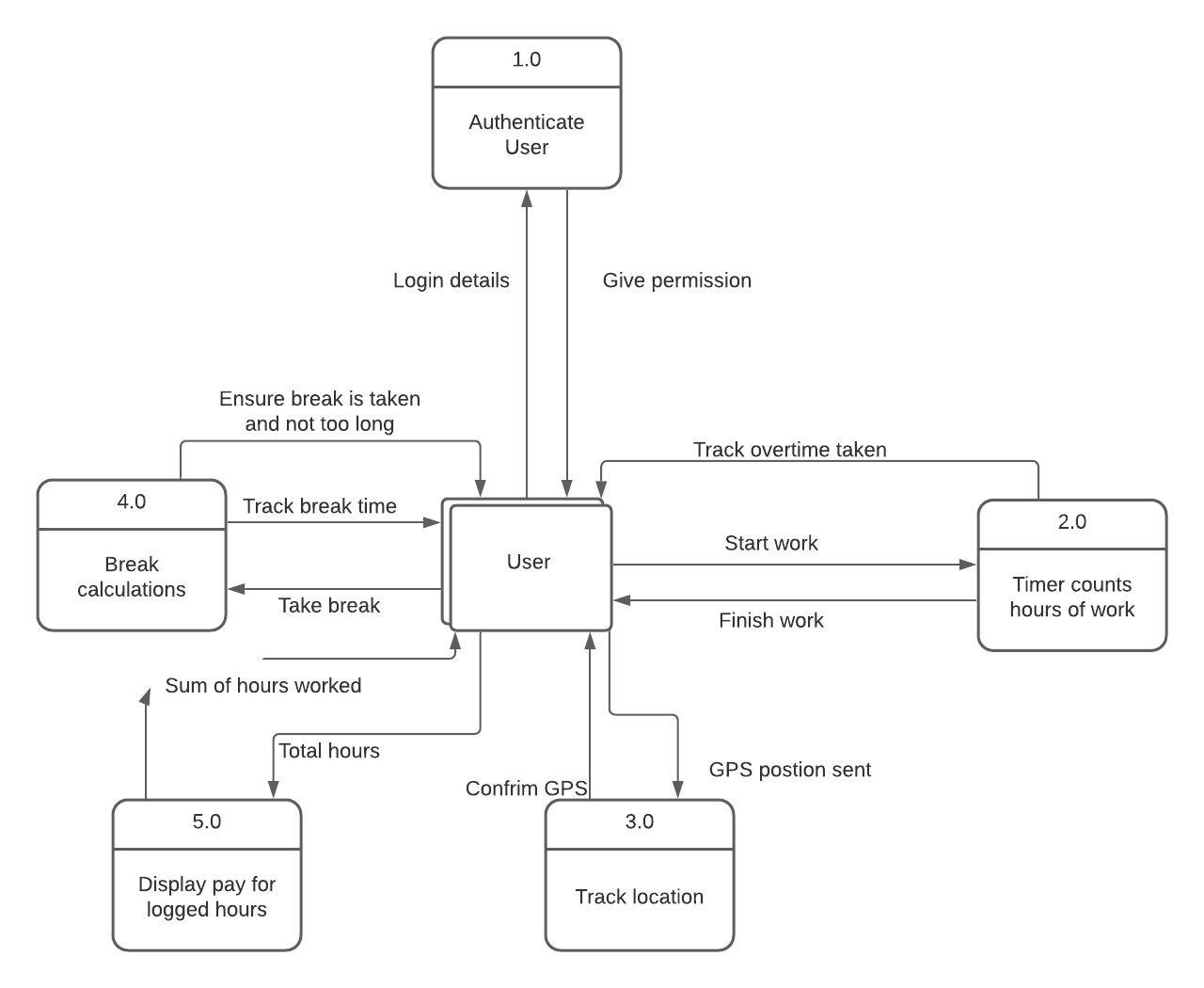


Figure 4.5.8 - Example screen

Figure 4.5.9 – Data Flow Diagram

## 4.6 Use Case Diagram

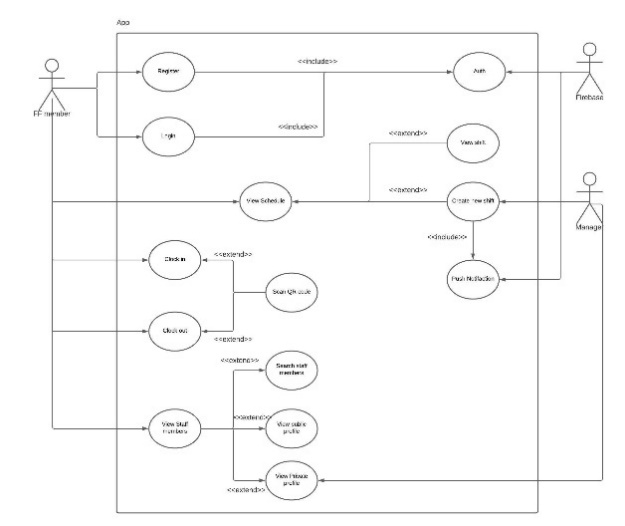


Figure 4.5.10 – Use case diagram

Full diagram available in Appendix B

The use case diagram demonstrates the uses of our application and the goals it needs to complete to meet the objectives. There are detailed use case descriptions which are included in Appendix K.

## 4.7 UI design

Figure 4.5.11 – UI user flow diagram

Full Sized Diagram is included in Appendix C

The UI folow diagram is used to assist in the design of the front end of the application, by designing the layout before hand we were able to understand the reqirements for each page and ensure all features are included.

### 4.8.1 Colour palette

Chart, bubble chart

Description automatically generatedChart, treemap chart

Description automatically generated

Figure 4.5.12 - Colour palette

The colour palette was selected based on the design from the logo a simple blue colour was selected because it is easy to read text on based on the opacity scores (Google, 2021). Flutter allows for universal colours to be applied using an xml style file which was included from the material design tool.

## 4.9 Timesheet Application

Figure 4.10 – Timesheet Application

# Testing / Evaluation

## 5.1. Test Objectives

When the project passes its completion stage, it needs to be tested to ensure the application work as the requirement of the user. However, the testing phase is tested to ensure it works smoothly as per the user requirements. The testing was done into the four parts: functional testing, component test, usability testing, user experience testing and testing against the requirements and objectives.

### 5.1.1. Functional Testing

This testing stage involves making the application passes the functioned to be worked as expected. Therefore, the test was done into the two components of the application: the software and the database. The full test result is shown in the Appendix H.

### 5.1.2. Software functional testing

The first test was done for the software. 26 functional tests were carried out for the software, ensuring everything works and was free from the bugs. Out of 21, 5 tests were not achieved. However, these tests were rectifying at the end. However, the details test was done in order to make the software run smooth without any lags and bugs.

The first test that was failed was the loading bar at the screen to show the user that the application is in the process of opening. This is the visualization effect as it shows the user that the application is in the process of opening. Thus, this was failed due to the complexity of coding. Nevertheless, this was also removed after the questionnaire result feedback was achieved as many users wanted to open the application as faster as they could. Having the opening bar at the bottom of the screen would delay few second in the opening of the application.

The next test that was failed was the absence of the back button. As the user have to move from one screen to another and the back button is necessity. Before the back button, the user have to physically closed the application in order to go back. Having the back button now, the accessibility is very simple. User can flick into the different screen without closing the application.

The next test that was failed was not able to edit the rota from the application. As this should be included because the rota should be flexible and may need to be edited in the future. For example, the admin may need to add some shift for the person A. Absence of this function will not le the admin to add the shift into their names.

The other and final test that was failed was the search button. The application did not include the search button, but this functionality was not added in the software as there is minimum requirement for this. The application is very simplicity and user interface. Therefore, it is also very easy to navigate from one screen to another, so this functionality was not added.

## 5.2. Database functional test

The final part of the test was testing the database. The responsible of the database is to add the shift and details of the staff. The software pulls the information from the database and present to the screen. 15 tests were tested for the database of which one failed. However, the failed test was fixed.

The test that was failed regarding the inputting the username. The datatype for the application was Boolean, which was not suitable for person’s name. This issue was fixed by changing the data type. Once the data type was changed into the string, the test again was carried out to ensure the error was gone.

## 5.2. Component Test

### 5.2.1. User friendly

As we know the application should be user friendly to catch the eye of the customers. If the application is not user friendly it would be very hard to navigate around. To ensure it is user friendly the application was given to different people to use the functionality. The test was successful as the application was user friendly.   
  
5.2.2. Professional look

However, with the achievement of user friendly, the application should also look very professional. Keeping the font right size, right color and clear picture the application would look very professional. Similarly, the search box is also the right size for the application.

### 5.2.3. Scanning the QR code

This feature is very useful in this time. It will allow user to scan the barcode to log in. This feature will enhance the touch free facilities as it is very need nowadays. Test was made to see if the application really carries this functionality. Not only this is more hygienic, it also eliminates the possibility human errors. Similarly, it is also more reliable and faster than entering data by hand.

5.2.4. Adding the staff  
 This system on the application will allows manager to add the new staff in the application. There is no limitation on adding the staff on the application. Firstly, the detail of the user is entered. Secondly, the QR code is scanned and lastly the confirm screen appears to confirm to add into the system.

### 5.2.5. Calculating the wages

This application will be able to calculate the wages of the hour worked by the user. However, the average per hour should be stated by the admin. Taking account of the data provide by the admin, the software calculate the hour worked.

### 5.2.6. Easily navigation

As we are aware the software should include the navigation. However, the navigation should be simple and intuitive. Having the good navigation between the screen will enhance the experience of the users. Unfortunately, not every software is designed as it should be. However, in this application user can navigate between different screen with the help of the back button. Designing the navigation without question is the most important aspect of the software.

5.2.7. Adding the shift  
This system on the application will allows manager to add the new shift in the application. There is no limitation on adding the staff on the application. Firstly, the detail of the user is entered. With the days to be worked and the time to be worked. When the admin confirms the system. The data is recorded into the database. However, the user can QR code to access their respective shift added by the admin.   
  
5.3. Usability TestThis stage is to test to work out application when used by the customer. To ensure the test was reliable the test was done among the target audience. However, the application was tested by the participate and was asked related question. Likewise, the questionnaire was presented into the graphs and data is shown in Appendix I.

The result from the questionnaire was generally positive with more people pleasing with the software usability. The first two question was related to the personal details of the participation. For both questions, the participation answered the details stating their age and gender. The third question was followed with the attractiveness of the application. The participants were pleased with the application. The feedback was very positive. They stated the application was easy to navigate, simple and easy to use. However, there were one user who stated the layout could be improved in a certain degree to enhance the usability, but he was please with the icon, colour and fonts.

The fourth question was the degree of the attractiveness. The participation were asked to rate the attractiveness from 1 to 10. 1 being the lowest and 10 being the highest. Generally, the audience rated 8. This feedback is important because it shows the developer that they were able to accomplish the satisfaction of the user.

The fifth question asked to participant was the navigation. Like previously the response was positive. However, one of them asked to add the logout option as this was not included into the application.

The sixth question asked was about the software recommendation. The question was added to known if the participants would recommend this software to other people. The outcome result was very pleased. As this software contain QR code and different functionality. There were very pleased in the current situation as the user do not have to physically use the card in order to sign in. They can simply use the mobile application to log in and log out.

The last question was open question for the audience for the improvement of the application. All the participants gave the suggestion for polishing the application. The most requested improvement was to add some picture on the home page as it is very dull. The other suggestion was to implicate the background picture to the app. Similarly, the eye catching home page was suggested by one of the participants.

## 5.4. Testing against requirements/objectives

The final testing stage involved the software against the requirements and the objective of the product. The testing was done in order to meet the aim and objectives for building the application.   
  
5.4.1. Objectives

The first objective for building the application was let the user login. However, this objective was made of the two elements, database, and software. Therefore, the application meets the objects as the data can be inputted and users was able to log in the application.

The next objective was the interface created for the user to view their respective timesheet. The application contain a screen which shows the timesheet of the respective users. The NOSQL service is used as this provides the user the core functionality of managing the backend of the databases. Thus, the objective is achieved.

The third objective was the admin able to create the shift for the staff. With the help of Cloud Firestore, the admin could make the shift for the staff. These data were recorded into the firestorm and the data was pulled by the software in need. However, it is cloud database. Therefore, the objective was meet.

The final objective that was set in the analysis phase was to calculation of the wages of the users. Thus, this objective was made in order to get the wages details of the users. The application shows the wages of the user at the bottom of the screen. The wages is calculated into the pound. Because of this, the developed product meets the objective.

## 5.4.2. Requirements

The first requirements of the analysis phase was that; the software will only allow the authentic person. The genuine person are only able to login in the application. Thus, this will be achieved as the QR code is stored into the database. The correct QR entered will only let the software access the functionality.

The second requirement was the application will send the data to Cloud Firestore. The data is store into the Firestore. This will allow the application to communicate with database in quicker way. Thus, meeting the requirement.

The next requirement wat that; the insert the data into the database. This is the important feature as the software will be reading the data stored into the database as it have to present it to the user. The software will query for the timetable and insert the data into the database, thus meeting the requirement.

The next requirement was to present to the users via the screen. The software need the QR code before accessing the functionality. The result is displayed on the graphical and textual form, thus meeting the requirement.

The next requirement was application able to log in with the social media or with personal email account. This requirement was met by the software which display the timesheet for the users.

The final requirement was that; the application will implement the gamification features that is good textures and display. The purpose of this application was to display the timesheet for the user. Similarly, this was also developed for the alternative traditional way of recording the data. Thus, having the textures will increase the engagement of the user to the application.

# Conclusion / Evaluation

Our project was overall a success we created a fully functioning application that met the ‘Must’ objectives set by the group, although we had problems with engagement most team members completed their work as discussed within an acceptable time frame.

We were particularly proud of our groups ability to develop the presentation by assigning two group members to the task they were able to police themselves and complete the required work as appropriate without the need of intervention of the group leader. By assigning a member to lead the presentation development and delivery as well as a secondary member to assist the work was completed on time and to a high standard with a final review and revision only required by the group leader with all the team to make sure all information included in the presentation was accurate.

We developed our skills in a variety of ways, by being able to critically evaluate several options and pick the most appropriate technology for our project we created an effective application which was able to function as expected and be expanded to include all the objectives, without this we may have found the options we chose may not have been able to meet our goals and our project may have failed. Our team was able to design our application from the outset with clear objectives, by utilizing UML design we were able to create an application that was fit for purpose and considered future expandability.

## 6.1. Lessons learned.

All group members engaged with the tasked assigned to them but after discussion with the group we found that we would have done the following differently:

* Set clear milestone goals - The group often fell behind with stages of the design and development process by setting milestone goals we would have been able to ensure the work was completed at the appropriate time.
* Discuss dependencies and organize deadlines appropriately.
* Enforce members using Trello – The group leader leads the Trello board (Appendix E) and although an effective way of managing the group members often forgot to check the board, by ensuring all members check the board on a weekly basis would have kept the team on track.
* Shorter deadlines with more regular meetings.

COVID-19 made our work much more difficult, the group had to adapt from typical methods as we were not permitted to meet in person, while some members were not in the county or permitted to enter for extended periods of time over the course of the project. We learned a lot of new skills for remote work, including using code collaboration tools such as VS live code for pair programming, using Microsoft teams to store our documents and hold meetings which would have typically been held in the university library, which lead to difficulty in attendance for some members.

## 6.2. Further work

Our app achieves its most important objective, informing staff of their shifts and allowing them to clock in and out but due to a range of issues we were unable to include all the features we wanted.

We have several ideas for how the work we completed could be expanded in the future:

* IOS support
* Web dashboard – To allow schedule creation using a computer.
* Break tracking
* Note keeping - e.g Disciplinary's and recommendations.
* Automatic Rota creation using machine learning.
* Swap shift capability.
* Automatic pay generation – links to payroll software.

# References

BCS, THE CHARTERED INSTITUTE FOR IT, 2020. *CODE OF CONDUCT FOR BCS MEMBERS.* [Online]   
Available at: https://www.bcs.org/media/2211/bcs-code-of-conduct.pdf  
[Accessed 17 December 2020].

CDC, 2021. *SARS-CoV-2 and Surface (Fomite) Transmission for Indoor Community Environments.* [Online]   
Available at: https://www.cdc.gov/coronavirus/2019-ncov/more/science-and-research/surface-transmission.html  
[Accessed 20 March 2021].

COING Inc, 2020. *Clockify - Free employee timesheet app.* [Online]   
Available at: https://clockify.me/timesheet-app  
[Accessed 13 December 2020].

Dart, 2021. *Dart homepage.* [Online]   
Available at: https://dart.dev/  
[Accessed 28 April 2021].

Deputy, 2020. *Deputy.* [Online]   
Available at: https://www.deputy.com/gb/features/time-and-attendance  
[Accessed 26 January 2021].

Djordjevic, B., 2019. *Time Tracking in 2019: What Are the Stats Saying?.* [Online]   
Available at: https://www.workpuls.com/blog/time-tracking-2019-stats  
[Accessed 9 May 2021].

GetPayroll.com, 2020. *Timeclocks & Overpayment.* [Online]   
Available at: https://cdn2.hubspot.net/hubfs/4054682/Infographics/GetPayroll%20-%202018%20July%20-%20Timeclocks.pdf  
[Accessed 26 January 2021].

Google LLC, 2021. *Firebase Documentation.* [Online]   
Available at: https://firebase.google.com/docs  
[Accessed 19 April 2021].

Google, 2020. *Material Design.* [Online]   
Available at: https://material.io/

Google, 2021. *Material design colour tool.* [Online]   
Available at: https://material.io/resources/color/#!/?view.left=0&view.right=0&primary.color=00838F&secondary.color=00838F&primary.text.color=ffffff  
[Accessed 11 March 2021].

Kent Beck, M. B. A. v. B. A. C. W. C. M. F. J. G. J. H. A. H. R. J. J. K. B. M. R. C. M. S. M. K. S. J. S. D. T., 2001. *Manifesto for Agile Software Development.* [Online]   
Available at: http://agilemanifesto.org/  
[Accessed 9 May 2021].

Muslihat, D., 2018. *Agile Methodology: An Overview.* [Online]   
Available at: https://zenkit.com/en/blog/agile-methodology-an-overview/  
[Accessed 10 May 2021].

# Appendix

## Diagram Description automatically generatedFull size concept map

## Diagram Description automatically generatedB. Full size use case diagram

## A picture containing text, electronics, lined Description automatically generatedC. Full size user flow diagram

## D. Full size Firestore design diagram

## E. Trello board

Graphical user interface, application

Description automatically generated

## F. Meeting Agenda

|  |  |  |  |
| --- | --- | --- | --- |
| Week | 26 | Date: | 1/2/21 |
| Members Present  Leader(Bold) | Emerson Darwin  Bikram Gaire  Seb Price-Thomas  **Nick McCaig** | | |
|  |  |  |  |
| Agenda: | **Description** | **Assigned to:** | **Deadline** |
| Assign Roles |  | Nick: Project Manager  Bikram:  Emerson: System architect  Michael: System architect  Seb:  Yueqi: Programmer | N/a |
| Demo Code |  | Nick | In meeting |
|  |  |  |  |
| Discuss other features |  |  |  |
| Review feedback from supervisor |  | Nick | In meeting |
| Organize time for code review | Each member needs to have a review of the code with Nick especially those who choose to continue work. |  | Completed in meeting |
| Assign component tasks | Programmers will need to be assigned tasks and a deadline |  | Tasks are assigned on trello |
| Discuss timeframe | Discussion |  | Discussed in meeting tasks are assigned deadlines on trello |
| Discuss workflow | Discuss how workflow will work, including  How code is reviewed  How testing will be done  Documentation |  | Discussed in meeting |
| Assign tasks in to do list | Tasks that can currently be completed are in the To do tasks section and will be assigned to members with deadlines |  | Tasks are assigned on trello |
| Comments: Messaged supervisor with concern of lack of engagement by Michael and Yueqi | | | |

|  |  |  |  |
| --- | --- | --- | --- |
| Week | 31 | Date: | 22/2/21 |
| Members Present | Emerson Darwin  Bikram Gaire  Seb Price-Thomas  **Nick McCaig**  *Michael Odejimi (Not attended apologies sent with reason before meeting)* | | |
|  |  |  |  |
| Agenda: | **Description** | **Assigned to:** | **Deadline** |
| Review progress |  |  | In meeting |
| Assign Tasks: | Nick will describe each task each member will then write the steps they will take by the end of the day and send to Nick to make sure all members are o n track | |  |
| * Firebase Security rules | This member will take on the role of security engineer for the firebase application | Michael | To be determined – will write up plan by 26/2 |
| * Testing Plan/ Unit Tests | Will write up a testing plan for the application | Bikram | Plan by 26/2 |
| * GANTT chart | Will amend the GANTT Chart in the | Emerson | 26/2 |
| * Deliverable 2 Format work | Will format the work over the weekend to be shown to group leader | Nick | 28/2 |
| * Data flow diagrams |  | Emerson | 26/2 |
| * Use Cases |  | Seb | At least 50% by 26/2 |
| * UML designed database | I have already created a diagram liked in the trello it needs to be made into a uml diagram |  | 26/2 |
| * Email Yueqi | Email is to be sent to Yueqi after continuing to not attend meetings and miss deadlines. | Nick | 22/2 by end of day. |
| Discuss scaling back |  | Nick | In meeting |
| Project scaled back because of losing a member/ as it became clear the initial goals were not achievable with current time frame. We chose to not create a dashboard and limit the functionality to solely the app. | | | |
|  |  |  |  |
| Comments:  Michael had a separate conversation with Nick following his concerns he has committed to meet the deadline of deliverable 2 to re-write his original tasks.  Yueqi Zhai was sent an email (22/2) included below.  CC: All group members.  Good afternoon Yueqi,  I am letting you know that following the rules set out in the group we have had to reassign the work you have been asked to do.  Following the 3-strike rule, I have attempted to communicate and ask from work from you on several occasions.  18/1/21 on teams I requested your part of the work you said you were unable to send it I suggested you contact Andreas to arrange an NEC or find a way to send the work. I followed this message on 19/1/21 asking if you found a way you did not respond.  16/1/21 I asked if you did the work on WhatsApp and offered you help if you needed it you did not respond.  2/2/21 after you did not attend a scheduled meeting I sent you the following message you did not respond "Hi, I just wanted to remind you of the importance of attending the meetings for PPM we are marked on our ability to work together and engage the module to solve problems including attending meetings so please let me know when you can't attend a meeting for whatever reason so I can note it in the agenda as to not affect your grade." You did not respond you have also not attended any group scheduled meeting since term 1 - you have also not informed the group you were unable to attend as per the rules in the definition document.  You have not engaged with the group chat on teams or logged into the Trello board except for agreeing to do the functional requirements which was reassigned when you missed the deadline. I informed you of the work you needed to do on WhatsApp and directed you to the Trello board for the deadline of the work which you have missed.  I hope you understand the choice the group has made to inform Andreas of your lack of engagement with the group. I hope all is well and if you have a legitimate reason for not completing the work then we will be more than happy to welcome you back with open arms to contribute to the project.  Please contact Andreas if you have a legitimate reason, I'm sure he will be happy to help arrange an NEC or any other support you need as we would all much rather have you in our group!  Kind regards,  Nicholas McCaig  Computer science 2nd year  N0787115 | | | |

|  |  |  |  |
| --- | --- | --- | --- |
| Week | 32 | Date: | 1/2/21 |
| Members Present  Leader(Bold) | Emerson Darwin  Bikram Gaire  Seb Price-Thomas  **Nick McCaig** | | |
|  |  |  |  |
| Agenda: | **Description** | **Assigned to:** | **Deadline** |
| Go over progress review |  |  | Completed in meeting |
| Remind members to write up on their report tasks |  |  | End of week 32 – 4th march |
| Remind Michael to do assigned tasks | deadline is 3rd of march with no progress | Michael | Discussion in meeting |
| Show app progress | App development shown in meeting | Nick McCaig | In meeting |
| Assign tasks in to do list | Tasks that can currently be completed are in the To do tasks section and will be assigned to members with deadlines |  | Tasks are assigned on trello |
| Comments:  Micheal did not attend | | | |

|  |  |  |  |
| --- | --- | --- | --- |
| Week | 33 | Date: | 8/2/21 |
| Members Present  Leader(Bold) | Bikram Gaire  **Nick McCaig** | | |
|  |  |  |  |
| Agenda: | **Description** | **Assigned to:** | **Deadline** |
| Review Previous meeting progress |  |  | Completed in meeting |
| Review design completion | Design decisions need to be finalized | Nick McCaig | Completed in meeting |
| Testing plan | Member needs to be assigned | Bikram Gaire |  |
| Comments:  Members not present sent agenda to confirm discussions in meeting. (All members should have been present) | | | |

|  |  |  |  |
| --- | --- | --- | --- |
| Week | 35 | Date: | 22/2/21 |
| Members Present  Leader(Bold) | Bikram Gaire  **Nick McCaig** | | |
|  |  |  |  |
| Agenda: | **Description** | **Assigned to:** | **Deadline** |
| Review Previous meeting progress |  |  | Completed in meeting |
| Review feedback | Feedback from meeting with project supervisor on trello. | Various individuals on trello. | 8/2/21 (next meeting) |
| Comments:  Members not present sent agenda to confirm discussions in meeting. (All members should have been present) | | | |

|  |  |  |  |
| --- | --- | --- | --- |
| Week | 37 | Date: | 8/3/21 |
| Members Present  Leader(Bold) | Emerson Darwin  Bikram Gaire  Seb Price-Thomas  **Nick McCaig** | | |
|  |  |  |  |
| Agenda: | **Description** | **Assigned to:** | **Deadline** |
| Review Previous meeting progress |  |  | Completed in meeting |
| Review feedback task progress | Some tasks were not completed their deadline has been extended | Seb Price-Thomas |  |
| Assign tasks in to do list | Tasks that can currently be completed are in the To do tasks section and will be assigned to members with deadlines |  | Tasks are assigned on trello. |
| Comments:  Members not present sent agenda to confirm discussions in meeting. (All members should have been present) | | | |

|  |  |  |  |
| --- | --- | --- | --- |
| Week | 38 | Date: | 15/3/21 |
| Members Present  Leader(Bold) | Emerson Darwin  Bikram Gaire  Seb Price-Thomas  **Nick McCaig** | | |
|  |  |  |  |
| Agenda: | **Description** | **Assigned to:** | **Deadline** |
| Review Previous meeting progress |  |  | Completed in meeting |
| Assign presentation tasks | Decision needs to be made on who will write/present our project | Seb Price-Thomas  Emerson Darwin | Final deadline 10/5/21  Next meeting – Start Script and layout presentation |
| Comments: | | | |

|  |  |  |  |
| --- | --- | --- | --- |
| Week | 39 | Date: | 22/3/21 |
| Members Present  Leader(Bold) | Emerson Darwin  Bikram Gaire  Seb Price-Thomas  **Nick McCaig** | | |
|  |  |  |  |
| Agenda: | **Description** | **Assigned to:** | **Deadline** |
| Review Previous meeting progress |  |  | Completed in meeting |
| Presentation | Review progress of presentation continue development discuss any technical aspect that needs to be included. | Seb Price-Thomas  Emerson Darwin | 12/4/21 |
| Use case definitions not completed | Group member not completed work reminder sent deadline extended | Nick McCaig  (Michael) | 5/4/21 |
| Comments:  Reminder sent as member was not present in meeting:  Graphical user interface, text, application  Description automatically generated | | | |

|  |  |  |  |
| --- | --- | --- | --- |
| Week | 40 | Date: | 29/3/21 |
| Members Present  Leader(Bold) | Emerson Darwin  Bikram Gaire  Seb Price-Thomas  Michael Odejimi  **Nick McCaig** | | |
|  |  |  |  |
| Agenda: | **Description** | **Assigned to:** | **Deadline** |
| Review Previous meeting progress |  |  | Completed in meeting |
| Testing review | Testing reviewed to check work with group. | Bikram Gaire | Completed in meeting |
| Comments:  Testing was on the right track. | | | |

|  |  |  |  |
| --- | --- | --- | --- |
| Week | 41 | Date: | 5/4/21 |
| Members Present  Leader(Bold) | Bikram Gaire  Seb Price-Thomas  **Nick McCaig** | | |
|  |  |  |  |
| Agenda: | **Description** | **Assigned to:** | **Deadline** |
| Review Previous meeting progress |  |  | Completed in meeting |
| Use case definitions reviewed | Delayed task was reviewed – member was not present but task has been completed. Member sent a thank you. | | Completed in meeting |
| Comments:  Members not present sent agenda to confirm discussions in meeting. (All members should have been present) | | | |

|  |  |  |  |
| --- | --- | --- | --- |
| Week | 42 | Date: | 12/4/21 |
| Members Present  Leader(Bold) | Emerson Darwin  Bikram Gaire  Seb Price-Thomas  **Nick McCaig** | | |
|  |  |  |  |
| Agenda: | **Description** | **Assigned to:** | **Deadline** |
| Review Previous meeting progress |  |  | Completed in meeting |
| Code review | Code reviewed with all members as well as a demo. | Nick McCaig | Completed in meeting |
| Trello board review | Tasks in trello checked and all members discussed progress |  | Completed in meeting |
| Comments:  Members not present sent agenda to confirm discussions in meeting. (All members should have been present) | | | |

|  |  |  |  |
| --- | --- | --- | --- |
| Week | 43 | Date: | 19/4/21 |
| Members Present  Leader(Bold) | Emerson Darwin  Bikram Gaire  Seb Price-Thomas  **Nick McCaig** | | |
|  |  |  |  |
| Agenda: | **Description** | **Assigned to:** | **Deadline** |
| Review Previous meeting progress | Meeting was quick discussion to inform each other what progress they made |  | Completed in meeting |
| Presentation | Demo of presentation | All members present | 22/4/21 |
| Comments:  Members not present sent agenda to confirm discussions in meeting. (All members should have been present) | | | |

|  |  |  |  |
| --- | --- | --- | --- |
| Week | 44 | Date: | 26/4/21 |
| Members Present  Leader(Bold) | Emerson Darwin  Bikram Gaire  Seb Price-Thomas  **Nick McCaig** | | |
|  |  |  |  |
| Agenda: | **Description** | **Assigned to:** | **Deadline** |
| Review Previous meeting progress | Meeting was quick discussion to inform each other what progress they made |  | Completed in meeting |
| Review presentation progress | Presentaion in current state reviewd and tasked assigned on trello to be completed by the following week | Seb Price-Thomas  Emerson Darwin | Completed in meeting |
| Demonstration | Member assigned to create video demonstration of the application | Nick McCaig | 6/5/21 |
| Code completion review | Code checked by members to confirm goals had been met. | (lead by Nick McCaig) | Completed in meeting |
| Comments:  Members not present sent agenda to confirm discussions in meeting. (All members should have been present) | | | |

|  |  |  |  |
| --- | --- | --- | --- |
| Week | 46 | Date: | 6/5/21 |
| Members Present  Leader(Bold) | Emerson Darwin  Bikram Gaire  Seb Price-Thomas  **Nick McCaig** | | |
|  |  |  |  |
| Agenda: | **Description** | **Assigned to:** | **Deadline** |
| Review Previous meeting progress |  |  | Completed in meeting |
| Demonstration review | All members watched demonstration video to confirm goals had been met. | (lead by Nick McCaig) | Completed in meeting |
| Presentation review | Presentation script and slides reviewed to confirm all goals had been met. | (Lead by Seb Price-Thomas / Emerson Darwin) | Completed in meeting |
| Comments:  Members not present sent agenda to confirm discussions in meeting. (All members should have been present) | | | |

## G Git log

commit dbfac186a4974ec98577823050bf1f65367ce356

Author: Nicholas McCaig <nick.mccaig@hotmail.co.uk>

Date: Tue Mar 2 21:22:51 2021 +0000

Refactoring

commit e8808a1344bd136d78766f6f21dee87089139306

Author: Nicholas McCaig <nick.mccaig@hotmail.co.uk>

Date: Tue Mar 2 16:40:18 2021 +0000

update

commit 72476fffecdd5a0e0a114ada5d32a7c3e6175d18

Author: Nicholas McCaig <nick.mccaig@hotmail.co.uk>

Date: Sun Jan 3 18:57:07 2021 +0000

loading update

commit 88aae8dfb7168f5a6130de096c67b07300d26e1a

Author: Nicholas McCaig <nick.mccaig@hotmail.co.uk>

Date: Sat Jan 2 11:30:11 2021 +0000

Fixed missing import glitch.

commit 4f0eff0260f741bedd19dab74e9b51d21b826c87

Author: Nicholas McCaig <nick.mccaig@hotmail.co.uk>

Date: Fri Jan 1 21:03:23 2021 +0000

Tabs

commit 45de7d59dbc3d740b54659ed13124b8fcc16eae3

Author: Nicholas McCaig <nick.mccaig@hotmail.co.uk>

Date: Fri Jan 1 15:50:28 2021 +0000

Update

commit 6d932448fc79d91da81b033dd831dfabcc4a04bc

Author: Nicholas McCaig <nick.mccaig@hotmail.co.uk>

Date: Tue Dec 29 15:43:33 2020 +0000

Clock Funconality

commit 8993898bdc2a3a3108d7b31a6973db659724d3e3

Author: Nicholas McCaig <nick.mccaig@hotmail.co.uk>

Date: Sun Dec 27 16:03:59 2020 +0000

Images/testing

commit 5d5e70b3cb71a84024cbd769d93f4dceec1ae400

Merge: 9a501cb 6fe2328

Author: Nicholas McCaig <nick.mccaig@hotmail.co.uk>

Date: Sun Dec 27 15:21:01 2020 +0000

Merge branch 'main' of https://github.com/NickMcCaig/RotaAppNtu into main

commit 9a501cbf9b0574a476b22c009f27dee1b6efa18d

Author: Nicholas McCaig <nick.mccaig@hotmail.co.uk>

Date: Sun Dec 27 15:20:47 2020 +0000

Update build.gradle

commit 6fe23281f09cd0ff764522509426f5f68e14d59a

Author: Nicholas McCaig <nick.mccaig@hotmail.co.uk>

Date: Sun Dec 27 15:14:29 2020 +0000

Update README.md

commit b1128deb9d3110d0dfc915853b6ad9e3a708149d

Author: Nicholas McCaig <nick.mccaig@hotmail.co.uk>

Date: Sun Dec 27 15:13:46 2020 +0000

Inital

commit 0874cc2a9c4632522e7b79905ede1f0c70bc003a

Author: Nicholas McCaig <nick.mccaig@hotmail.co.uk>

Date: Sun Dec 27 15:12:52 2020 +0000

Initial commit

## H. Software Testing

Table 3 - Software testing

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| # | Test | EXPECTED RESULT | ACTUAL RESULT | PASS/FAIL | FIX |
| 1 | Home page loads in with the bar at the bottom and the timer. | The expected result from this test was the application loads with the status bar and the timer. | The actual result from this test was the home page loading with a status bar. | Fail | The bar was not added at the bottom. |
| 2 | The home form loads with content. | The expected result from this test was to display the application name and the author’s name. | The actual result from this test was the home page includes the application name and the author’s name. | Pass | N/A |
| 3 | Login in form loads of content. | The expected result from this test was to display the content in Login in. | The actual result from this test was to display the contents in Login in. | Pass | N/A |
| 4 | Login buttons works | The expected result from this test was to test the login in buttons. | The actual result from this test was to test the login in buttons. | Pass | N/A |
| 5 | Leading QR code when clicking the three lines button. | The expected result from this test was to lead to the QR code while clicking the three lines button. | The actual result from this test was to lead to the QR code while clicking the three lines button. | Pass | N/A |
| 6 | Clicking the back button leads to the front screen from the. | The expected result from this test was to leading to front screen when clicking the back button. | The actual result from this test was to leading to front screen when clicking the back button. | Fail | There was no present of the back button in the application, but was added at the last |
| 5 | Clicking the close button closes the application. | The expected result from this test was to close the application while clicking the close button. | The actual result from this test was to close the application while clicking the close button. | Fail | As close button was not in need as the user can flick to different screen with the help of back button. |
| 7 | Clicking the clocking button shows the shift. | The expected result from this test was to shows the shift while clicking the clocking button. | The actual result from this test was to shows the shift while clicking the clocking button. | Pass | N/A |
| 8 | Shows notification, shift and New Schedule in Greggs Rota as a topic. | The expected result from this test was to shows notification, shift and New Schedule in Greggs Rota as a main topic. | The actual result from this test was to shows notification, shift and New Schedule in Greggs Rota as a main topic. | Pass | N/A |
| 9 | Clicking the calendar button shows the shift. | The expected result from this test was to shows the shift while clicking the calendar button. | The actual result from this test was to shows the shift while clicking the calendar button. | Pass | N/A |
| 10 | Clicking the new shift while clicking the “New schedule” button shows the shift. | The expected result from this test was to shows the new shift while clicking the “New schedule” button shows the shift. | The actual result from this test was to shows the new shift while clicking the “New schedule” button shows the shift. | Pass | N/A |
| 11 | Able to edit the data. | The expected result from this test was to data the rota. | The actual result from this test was able to delete the rota from the list. | Fail | This functionality was added because the rota should be able to be edited by the responsive person. |
| 12 | Notify user about the new rota. | The expected result from this test was to notify the user about the new rota. | The actual result from this test was to present a list from the combo box. | Pass | N/A |
| 13 | Application to run smoothly without crashing. | The expected result from this test was to run smoothly without crashing. | The actual result from this test was to run smoothly without crashing. | Pass | N/A |
| 14 | Search button at search form search typed items. | The expected result from this test was to search typed items while clicking the search button on the search form. | The actual result from this test was to search typed items while clicking the search button on the search form. | Fail | The search button was not included as there is minimum requirement of this task. |
| 15 | Manager able to create a staff. | The expected result from this test was to create a staff by the manager. | The actual result from this test was to a staff by the manager. | Pass | N/A |
| 16 | Manager able to delete a staff. | The expected result from this test was to create a staff by the manager. | The actual result from this test was to a staff by the manager. | Pass | N/A |
| 17 | Editing the profile. | The expected result from this test was to edit the profile. | The actual result from this test was to edit the profile. | Pass | N/A |
| 18 | Displaying the error message when wrong code is typed at QR code box. | The expected result from this test was to displaying the error message when wrong code is typed at QR code box. | The actual result from this test was to displaying the error message when wrong code is typed at QR code box. | Pass | N/A |
| 19 | Displaying the staff. | The expected result from this test was to display the staff added into the staff screen. | The expected result from this test was to display the staff added into the staff screen. | Pass | N/A |
| 20 | Displaying the weeks. | The expected result from this test was to display the weeks. | The actual result from this test was to display the weeks. | Pass | N/A |
| 21 | Staff link takes to staff screen. | The expected result from this test was to takes to staff screen after clicking staff option. | The actual result from this test was to takes to staff screen after clicking staff option. | Pass | N/A |
| 22 | Edit link takes to the editing screen. | The expected result from this test was to takes to edit screen after clicking edit option. | The actual result from this test was to takes to edit screen after clicking edit option. | Pass | N/A |
| 23 | The cancel button clears the context of all the input boxes. | The expected result from this test was to cancel button clears the context of all the input boxes. | The expected result from this test was to cancel button clears the context of all the input boxes. | Pass | N/A |
| 24 | The homepage screen is titled homepage. | The expected result from this test was to titled homepage. | The actual result from this test was to titled homepage. | Pass | N/A |
| 25 | The add staff screen is titled staff members. | The expected result from this test was to titled staff members. | The actual result from this test was to staff members. | Pass | N/A |
| 26 | Hyperlinks works. | The expected result from this test was working hyperlinks. | The actual result from this test was hyperlinks works. | Pass | N/A |

## H. Database Testing

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| # | Test | EXPECTED RESULT | ACTUAL RESULT | PASS/FAIL | FIX |
| 1 | The program should ask the user what time they would like to start. | The expected result from this test was for the software to ask the admin to enter the time they would like to start. | The actual result from this test was for the software to asking the admin to enter the time they would like to start. | Pass | N/A |
| 2 | The program should ask the user what time they would like to end. | The expected result from this test was for the software to ask the admin to enter the time they would like to start | The actual result from this test was for the software to asking the admin to enter the time they would like to start. | Pass | N/A |
| 3 | The program should ask the admin what day they would like to start. | The expected result from this test was for the software to ask the admin to enter the day they would like to start. | The actual result from this test was for the software to asking the admin to enter the day they would like to start. | Pass | N/A |
| 4 | The program should ask the user what day they would like to end. | The expected result from this test was for the software to ask the admin to enter the day they would like to start | The actual result from this test was for the software to asking the admin to enter the day they would like to start. | Pass | N/A |
| 5 | The program should ask the user the code or to scan the QR. | The expected result from this test was for the software to ask the user the code or to scan QR. | The actual result from this test was for the software to asking the admin to ask the user the code or to scan QR. | Pass | N/A |
| 6 | The program should ask the create a timesheet created by the admin. | The expected result from this test was for to create a timesheet created by the admin. | The actual result from this test was for creating a timesheet created by the admin. | Pass | N/A |
| 7 | The program should add the timesheet as many as the users had entered. | The expected result from this test was for the create a timesheet created by the admin. | The actual result from this test was for creating a timesheet created by the admin. | Pass | N/A |
| 8 | The program should add each timesheet to a list of timesheets. | The expected result from this test was to add each timesheet to a list of timesheets. | The actual result from this test was adding each timesheet to a list of timesheets. | Pass | N/A |
| 9 | The program should not crash if the user enters anything other than string. | The expected result from this test was not to crash if the user enters anything other than string. | The actual result from this test was not to crashing if the user enters anything other than string. | Pass | N/A |
| 10 | The program should not allow a negative number to enter in the timesheet. | The expected result from this test was not to allow a negative number to enter in the timesheet. | The actual result from this test was not to allow a negative number to enter in the timesheet. | Fail | An additional check were done again, however this function was not achieved due to the compacity of the code. |
| 11 | The hour per rate should be entered in the database. | The expected result from this test was to enter the hour per rate should be entered in the database. | The actual result from this test was to enter the hour per rate should be entered in the database. | Pass | N/A |
| 12 | The program should ask the user the phone number. | The expected result from this test was for the software to ask the user the phone number. | The actual result from this test was for the software to asking the admin to ask the user the phone number. | Pass | N/A |
| 13 | The program should ask the user the week of the year. | The expected result from this test was for the software to ask the user the week. | The actual result from this test was for the software to asking the admin to ask the user the week. | Pass | N/A |

## I. Report

## I.9.1. Introduction

The survey was done to test the application into various age group. Questions were asked to 100 people in Nottingham bearing in mind with the social distancing should be followed due to this situation. However, the response rate was 100%.

#### I.9.1.1 FINDINGS

The finding will be presented below into two sections: -

* Result
* Age

**AGE**

Fig.1

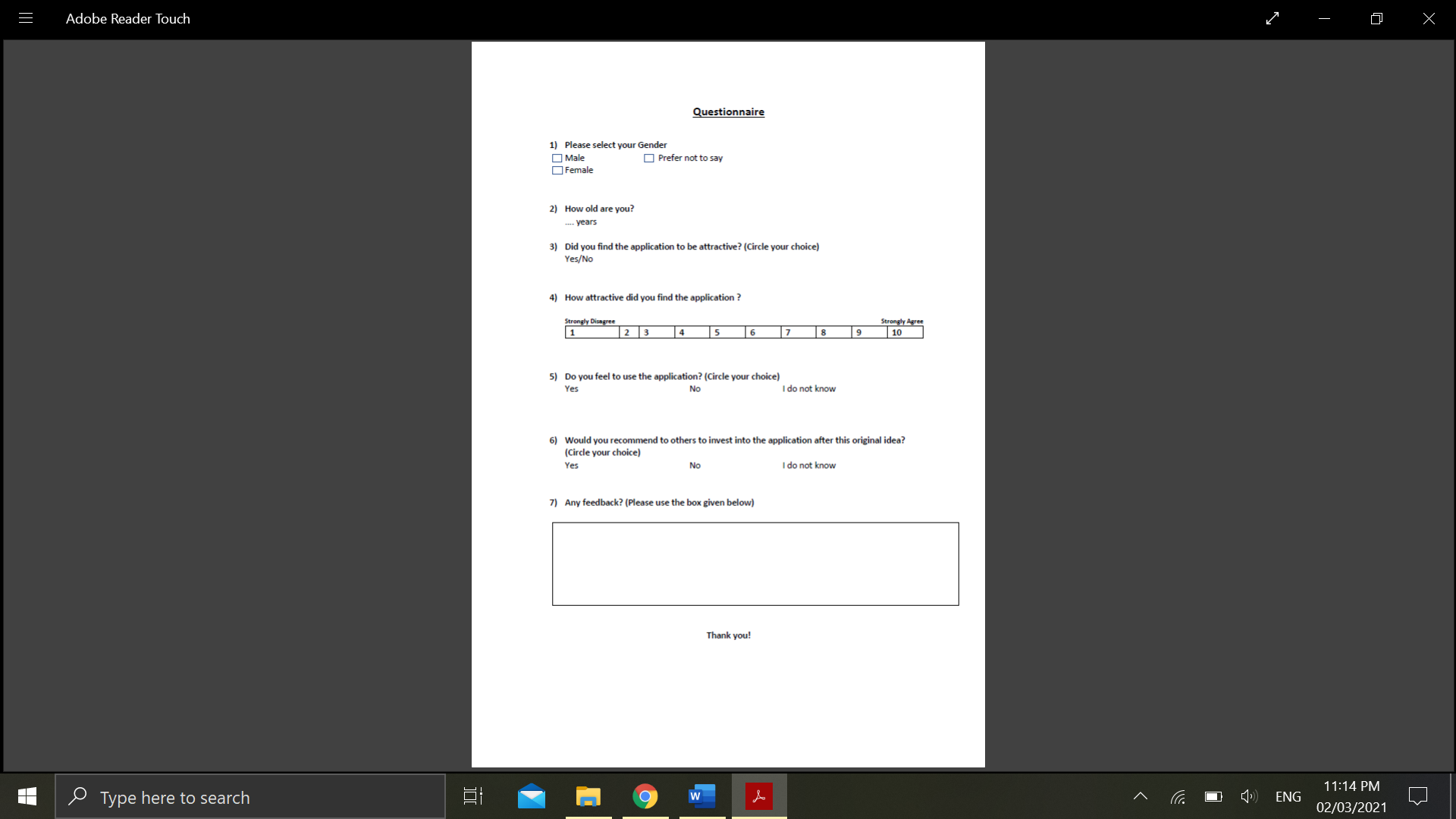
Different age group were questioned by me through face-to-face and phone. They are as follows: -

* 18-29
* 30-39
* 40-49
* 50+

#### I.9.1.2. RESULT

Fig.2  
  
From fig.2, many people believed the application is usable with 67%. However, 28% of people believed the application is not very effective as it does not include various functionality. Also, 5% of people were in dilemma to choose the answer and provided some suggestion to improve the application.

## J.10. Questionnaire



## K – Use case Diagrams

|  |  |  |  |
| --- | --- | --- | --- |
| Use Case Name: Log the times of the users | | | ID Number: 1 |
| Short Description: Users will log the time that they arrive to work and time they leave work | | | |
| Trigger: Tsheets | | | |
| Type: | External | | |
| Major Inputs Description:  The user will be able to view their shifts on the app, and it will allow them to enter the clock code or scan the QR code which will automatically enter in the correct clock code. Once the correct code has been entered in, the user will click the clock in button which will register the time that they clocked in. When their sift ends they will be able to click the clock out button which will register the time they clocked out    Source:  User | | **Major Outputs Description:**  The returned information will that the user has clocked in for their shift, once they have clocked out the returned information will be that the user has clocked out of their shift. The timer will count the hours of work and it will also count any overtime taken.  **Destination:**  User | |
| Major Steps Performed:   1. The user checks if they have an upcoming shift 2. The user scans the QR code or manually enters in the clock code 3. The user clocks in 4. The user clocks out | | **Information Required:**   1. An upcoming shift is required 2. User enters correct clock code or scans QR code | |
|  |  |  |  |

|  |  |  |  |
| --- | --- | --- | --- |
| Use Case Name: Confirm user is in designated work location | | | ID Number: 2 |
| Short Description: Confirms the user is in the correct location when they clock in and when they clock out. This prevents any staff clocking in at work when they aren’t at work | | | |
| Trigger: Tsheets | | | |
| Type: | External | | |
| Major Inputs Description:  In order for the app to work the user will need to making sure that GPS tracking is enabled on their device. Access will be denied if permissions for GPS is not given. Once the user scans the QR code or manually enters in the clock code. It will register the user location  Source:  User | | **Major Outputs Description:**  The returned information will that the user has clocked in for their shift and their location will be stored. This also happens for when they clock out.  **Destination:**  User | |
| Major Steps Performed:   1. The user will need to make sure GPS permissions are given 2. User will need to clock in to register location 3. User will need to clock out to register location | | **Information Required:**   1. User will need to scan QR code or enter clock code 2. User will need an upcoming shift | |
|  |  |  |  |

|  |  |  |  |
| --- | --- | --- | --- |
| Use Case Name: Adding a shift | | | ID Number: 3 |
| Short Description: How to add a shift | | | |
| Trigger: Tsheets | | | |
| Type: | External | | |
| Major Inputs Description:  Managers will be able to create shifts and assign them to different users. Managers will need to click a user for example alex nelson, then they will need to click add shift. Two dropdown menus will appear “Role” and “Week”. They will need to input what role and what week and set the time of the shift.  Source:  User | | **Major Outputs Description:**  The returned information will that the user has added a new shift for another user.  **Destination:**  User | |
| Major Steps Performed:   1. The user will click a employee to add a shift to 2. The user will pick what role and week 3. The user sets the time for the shift | | **Information Required:**   1. What user to add new shift 2. What role and week 3. Time of the shift | |
|  |  |  |  |

|  |  |  |  |
| --- | --- | --- | --- |
| Use Case Name: Checking correct code is submitted | | | ID Number: 4 |
| Short Description: Ensures that the users enters in the correct clock code | | | |
| Trigger: Tsheets | | | |
| Type: | External | | |
| Major Inputs Description:  The user will be prompted to manually enter the clock code, or they can scan the QR code which will automatically enter in the code.  Source:  User | | **Major Outputs Description:**  If the incorrect code has been entered in then the returned information will be that the check code is incorrect. The code needs to be 5 digits long otherwise it will inform the that the to enter the correct length code  **Destination:**  User | |
| Major Steps Performed:   1. The user checks if they have an upcoming shift 2. The user scans the QR code or manually enters in the clock code 3. The user clocks in | | **Information Required:**   1. An upcoming shift is required 2. User has correct clock code or scans QR code 3. Code has to be 5 digits long | |
|  |  |  |  |

## L – Trello boared agile development

