

DELTA

The Contemporary Employee Management System

Done by: Team 53

| | |
|---|-----------|
| Introduction | 3 |
| Planning and Research | 3 |
| Milestones and Processes | 4 |
| Prototyping | 5 |
| Iterative Design: | 5 |
| Our design decision & justification process: | 14 |
| System Development | 18 |
| Scope | 18 |
| User Stories | 18 |
| Sprint 1 (4th August 2021 - 14th August 2021) | 22 |
| Sprint 2 (15th August 2021 - 27th August 2021) | 27 |
| Sprint 3 (28th August 2021 - 7th September 2021) | 36 |
| Results of Sprints | 43 |
| Summative Review of Project: | 46 |
| Technologies and External Packages used: | 46 |
| The Core Components of the System: | 47 |
| Discussion of results | 71 |
| Evaluation | 73 |
| Original proposal & plan vs results | 73 |
| Analysis of our products vs others in the market: | 77 |
| Things to improve: | 79 |
| Push notifications | 79 |
| Enhanced security through biometrics | 79 |
| Time-in Time-out | 79 |
| Learning On the Move | 79 |
| Advanced Recruiting Tools | 80 |
| User Guide & Documentation | 81 |
| Login | 81 |
| Main page | 82 |

| | |
|---------------------------------------|------------|
| App Navigation - Menu Drawer | 83 |
| User Profile - Personal Data | 84 |
| User Profile - Work Stats | 85 |
| Request | 86 |
| Leaves | 87 |
| To-Do List | 89 |
| FAQ | 90 |
| Announcements | 91 |
| Logout | 92 |
| Test Case | 93 |
| Check the Login Functionality | 93 |
| Check the Main Page - Navigation Menu | 97 |
| Check Calendar Add Event Page | 100 |
| Check the Leave Application Page | 104 |
| Check the Request Page | 110 |
| Check the User Profile Page | 112 |
| Check the To-Do List Page | 114 |
| Check the Announcement Page | 120 |
| Check the FAQ Page | 123 |
| Check the Logout Page | 125 |
| Conclusion and Future Work | 127 |
| Personal Reflection | 129 |
| References | 130 |

DELTA

Introduction

We created an employee management system run on a mobile app to help HR managers in an organization effectively connect and engage with their employees. This improves communication and saves time and money in allocating tasks for teams. DELTA aims to combine the power of HR, payroll and time to deliver a personalised experience for its users.

****For employees:***

It is a personalised app that allows employees to view their personal details, payslip, company announcements and notifications. DELTA-53 also provides other useful functionalities such as a work calendar for easy scheduling purposes.

For HR personnel:

It is an app that allows HR managers to effectively manage an organisation's people and through the use of a database. It allows for easy tracking, management, and approval purposes such as issuing payslips, broadcasting announcements as well as leave and attendance.

In this report, we will be discussing DELTA as a personalised app for an organisation's employees.

Planning and Research

To create the application, we researched many different coding languages we could use to code with the app and eventually picked Flutter.

Flutter is an open sourced UI software development kit created by Google, and can be used to develop cross platform applications for Android, iOS and the web. Other languages we considered were Java & Kotlin.

For our database, we settled on Firebase. Firebase is a MBaaS (Mobile back-end as-a-service) that offers real-time database management, and a pay-as-you-go pricing approach. Another database considered was SQLite as it felt slightly more complicated to use than Firebase.

In regards to the working process, as a team we chose Scrum/Sprint. For designers, Iterative Design. We chose these as our work process as we felt that this was the best fit. We have a total of three sprints taken to finish our application. For details of the breakdown, please view the System Development section. Iterative design, the process will be discussed in the Prototyping section and a brief touch on it in the System Development section.

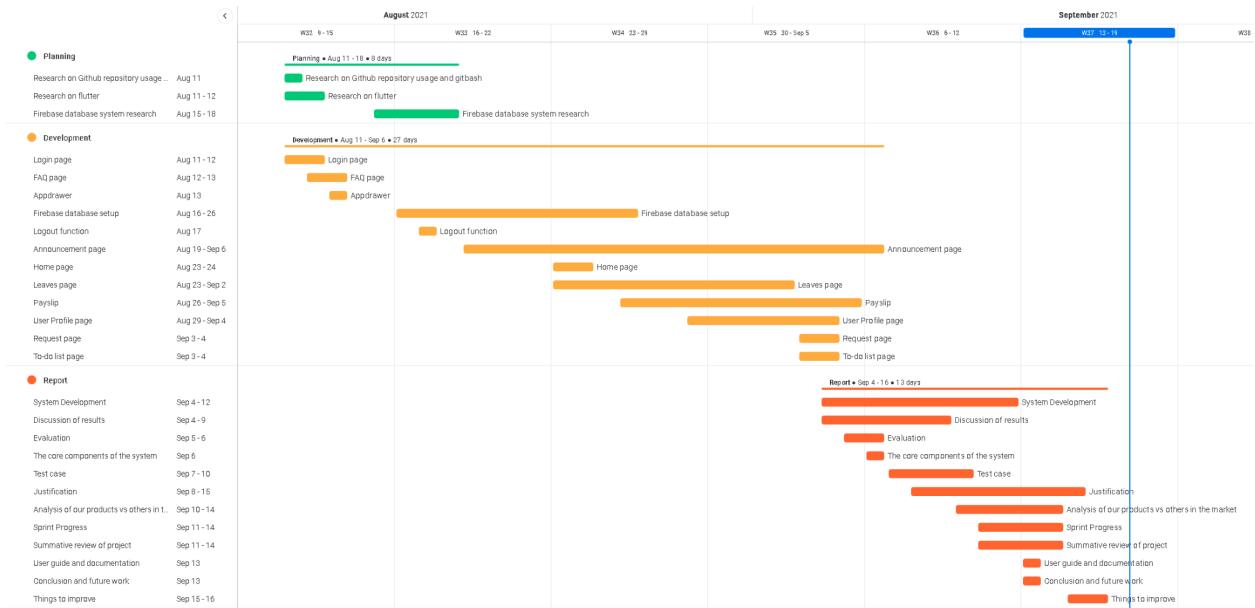
Milestones and Processes

This is the gantt chart showing all the different stages throughout the process of building the app.

The initial planning consisted of doing the research required to proceed with the project. This includes researching on how to use github and gitbash to manage the main repository shared amongst the developers. Then we had to research on how to use firebase to manage our online databases required to make the app work.

During the development stage, we tried to stay as close to the original wireframe design as much as possible. We implemented the login page, faq page, the appdrawer to access all the different pages in the app, setup of the firebase database system, a function to log out from the app, announcement page, home page, leaves page, user profile page which included the users profile and information regarding the amount of leaves they have, their attendance and their payslip, request page and the to-do list. There is some functionality missing as well as the notification page. We felt that the notification page was made redundant by the announcement page.

As for the report, it shows basically the contents found in this report. It shows the system development, discussion about the results of this project, evaluation of the project, the core components making the app, test cases, our justification for why we designed the app that way, analysis of similar products on the market, the break down of the sprint progress, summative review of our project, a guide to demonstrate how the app works as well as the conclusion and things we could improve on in the future.



Prototyping

We based this prototype off our first document where we first introduced Delta. We created it based on our envision of the wireframes designed and made improvements as we went along. We built basic functionalities of our application and added more functionalities as we progressed. The stakeholders were generally happy with the first high-fidelity wireframes so there were no changes from there. The main addition was the To-Do list function for the application. In iterative design, we will discuss our thought process for the new function and how we kept the interest of the stakeholders as priority. We will explain the reasons behind our design and how much it obeys the heuristics evaluation. Finally, building the prototype and feedback from the stakeholders.

Iterative Design:

It was towards the end of Sprint 2 that we received a request from the stakeholders about incrementing another function. This section will be discussed in terms of weeks.

We first gathered information and requirements about the users, in this case our stakeholders. We gathered information following the Early Focus on Users and Tasks principle.

Week 1 (18th August 2021)

Early Focus on Users and Tasks

1. Users' tasks and goals are the driving force behind the development.

The tasks and goals of this To-Do list function - the stakeholders want this function to provide users a better sense of responsibility to ensure their tasks are kept on track.

2. Users' behavior and context of use are studied, and the system is designed to support them.

We gathered information about user behaviour from the stakeholders as well as from ourselves. What we have discovered is that many prefer to work on To-Do lists like a notepad, when they are done, they either strike out the task or mark a check next to the goal. Some might make another page called Completed and then write down the completed task. Other options to display their To-Do list is in the form of pin up boards, where users will pin sticky notes and have a label at the top to indicate the progress from start to finish. Another option was with whiteboards, same process with sticky notes but just a different base.

3. Users' characteristics are captured and designed for.

We took into consideration the characteristics of the function and how it should be designed. We try to ensure that as the users use the To-Do list, there would be little errors (in design and functionality) and it would be simple to use. We took into consideration only cognitive aspects and nothing on physical aspects as it is a function, and it revolves mainly on tapping the function. Our focus for cognitive aspects is attention, perception and reading.

Attention-wise, we understand the focus of the function and how it is being conveyed to the user with an appropriate presentation of information. The intention of the function is to promote how one can focus their attention to any task they wish to add. We will keep the design nice and clear so that focus is a lot easier, and keep in mind not to make it too cluttered.

Perception-wise, the colours used will be similar to the application theme but ensuring that the fonts and buttons are able to stand out but not too much. We will consider using a separator to divide between tasks finished and unfinished, or tabs for the user to alternate.

Reading-wise, tasks should be easy to read and not too long. The shorter the task is written, makes reading and processing of information faster. Reading can be easily achieved if it is designed from an item that matches to the real world, which is either the notepad or the whiteboards or pin up boards.

4. Users are consulted throughout development from earliest phases to the latest.

5. All design decisions are taken within the context of the users, their activities, and their environment

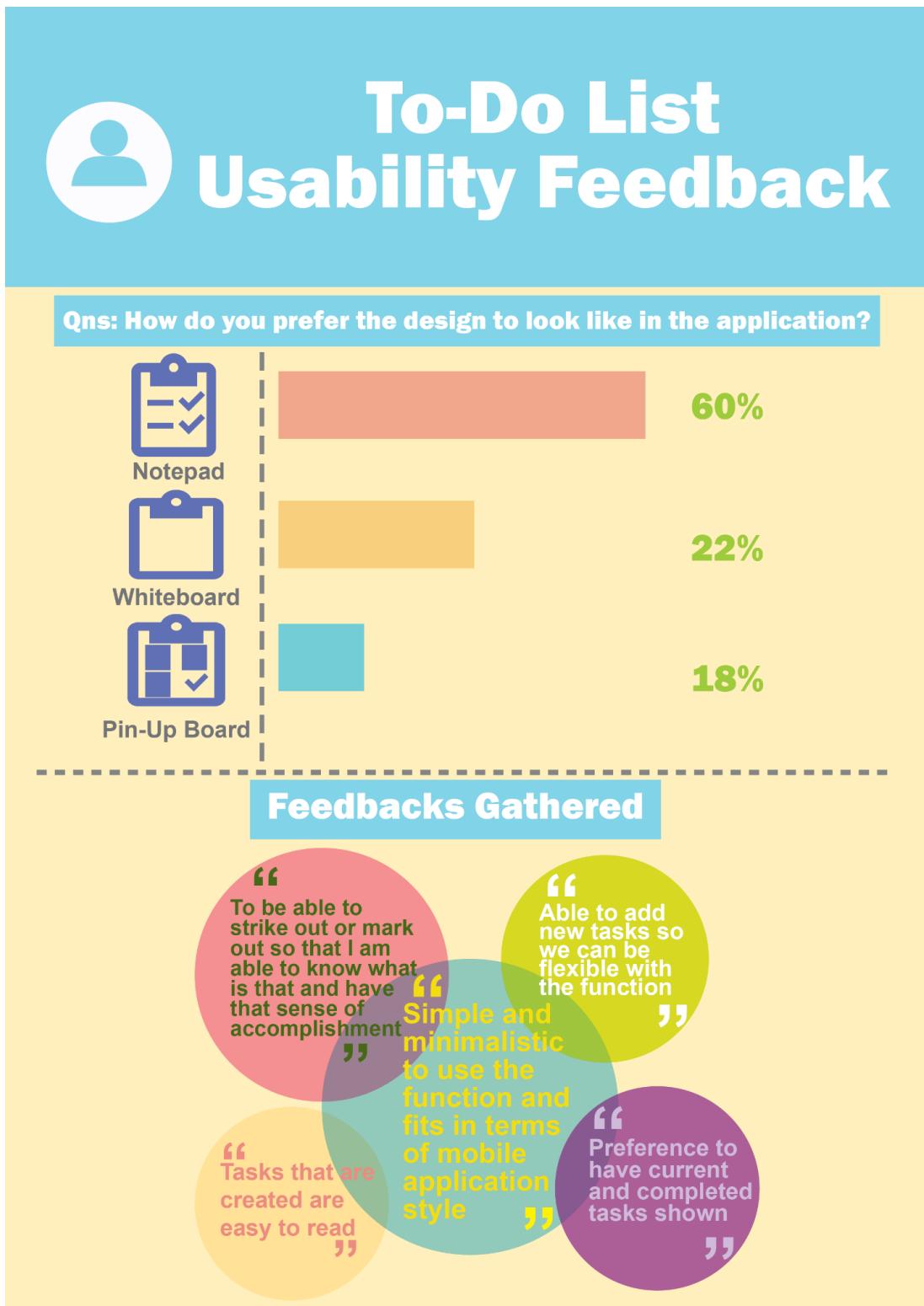
We consulted the stakeholders when there was a sprint review. Using the survey results, we adjusted the design based on what the stakeholders and potential users will like.

The environment can be anywhere but mainly in the office since it is a function from a mobile application. Their activities are mainly office work but the team felt that this function is indeed flexible.

Survey link:

<https://forms.gle/9x7xaRDeepsxBiMw8>

The results gathered from our surveys are as followed:

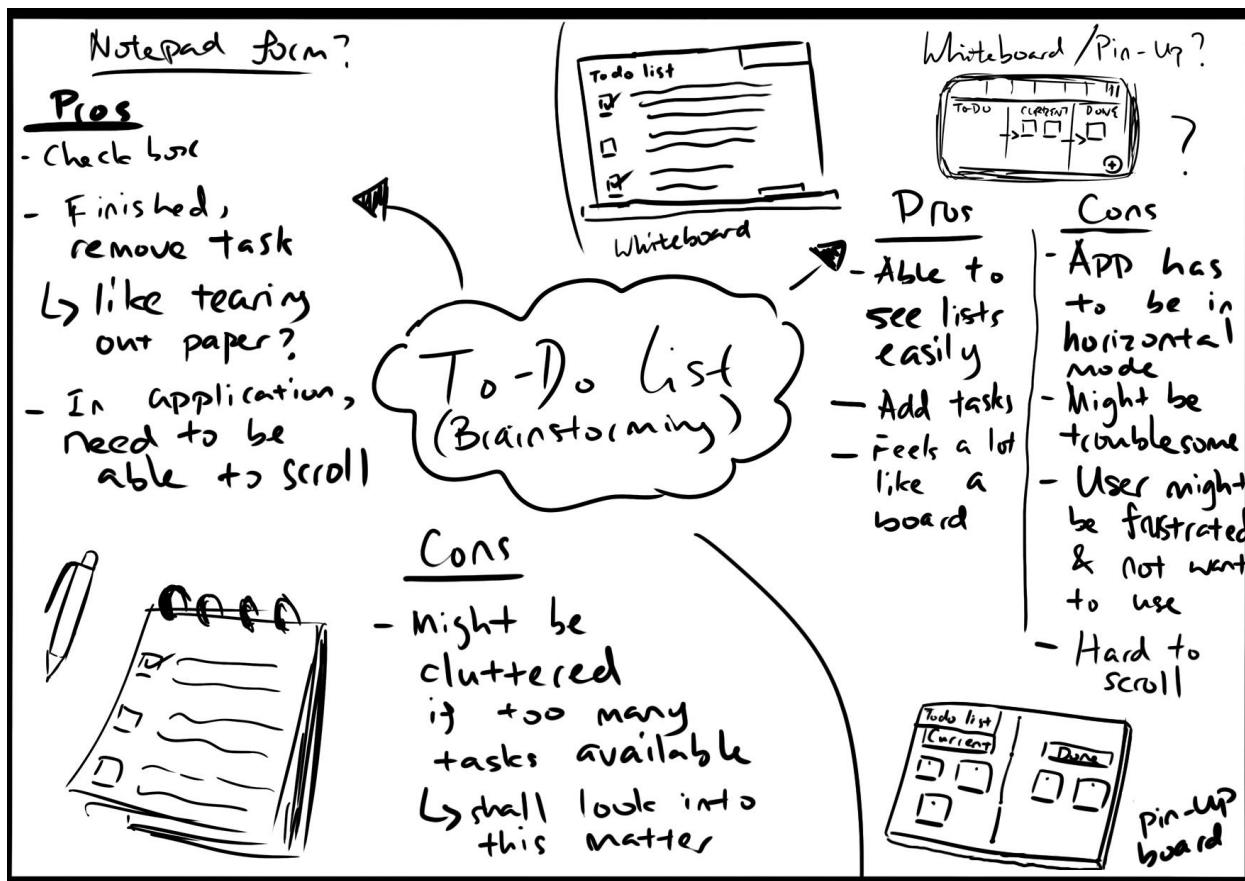


Week 2 (25th August 2021)

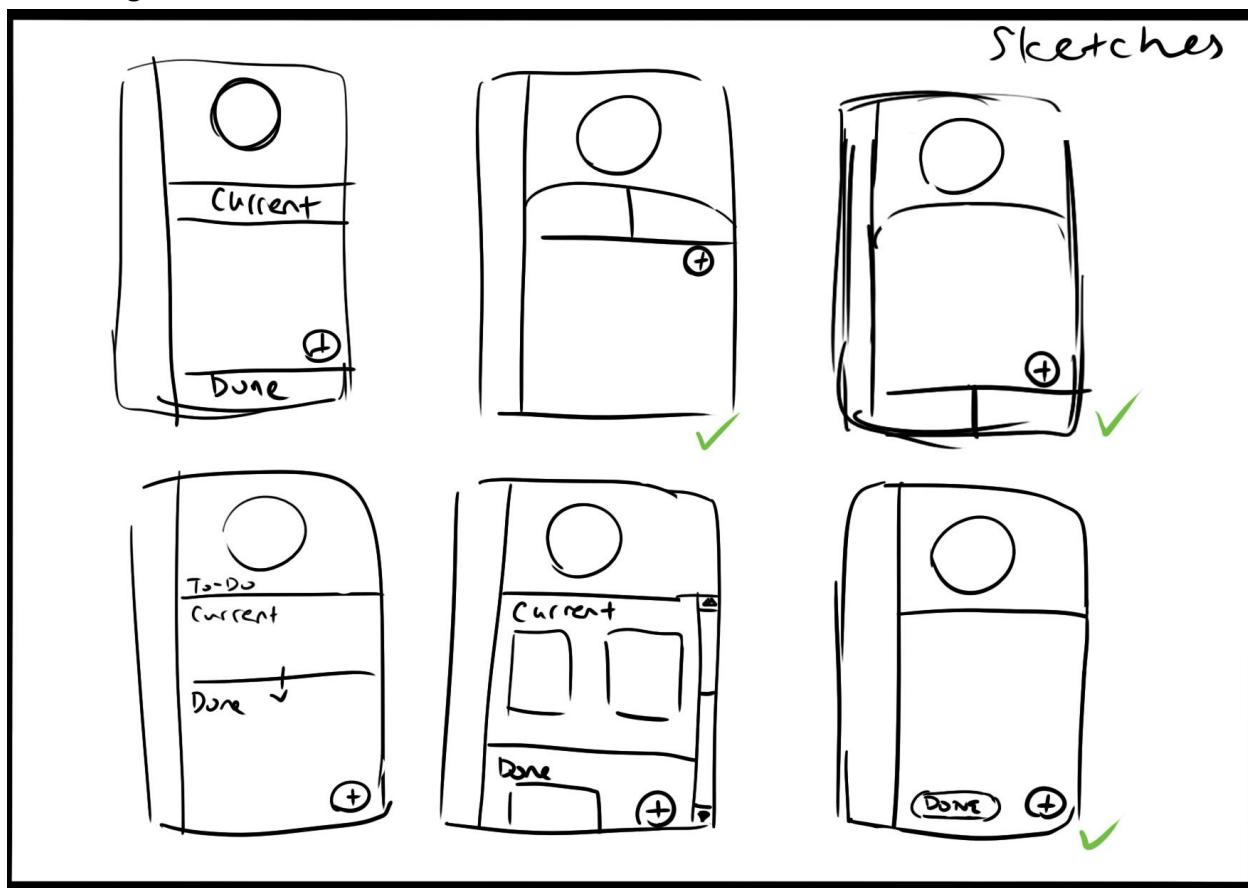
We came up with the wireframes needed for the function to be built, while keeping the interests of our stakeholders as priority. The designer came up with a couple of sketches and then came up with one to be chosen. By comparing the results, we chose three possible designs as the designs fit most of what the users need. And finally, we singled out one final design to work as the low prototype.

The brainstorming, sketching and 3 iterative designing processes will be shown below and explained:

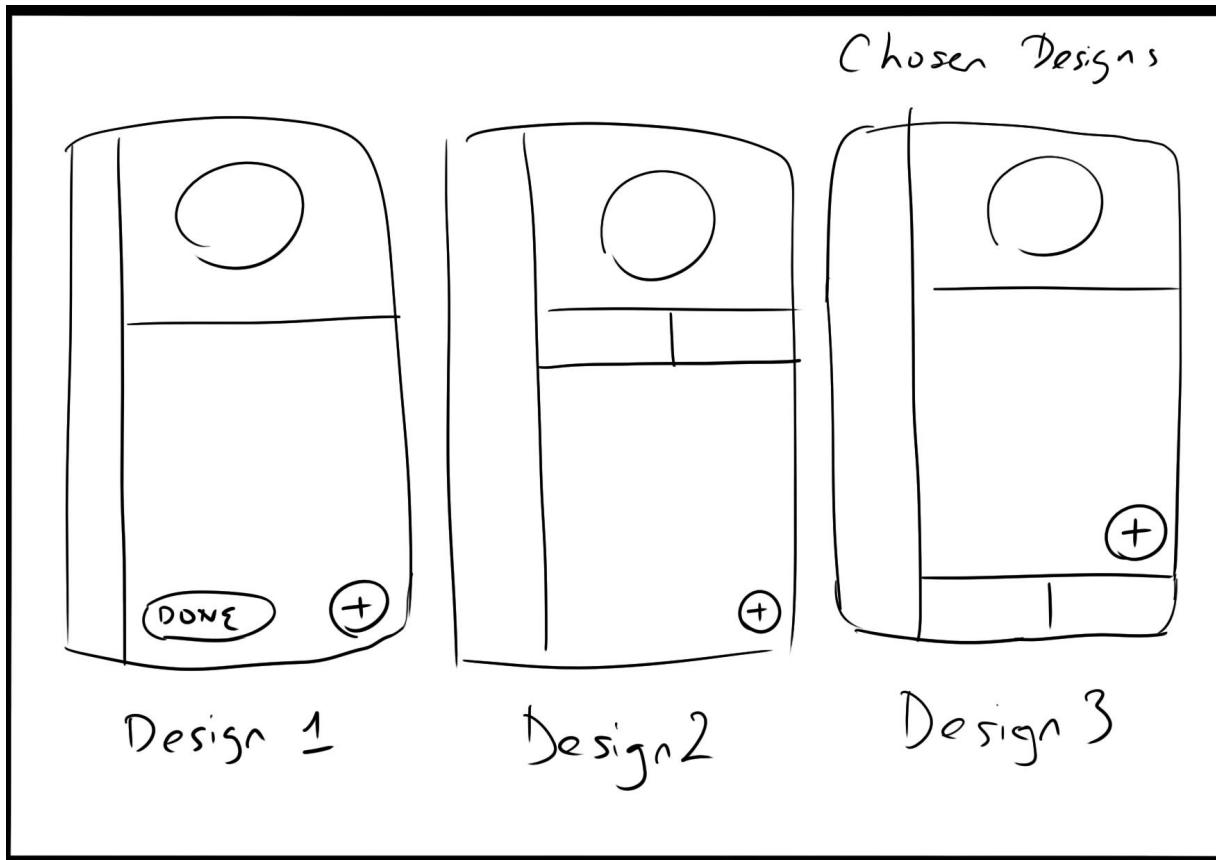
Brainstorming design:



Sketching



3 iterative designs



All 3 designs will be based on a notepad.

From design 1, we liked the idea of alternating pages with a single button and seems to be more minimalistic in terms of aesthetics. The button will change color each time the user wishes to see the current and completed tasks. The add new task button is right next to it so the user does not really have to look around to find where to add new tasks. However, error prevention will be difficult because users might make mistakes like finding where the completed tasks will be. New users might not know where to find their completed task when they first use it. If we are dealing with new or older users, then they might have problems recalling how to alternate between current and completed tasks. Hence, even though the idea is minimalistic and appears simple to use, we decided not to select design 1 from these reasons.

From what we gathered from design 1, we tried to resolve that issue in design 2. For design 2, we liked how the tasks and completed tasks are side by side - it is easy to see as the user just needs to alternate between pages. This avoids confusion and error prevention. The add button is located at the bottom because users usually read from top to bottom. However, we did not choose design 2 because we felt that if the user finished a task, it would be more appropriate to

add the button below instead. This hinders the user's recognition of the process as it feels that there might be a hindrance in the flow of this function. Hence, we did not choose design 2.

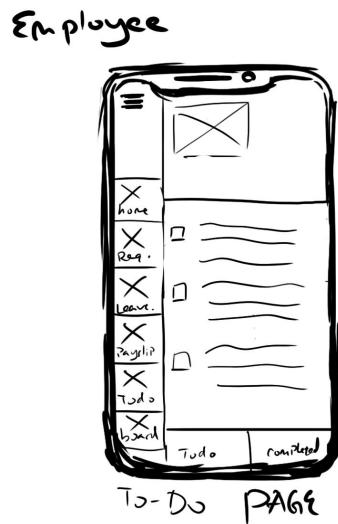
For design 3, the designs with 2 and 3 are similar, but with a different placement of the current tasks and completed tasks. We like how it solves the issue in design 2 and now the user has a clearer flow of using this function.

When the user reads from top to bottom, they are able to see clearly the current and completed tasks, as well as the add button right next to it. In terms of the 10 usability heuristics evaluation, we felt this hits the most points. It seemed to reduce the errors that a user might encounter when using the app because it is simple enough to understand - the app interface has a minimalistic view and the system is a close match with the real world. There is room for consistency to be promoted for this function as well. Hence, we decided that design 3 is the way to go.

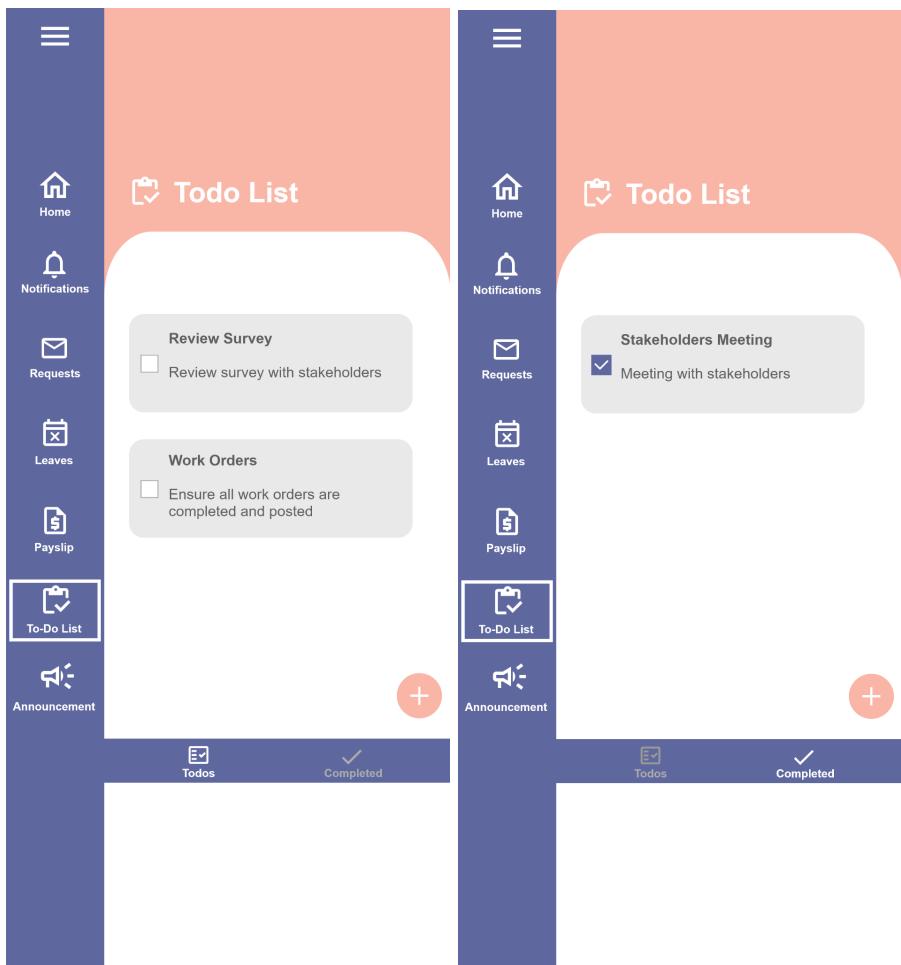
We showed the stakeholders the shortlisted three designs and we were right that they preferred design 3. As we would not like to waste time on it, we created a high-fidelity prototype instantly.

During the sprint 2 review, the stakeholders were happy with the design and were satisfied that it meets the needs of the users. They gave us the go ahead for this to move on to the function as there were no reviews or any critique.

Low-Fidelity Wireframe



High-Fidelity Wireframe



Week 3 (3rd September 2021)

We moved onto implementing the function similar to the wireframe, and then presented it to the stakeholders during the sprint 3 review. We received no critiques for this presentation as they were happy with what they saw and it satisfied their requirements and needs. Some parts of the iterative design will be reiterated during the sprints.

Our design decision & justification process:

The table below shows our design decisions and why we chose to add/remove certain features.

| | Design decisions |
|--------------------------------|---|
| Login | <p>The login page is where the user will enter in their credentials so that the app can confirm which user is currently logged in and show all relevant information, from public info to personal info to the user.</p> <p>We added this feature for authentication and security purposes. Only employees from the company can access the system with their company-registered email and password.</p> <p>We designed this using Firebase's Authentication Service.</p> |
| Main Page - Calendar widget | <p>We utilised a calendar with a weekly view for the user to view/add events to their personal calendar. All the events will be displayed in a scrollable vertical timeline based on the days selected.</p> <p>To do so, the user would just need to click on the '+' button on the left hand corner.</p> <p>The user can select the date (to-from), timing and add an event name, choose the timing and add a short description to the event.</p> <p>We added this feature because the main page is the user sees right after logging in. This provides an immediate consolidation of their day's activities and gives them a refresher of their tasks to start the day. The user has the freedom to choose the date range and time. The user is aware that they are on the home page thanks to the navigation panel. The function is easy to use and it is flexible to users as the moment they see the function, it would just tell them what event they have for today and at what time the event starts.</p> <p>We designed this using the Syncfusion Calendar from Flutter. We used a week view calendar with a day view timeline and added a form widget to collect information about the event and display it afterwards.</p> |
| Requests | <p>Users are able to view leaves requests made and the status of the request with the help of firebase. We linked the firebase collection of leaves and requests together such that all leaves made will be displayed onto the request page, waiting to be approved/rejected by the HR. The HR can change the status of the request by amending the firebase directly.</p> <p>We added this feature because it will be easier for users to view all the leaves requests they have made and track them. Not only that, users can also view the details with just a tap such as the date of leaves and also the leaves type and the reason for leaves.</p> |

| | |
|--|---|
| | <p>We designed it using firebase and list view. We did it by accessing the user id of the login user and retrieving the information from the leaves collection and displaying it as a list view. This is much easier to view all the contents instead of the calendar widget in the leaves page as that page is only used for applying leaves while the request page is used for viewing status of leaves.</p> |
| Leaves | <p>Users are able to apply for leave (medical, casual, maternity, compensation time off, etc) according to their needs and wants. This leave application is supposedly stored on firebase, where the HR will be able to view the proposed leave dates, and from there, HR will decide to approve it or to decline the leave application from the firebase itself.</p> <p>We added this feature because the leave page was designed in a way where users are able to easily choose their desired dates and time for their leave application, and this feature is extremely vital in a company as it shows the total manpower at any given day and time. The goal for this feature is that the user must be able to select the date of the leave and enter into the description. It must be able to be stored in the firebase and after logging out and back in again, the information is still there. The user has the freedom to choose the date range. It has visibility of system status due to the user selecting the leaves page indicating that they are on the leaves page. Match between system and the real-world is definitely there as anyone would recognise a calendar. The function is easy to use and it is flexible to users in terms of date selection, as most people would have used a calendar to jot down important dates.</p> <p>We designed it using a calendar widget, a form as well as firebase. The calendar widget shows the view of the month, in which users can select a date and fill in details using the form widget. This information is passed to the firebase for HR's use. This applies for every user who applies the leaves, the information will be stored in the Leaves collection in the Employees firebase. HR can amend the leaves status and reflect it on the firebase which the request page can retrieve and display.</p> |
| User Profile (payslip, stats, personal data) | <p>There are two tabs under the user profile - work statistics and personal data.</p> <p>Under work stats, the user can find data relating to how many leaves they have taken for the current year, how many days of the year they have attended and their payslip.</p> <p>Under personal data, we can view all information about the logged in user such as name, id, email etc.</p> |

| | |
|--------------|---|
| | <p>We added this feature because users can see all the information clearly on one screen so that they can better track their personal records. Our goal for this feature is to ensure that they can find their personal information easily. Users can have the freedom of alternating between profile and work stats easily. It is easy to use as the ability to alternate between one information to another is easy. Their visibility status is clear thanks to the navigation page. There is no room for error as the information is from the firebase and can only be wrong due to wrong information provided by the user before keying it in. Users do not need to have the urge to recall any part of the function as the information is just right in front of them.</p> <p>We designed it using a list view and the information from the employee collection in firebase. On the user interface, the Work Stats tab displays, from top to bottom, Leaves, Attendance and Payslip. The personal data shows the name of the user, position and other personal information such as email and birthday.</p> |
| To-Do List | <p>Users can make a list of all the tasks they need to complete during the day, using the add function. Once the task is marked as completed, it will go under the completed tab.</p> <p>We added this feature because users can have an overall view of the tasks needed to be done on that day itself, and they can allocate which ones are of high priority. One of the most important reasons to utilise a to-do list is to stay organised. When users make a list of all their tasks and responsibilities, they become more manageable. It is easier to keep focused when users have a clear outline of the tasks they need to do and those they have finished. While freeing up mental space for more creative endeavours.</p> <p>We designed it using firebase to store the string of text. Once the task is stored, it will be saved to the firebase and it can only be viewed by the user. This task can be edited or deleted.</p> |
| Announcement | <p>Users are able to pin announcements to the announcement board with the add function for other users to see when they go onto the announcement tab.</p> <p>We added this feature because it can serve as a reminder or a platform for users to leave important announcements for the whole company/organisation to view.</p> <p>We designed it using firebase to store the string of text including the</p> |

| | |
|--------|--|
| | <p>title and the body of text. Once saved, the announcement is saved to the firebase and is then displayed to everyone. The announcement can be edited and deleted.</p> |
| FAQ | <p>In FAQ, this is where instructions and guidance are provided to users to maneuver around the application.</p> <p>We added this feature because users of this app will go to a well-written and well-organized FAQ page as their first source of reference. It will assist them in locating solutions to questions that they would otherwise approach you about on an individual basis, taking time away from your many other responsibilities.</p> <p>People want to handle their problems quickly and independently, rather than waiting for a live chat agent or being put on hold for an hour. Having a comprehensive set of FAQs on hand may assist our current users in troubleshooting and solving common issues on their own, allowing them to return to enjoying our app.</p> <p>We designed it using fixed text as the FAQ questions and answers are not dynamic and will only need updating occasionally through the code. A database was not required for the FAQ to be implemented.</p> |
| Logout | <p>The user is able to log out from the current session by clicking on this button. A prompt will be given to confirm the logout request</p> <p>We added this feature because the user may need to logout from their device either to switch accounts or to keep their data secure while the device is unattended.</p> <p>We designed it using a popup prompt that asks the user to confirm if they want to logout. If the user clicks on confirm, the page will route back to the login page for the user to log back in or switch accounts.</p> |

System Development

Our approach: Scrum, Sprint

Scope

For this project we made an application for the employees first. The application has basic functionalities that allows employees to display their personal information, add or remove leaves and an announcement page to allow all employees, etc. We have adhered to the user's requests and complied to all of it. We added new functionalities such as a To-Do list and user profile. We removed the notifications page as it felt redundant as we have already included an announcement page.

User Stories

Roles: User, HR Employee, HR Executive, CEO

| ID | User Story | Requirements |
|----|---|--|
| #1 | As a user, I want to have a To-do list so that I can see my tasks to do for the day. | <ul style="list-style-type: none">• Make To-Do list page• List out work required for the day• Add To-Do list if user wish to edit• Data added to Firebase |
| #2 | As a user, I want my profile to be seen on the application so I can see my personal details. | <ul style="list-style-type: none">• Make a user profile section or page• Display personal details such as birthday |
| #3 | As a user, I want a page that is colourful, or something different from the application so that it feels more vibrant and fun to use. | <ul style="list-style-type: none">• Page showing something that is fun• Use of different colors to show it is fun |
| #4 | As a user, I want my questions to be answered so that I do not need to contact HR often. | <ul style="list-style-type: none">• Frequently asked questions will be added• Reasonable questions with decent responses |

| | | |
|----|---|---|
| #5 | As a HR employee, I want the To-Do list to show me my completed tasks so I can focus on something else. | <ul style="list-style-type: none"> • Each card of To-Do list will have a checkbox, once ticked it is done • Completed section to show finished tasks |
| #6 | As a HR employee, I want to be able to use the application to check my remaining leaves so that I can plan for a family holiday. | <ul style="list-style-type: none"> • Page that displays leaves from the database • Use firebase |
| #7 | As a HR employee, I want an announcement page so that everyone can be updated for future events. | <ul style="list-style-type: none"> • Create announcement page • Allow new announcements to be added • Firebase will be used to ensure the same announcement can be heard from everyone |
| #8 | As a HR employee, I want to be able to remove my leave so that I can select it again, in the event where I made a wrong submission. | <ul style="list-style-type: none"> • Edit leaves setting • User can remove leave(s) • Database will register the new changes • Make Calendar at Leaves Page |
| #9 | As a HR employee, I want to see the requests I have made on the application so I know if it is pending. | <ul style="list-style-type: none"> • Make a requests page on the application • Allow the user to see if the request is done • Changes reflected in Firebase |

| | | |
|-----|---|---|
| #10 | As a HR executive, I want a FAQ section so that employees do not have to ask me often about something. | <ul style="list-style-type: none"> Request to ask HR executive to list out frequently asked questions Proper responses will be added |
| #11 | As a HR executive, I want to be able to edit or delete my announcements so that any mistakes made, I can edit or remove it. | <ul style="list-style-type: none"> Allow editing and deleting to be done on the announcement page, on the announcement card. Firebase will be updated of the new changes. |
| #12 | As a HR executive, I want to be able to sign in to see my particulars so that I know it is still intact (events etc) each time I login. | <ul style="list-style-type: none"> Test if different account sign in displays different personal information Use Firebase |
| #13 | As a HR executive, I want the application to view my payslip and as well as for the rest of the employees so that they can gauge or check it on their own. | <ul style="list-style-type: none"> Payslip page Get data from Firebase |
| #14 | As a HR executive, I want the logo design to have the same motive as the app, preferably a folder so it looks more professional. | <ul style="list-style-type: none"> Folder with the words Delta 53 has been added as the app logo |
| #15 | As a HR executive, I want the home page calendar of the application, as well as the leaves calendar, to allow time selection for the calendar so that it is flexible for me to choose how long the MC was given and how long my leave is. | <ul style="list-style-type: none"> Allow a dialog to be prompt when the user selects the add button User gets to choose the time and date period. Make Calendar at Home Page Add events in Home Calendar Reflect changes in Firebase |

| | | |
|-----|---|---|
| #16 | As a CEO, I want the design of the application to have some nice aesthetics so that it looks professional and good for the company. | <ul style="list-style-type: none"> • Show login page as it is the first impression for any user |
| #17 | As a CEO, I want a profile page with my profile photo displayed so that people can differentiate me. | <ul style="list-style-type: none"> • User profile page will display profile picture • Get data from Firebase |
| #18 | As a CEO, I want the login page to have a Contact Administrator at the bottom so that it allows the user to know it for sign up purposes. | <ul style="list-style-type: none"> • Contact Administrator will be highlighted and placed below the Sign In button. |
| #19 | As a CEO, I want a navigation page on the top right, so that it is easier to see other functions. | <ul style="list-style-type: none"> • Menu button to be placed on the top right • Able to see other functions of the application |
| #20 | As a CEO, I want to be able to see my attendance so that I can prepare any letter of reason to the HR Executive to explain my absence. | <ul style="list-style-type: none"> • Attendance to be displayed at the payslip page, following the wireframe • Get data from Firebase |
| #21 | As a CEO, I want to see a logout page with a proper logout dialogue so before I forget to check something, I can press cancel. | <ul style="list-style-type: none"> • Allow users to be prompt, if they wish to logout • Able to choose logout or cancel |

Sprint Progress

Sprint 1 (4th August 2021 - 14th August 2021)

Version log for Sprint 1 (#143 - 4th August 2021 to #92 - 14th August 2021)

| | | | | | | | |
|-----|----------|----------|---|---|--|--|--|
| 92 | 4ebbe65 | doubleOZ | monsterdaniel97@gmail.com | Merge branch 'main' of https://github.com/dduck123/Delta-53 | | | |
| 93 | 64bae64 | doubleOZ | monsterdaniel97@gmail.com | bruh | | | |
| 94 | 2.11E+79 | mvaeve | Maeveng25@gmail.com | added working FAQ page | | | |
| 95 | 5ddcd48 | mvaeve | Maeveng25@gmail.com | drawer class | | | |
| 96 | d6d6a97 | mvaeve | Maeveng25@gmail.com | added date timeline package | | | |
| 97 | 649153c | doubleOZ | monsterdaniel97@gmail.com | small visual changes | | | |
| 98 | 08e628e | doubleOZ | monsterdaniel97@gmail.com | added faq tab into app drawer | | | |
| 99 | 2661877 | mvaeve | Maeveng25@gmail.com | new drawer class and work attendance | | | |
| 100 | abd970d | mvaeve | Maeveng25@gmail.com | update linking and removed todo | | | |
| 101 | ebb7f45 | mvaeve | Maeveng25@gmail.com | added icon and linked logout in mainpage.dart | | | |
| 102 | 3f4f2b4 | doubleOZ | monsterdaniel97@gmail.com | fixed text box color | | | |
| 103 | 3fa0a324 | doubleOZ | monsterdaniel97@gmail.com | updated contents inside login_page.dart and constants.dart | | | |
| 104 | 9c3f39e | doubleOZ | monsterdaniel97@gmail.com | merge login_page.dart | | | |
| 105 | 676947b | mvaeve | Maeveng25@gmail.com | main page with drawer | | | |
| 106 | 0b5296e | mvaeve | Maeveng25@gmail.com | main.dart simplified | | | |
| 107 | 14ab634 | Maeve Ng | mjrng002@mymail.sim.edu.sg | cleaned up main.dart | | | |
| 108 | 52d124c | doubleOZ | monsterdaniel97@gmail.com | removed test1.txt | | | |
| 109 | b9cd57a | doubleOZ | monsterdaniel97@gmail.com | remove deltaapp file | | | |
| 110 | 80dea68 | Rhonda | srlim003@mymail.sim.edu.sg | Added full file | | | |
| 111 | 58bd94e | doubleOZ | monsterdaniel97@gmail.com | gone | | | |
| 112 | 41fd493 | Rhonda | srlim003@mymail.sim.edu.sg | Login page added | | | |
| 113 | 869ad9e | ohecc | 88427559+ohecc@users.noreply.github.com | Merge pull request #4 from dduck123/Leonard | | | |
| 114 | 348b32b | ohecc | lyjsoon001@mymail.sim.edu.sg | Create test1.txt | | | |
| 115 | 29ffb52 | ohecc | lyjsoon001@mymail.sim.edu.sg | Create test1.txt | | | |
| 116 | 47f028e | doubleOZ | monsterdaniel97@gmail.com | created app | | | |
| 117 | 75f8641 | doubleOZ | monsterdaniel97@gmail.com | remove homepage.dart | | | |

| | | | | | | | |
|-----|----------|----------------|--|---|--|--|--|
| 118 | 780ceea | doubleOZ | monsterdaniel97@gmail.com | deleted test1.txt and test2.txt | | | |
| 119 | 3d95895 | doubleOZ | monsterdaniel97@gmail.com | deleted delta.txt | | | |
| 120 | 852d2e4 | doubleOZ | monsterdaniel97@gmail.com | added homepage.dart | | | |
| 121 | 4423694 | ohecc | 88427559+ohecc@users.noreply.github.com | Merge pull request #3 from dduck123/Leonard | | | |
| 122 | a588029 | ohecc | lyjsoon001@mymail.sim.edu.sg | Create test2.txt | | | |
| 123 | 3097129 | rizfebriansyah | 88428142+rizfebriansyah@users.noreply.github.com | Merge pull request #2 from dduck123/test | | | |
| 124 | 5b03efd | ohecc | lyjsoon001@mymail.sim.edu.sg | Create test1.txt | | | |
| 125 | 4fd7d238 | ohecc | lyjsoon001@mymail.sim.edu.sg | Merge branch 'main' of https://github.com/dduck123/Delta-53 into main | | | |
| 126 | 0f3384c | ohecc | lyjsoon001@mymail.sim.edu.sg | Revert "Create test.txt" | | | |
| 127 | 651ae68 | doubleOZ | monsterdaniel97@gmail.com | Deleted hi.txt and wassup.txt | | | |
| 128 | 7e964d5 | doubleOZ | monsterdaniel97@gmail.com | Deleted deltamaeve.txt from repository | | | |
| 129 | 4a4a825 | doubleOZ | monsterdaniel97@gmail.com | Deleted dan.txt from repository | | | |
| 130 | 655e18a | ohecc | lyjsoon001@mymail.sim.edu.sg | Create test.txt | | | |
| 131 | d8cf13e | ohecc | lyjsoon001@mymail.sim.edu.sg | Merge branch 'main' of https://github.com/dduck123/Delta-53 into main | | | |
| 132 | 0242b19 | ohecc | lyjsoon001@mymail.sim.edu.sg | Revert "Create githubdesktest.txt" | | | |
| 133 | 029ae4d | ohecc | 88427559+ohecc@users.noreply.github.com | Merge pull request #1 from dduck123/test | | | |
| 134 | 6f6cb35 | ohecc | lyjsoon001@mymail.sim.edu.sg | Create githubdesktest.txt | | | |
| 135 | 7a5117a | ohecc | lyjsoon001@mymail.sim.edu.sg | Revert "Create test.txt" | | | |
| 136 | 7f755fd | ohecc | lyjsoon001@mymail.sim.edu.sg | Create test.txt | | | |
| 137 | ebd9164 | rizfebriansyah | rizqi980202@gmail.com | Added riz file | | | |
| 138 | 826a39a | doubleOZ | monsterdaniel97@gmail.com | dan file | | | |
| 139 | ad4d726 | ohecc | lyjsoon001@mymail.sim.edu.sg | Merge branch 'main' of https://github.com/dduck123/Delta-53 | | | |
| 140 | db0ee99 | ohecc | lyjsoon001@mymail.sim.edu.sg | Create wassup.txt | | | |
| 141 | 067bf03 | Maeve Ng | mjrng002@mymail.sim.edu.sg | added maeve file | | | |
| 142 | f3f78ad | Rhonda | srlim003@mymail.sim.edu.sg | Added file | | | |
| 143 | f9f37de | dduck123 | 77720833+dduck123@users.noreply.github.com | Initial commit | | | |

Sprint 1 Tasks

User Story (Sprint 1) ...

complete in-progress
incomplete closed

Sprint Color Guide

incomplete

4. As a user, I want my questions to be answered so that I do not need to contact HR often.

incomplete

16. As a CEO, I want the design of the application to have some nice aesthetics so that it looks professional and good for the company.

incomplete

18. As a CEO, I want the login page to have a Contact Administrator at the bottom so that it allows the user to know it for sign up purposes.

incomplete

19. As a CEO, I want a navigation page on the top right, so that it is easier to see other functions.

+ Add a card

□ ☰

Planning of Sprint 1:

The team gathered to commence the 1st Sprint. We have discussed that we will try out Github and Flutter for the application. We requested to try a few user requests first and received the following from above as tests from the users. Our plan was to finish the 4 chosen user requests in 10 days. We decided to go with color codes ranging from green, yellow, red and blue. Each color represents the state of the user story.

We made the ground rules for the process of tackling each of the user stories so that everyone is clear.

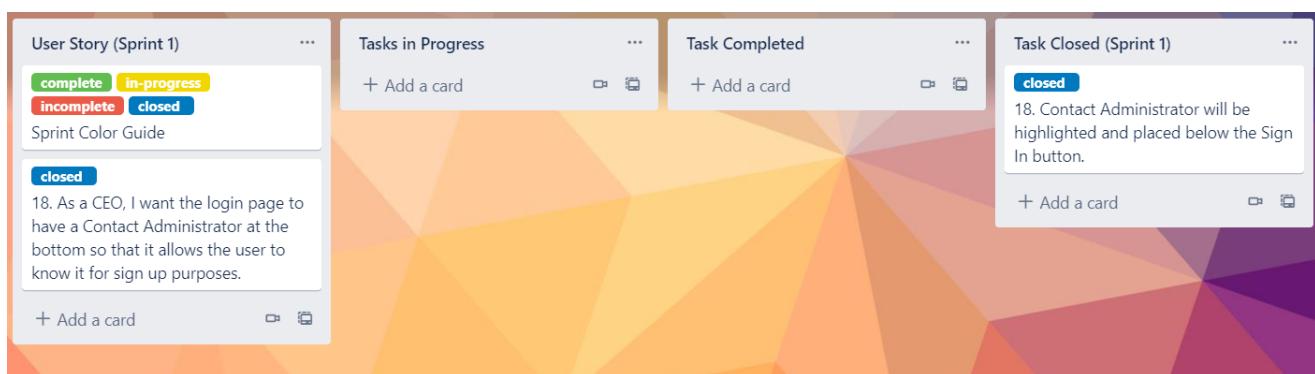
- Any in-progress user stories left will be sent back to the Backlog list.
- In-Progress user stories will be tackled again at the start of the sprint.
- Requirements that are completed on the day itself will go to the Task Completed section. If all are finished immediately with no trailing requirements, then it will go to the Task Closed section automatically.
- If a user story requirement is completed on the day itself, it will be in the Task Completed list first, then Task Closed in the next.

We then decided how long the first sprint should be. In 10 days, we aim to finish the following - a login page on 11th August, a user design interface on 12th August, a Frequently Asked Questions (FAQ) plus Navigation page 13th August.

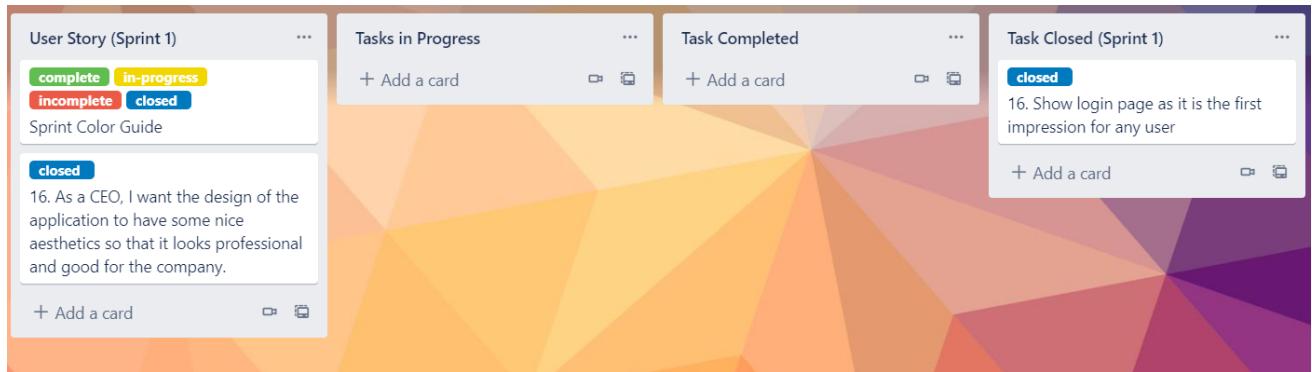
The following backlogs show our progress for the 1st Sprint.

Sprint 1 Backlogs

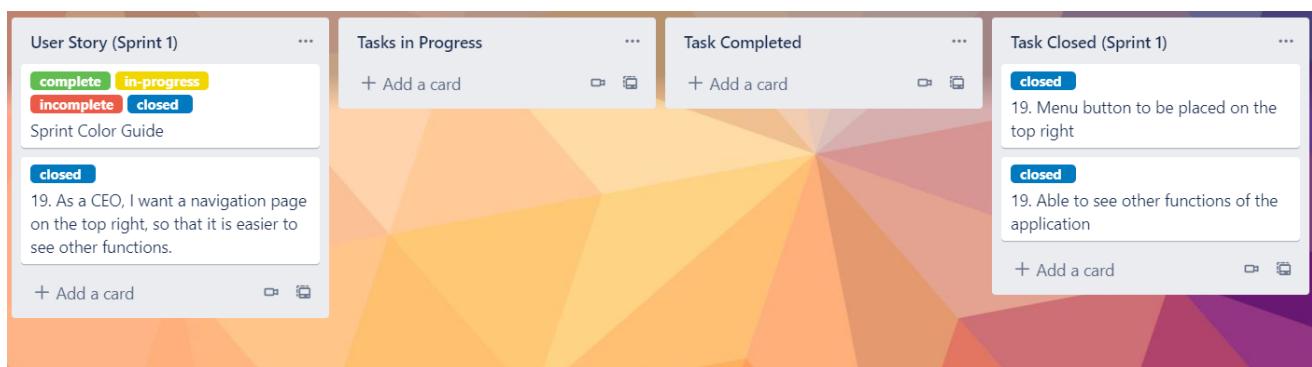
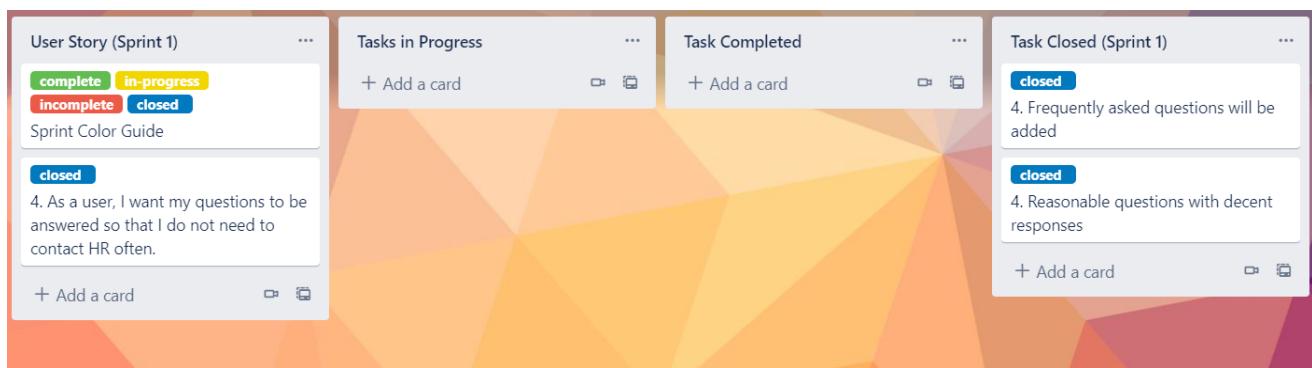
11th August 2021

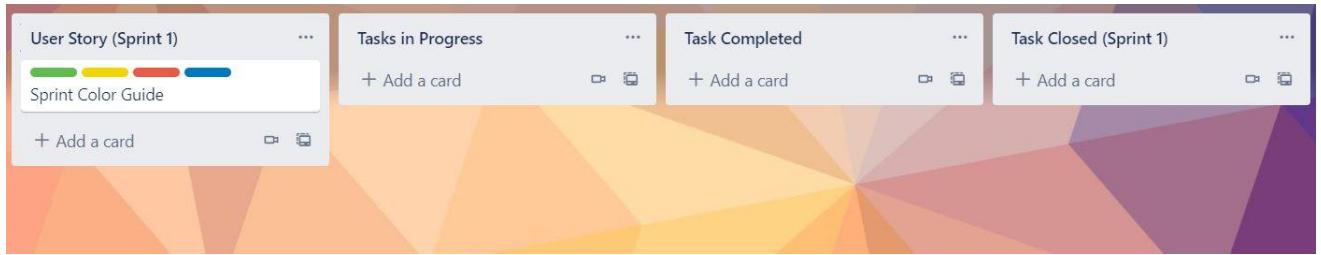


12th August 2021



13th August 2021





Finished Work:

A decent app has been made but it still does not have the full functions. It is not ready to be deployed as a fully functional application despite having a navigational page and login page.

Sprint Review:

Meeting was held with our stakeholders in regards to the progress and result of our first sprint. Generally, they felt that it was acceptable and happy with how the application looks.

Some of the new requirements they provided were that they would like to see more application functions, preferably a page with a calendar. They looked forward to seeing the results at the end of the 2nd Sprint.

Sprint Retrospective:

The team have discussed and agreed that the first sprint went well, there was no need for daily scrum for this sprint. The only issue faced was trying out Github and making sure we could add our files in. It took the team awhile to get used to it but everything went well in the end. Some of our team members tried alternatives like Github Desktop as it felt easier to them. Thereafter, we will try to add more functionalities as requested by the stakeholders.

Sprint 2 (15th August 2021 - 27th August 2021)

Version log for Sprint 2 (#91 - 15th August 2021 to #54 - 27th August 2021)

| | | | | | | | |
|----|---------|------------|----------------------------|--|--|--|--|
| 54 | f2f9ab3 | Dan Miller | monsterdaniel97@gmail.com | slight more changes to the layout of stats.dart | | | |
| 55 | f6df70f | Dan Miller | monsterdaniel97@gmail.com | more layout changes to the Attendance and Leave widgets | | | |
| 56 | b9ef170 | Dan Miller | monsterdaniel97@gmail.com | more changes to payslip and leaves tracking | | | |
| 57 | 709d6f5 | Dan Miller | monsterdaniel97@gmail.com | small changes to payslip widget | | | |
| 58 | 442841d | Dan Miller | monsterdaniel97@gmail.com | added firebase dependencies | | | |
| 59 | b0db7a9 | Dan Miller | monsterdaniel97@gmail.com | Merge branch 'main' of https://github.com/dduck123/Delta-53 | | | |
| 60 | 3d3ede6 | Dan Miller | monsterdaniel97@gmail.com | in the progress of creating the layout for the payslip and leave on the stats.dart | | | |
| 61 | d94261a | mvaeve | Maeveng25@gmail.com | calendar up | | | |
| 62 | 3a123c6 | Rhonda | srlim003@mymail.sim.edu.sg | Calender | | | |
| 63 | e75f285 | Rhonda | srlim003@mymail.sim.edu.sg | Calendar issues | | | |
| 64 | d28a92d | mvaeve | Maeveng25@gmail.com | change calendar code | | | |
| 65 | 1fd034a | mvaeve | Maeveng25@gmail.com | calendar | | | |
| 66 | 3485722 | Dan Miller | monsterdaniel97@gmail.com | added appdrawer to the main page | | | |

| | | | | | | | |
|----|---------|----------------|---|---|--|--|--|
| 67 | 661977e | mvaeve | Maeveng25@gmail.com | debugged calendar | | | |
| 68 | 5aa86c4 | mvaeve | Maeveng25@gmail.com | work on calendar in main | | | |
| 69 | 1224b4a | Rhonda | srlim003@mymail.sim.edu.sg | Calendar | | | |
| 70 | fb0cffc | Dan Miller | monsterdaniel97@gmail.com | small changes to drawer_nav.dart | | | |
| 71 | 02d48ae | Dan Miller | monsterdaniel97@gmail.com | changes to drawer_nav.dart | | | |
| 72 | fe690d9 | mvaeve | Maeveng25@gmail.com | small edits | | | |
| 73 | c81e051 | mvaeve | Maeveng25@gmail.com | work on the database | | | |
| 74 | 1e5aee4 | mvaeve | Maeveng25@gmail.com | annoucement upz | | | |
| 75 | 67ad030 | Rhonda | srlim003@mymail.sim.edu.sg | Adding database code | | | |
| 76 | d3fea02 | mvaeve | Maeveng25@gmail.com | clean up announcement page | | | |
| 77 | 170a6a0 | Dan Miller | monsterdaniel97@gmail.com | added drawer to caclendar page | | | |
| 78 | 4d2680c | Dan Miller | monsterdaniel97@gmail.com | Merge branch 'main' of https://github.com/dduck123/Delta-53 | | | |
| 79 | 2255318 | Dan Miller | monsterdaniel97@gmail.com | updated login if else statements | | | |
| 80 | 05ef9f2 | rizfebriansyah | rizqi980202@gmail.com | calendar widget | | | |
| 81 | ab00bb1 | Dan Miller | monsterdaniel97@gmail.com | added logout_page.dart and logout dialogue | | | |
| 82 | 6553e8d | mvaeve | Maeveng25@gmail.com | edited annoucment and add pop up menu | | | |
| 83 | c5598e5 | mvaeve | Maeveng25@gmail.com | create announcement page | | | |
| 84 | cdc9163 | mvaeve | Maeveng25@gmail.com | login page | | | |
| 85 | 3b8859d | Dan Miller | monsterdaniel97@gmail.com | updated package names to correspond with file names | | | |
| 86 | ecdee3b | Dan Miller | monsterdaniel97@gmail.com | removed flutter.db | | | |
| 87 | db89241 | Dan Miller | monsterdaniel97@gmail.com | added a db file | | | |
| 88 | 94b9515 | mvaeve | Maeveng25@gmail.com | add employee db | | | |
| 89 | 7cfa587 | ohecc | 88427559+ohecc@users.noreply.github.com | Merge pull request #5 from dduck123/LeonardTest | | | |
| 90 | 9f7ccb3 | ohecc | lyjsoon001@mymail.sim.edu.sg | Update mainpage.dart | | | |
| 91 | 442b4f1 | Dan Miller | monsterdaniel97@gmail.com | enter in some new entries into the faq section | | | |

Sprint 2 Tasks

Sprint 2

...
complete in-progress
incomplete closed
Sprint Color Guide

incomplete
3. As a user, I want a page that is colourful, or something different from the application so that it feels more vibrant and fun to use.

incomplete
6. As a HR employee, I want to be able to use the application to check my remaining leaves so that I can plan for a family holiday.

incomplete
7. As a HR employee, I want an announcement page so that everyone can be updated for future events.

incomplete
8. As a HR employee, I want to be able to remove my leave so that I can select it again, in the event where I made a wrong submission.

incomplete
11. As a HR executive, I want to be able to edit or delete my announcements so that any mistakes made, I can edit or remove it.

+ Add a card ⋮ ...

Sprint 2

...
incomplete
12. As a HR executive, I want to be able to sign in to see my particulars so that I know it is still intact (events etc) each time I login.

incomplete
13. As a HR executive, I want the application to view my payslip and as well as for the rest of the employees so that they can gauge or check it on their own.

incomplete
15. As a HR executive, I want the home page calendar of the application, as well as the leaves calendar, to allow time selection for the calendar so that it is flexible for me to choose how long the MC was given and how long my leave is.

incomplete
20. As a CEO, I want to be able to see my attendance so that I can prepare any letter of reason to the HR Executive to explain my absence.

incomplete
21. As a CEO, I want to see a logout page with a proper logout dialogue so before I forget to check something, I can press cancel.

+ Add a card ⋮ ...

Planning of Sprint 2:

As requested by the stakeholders, we added more functions to accomplish for our 2nd Sprint. We discussed which functions to start off with first and then continue to stick with the usual progression. Since we are required to work on more functions, we requested for a slightly longer period of time which is 12 days. Ultimately, the team did their best to finish most of the functions - mainly the user interface of the application.

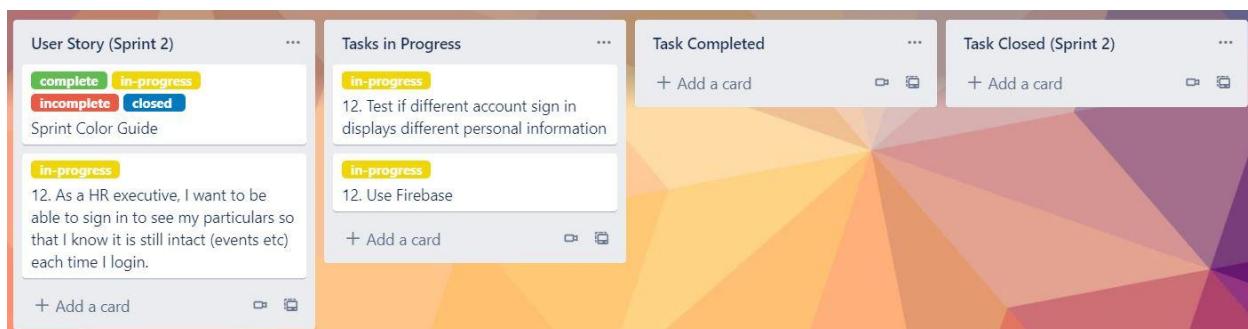
What we accomplished for Sprint 2 was the following:

1. The FAQ has been made with proper questions.
2. Logout page and a logout dialogue appears to prompt the user if they want to logout.
3. An announcement page is made and is displayed in grid view.
4. Payslip functionality has also been added.
5. A Calendar has been added to the home page for the user to use but it lacks the function to be stored in the database, similar to the request page.

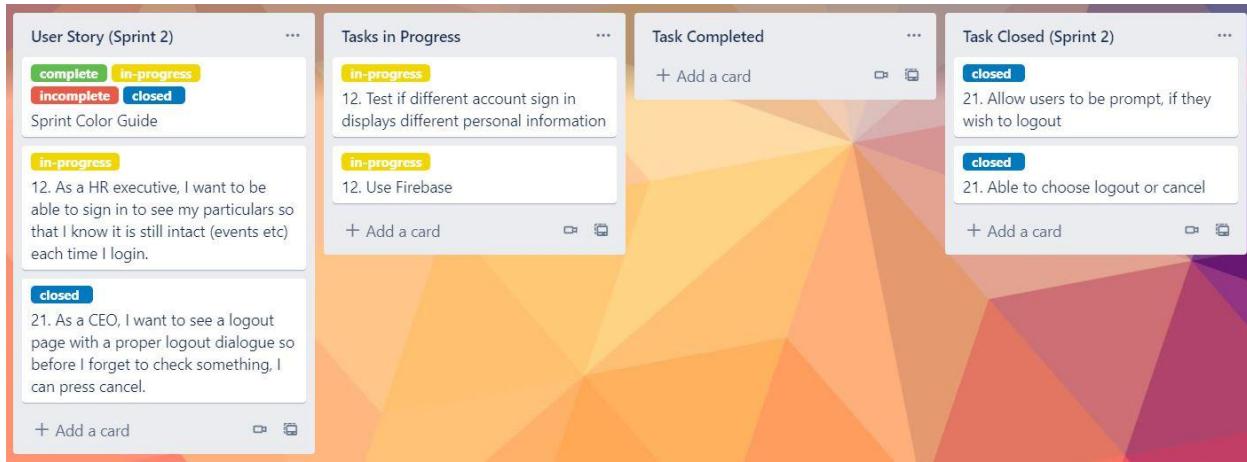
The dates above of each image below will show which functions have been done successfully and also review what functions are left to be implemented.

Sprint 2 Backlogs

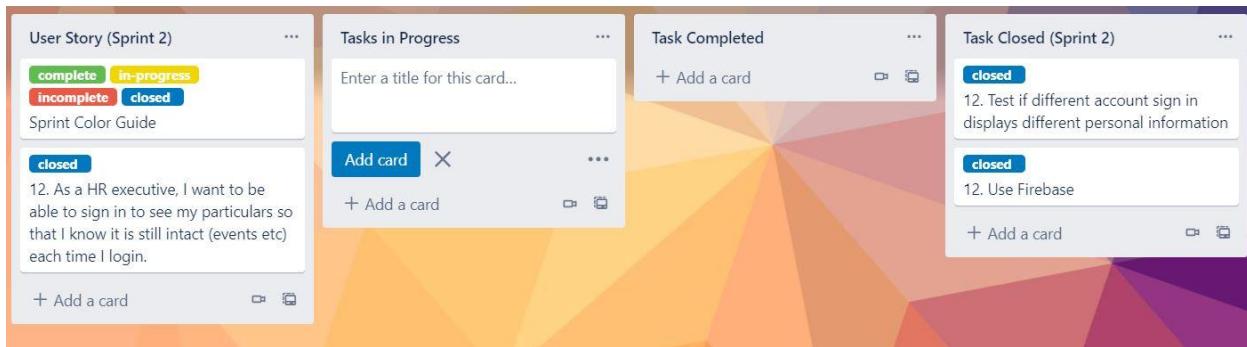
16th August 2021



17th August 2021



18th August 2021

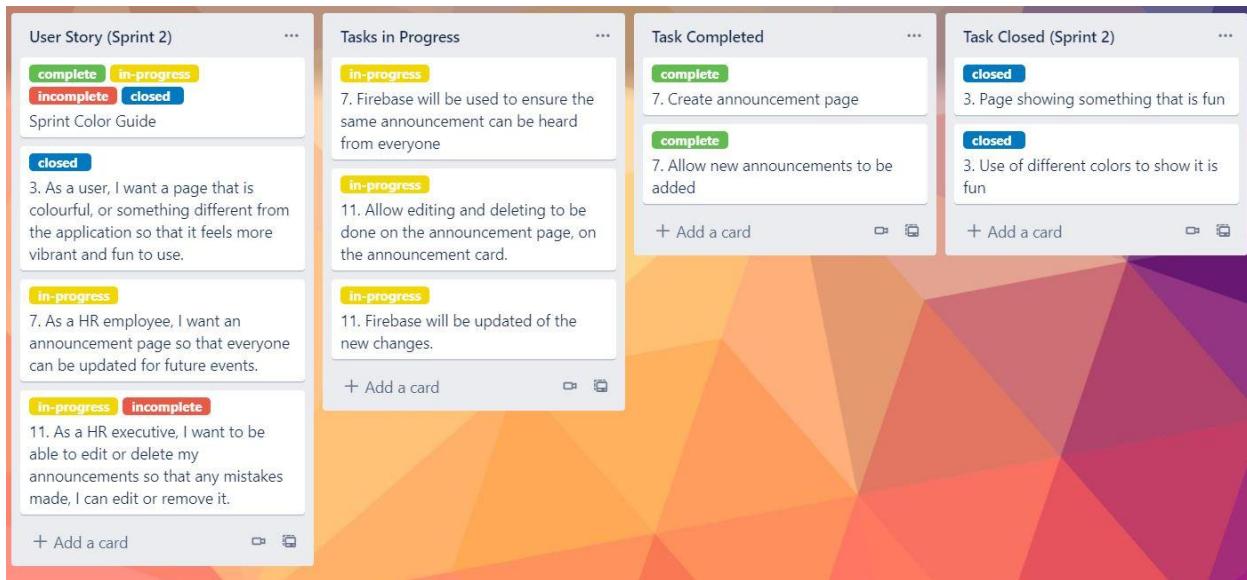


Daily Scrum 18th August 2021:

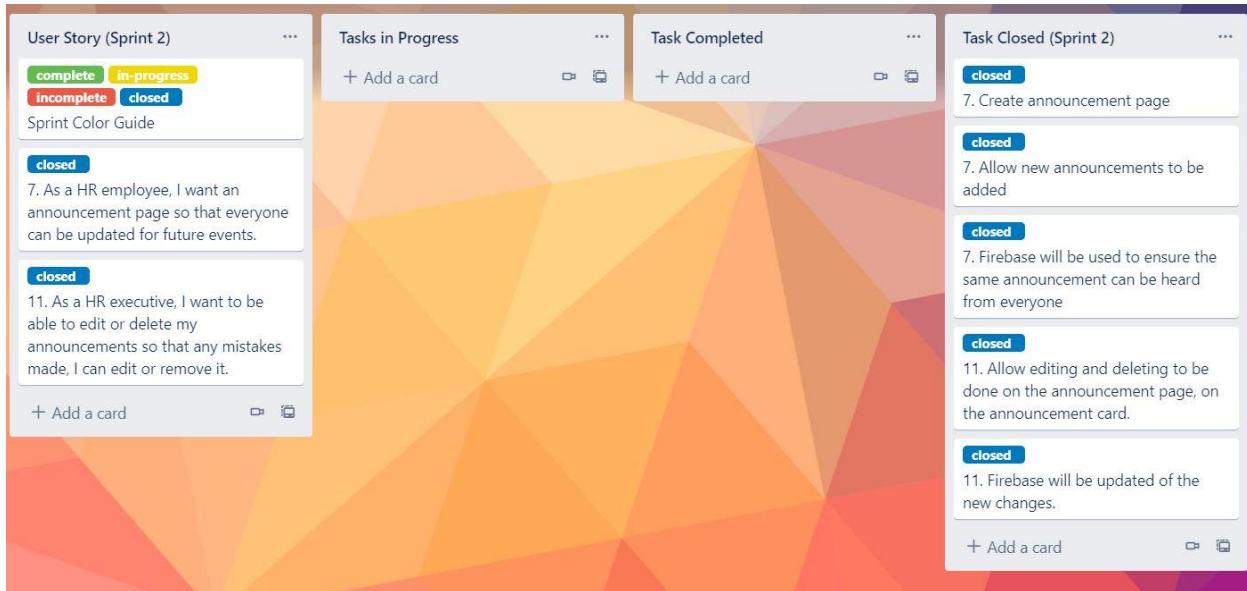
The team had a stand-up to check on the progress of everyone, and to see if anyone had any difficulties and how we can overcome it. Everyone was okay with the progress made so far, and encountered no difficulties. We then discussed the choice of database to use. The stand-up took about fifteen minutes, with each teammate giving their opinion about the recommended databases. We desired something that was simple but effective, as well as tutorials that are easy to follow. After much decision between Firebase and SQLite, we decided to go with Firebase as our choice of database platform. Firebase was really flexible due to its multiple forms of authentication and compatibility with mobile emulators and web explorer.

We later received a request from the stakeholders, indicating that they would like to add in another feature. This feature mentioned was the To-Do list. We were told that by adding the To-Do list, the To-Do list gives users the opportunity to ensure their tasks were all in one place. This promotes responsibility for the users and helps them to keep them on track. The designers immediately went to get user feedback and get some low wireframe done.

19th August 2021



20th August 2021



23rd August 2021

User Story (Sprint 2)

complete in-progress
incomplete closed

Sprint Color Guide

in-progress:

8. As a HR employee, I want to be able to remove my leave so that I can select it again, in the event where I made a wrong submission.

in-progress:

15. As a HR executive, I want the home page calendar of the application, as well as the leaves calendar, to allow time selection for the calendar so that it is flexible for me to choose how long the MC was given and how long my leave is.

+ Add a card

Tasks in Progress

in-progress
in-progress
in-progress
in-progress
in-progress

8. Make Calendar at Leaves Page
8. Edit leaves setting
8. User can remove leave(s)
8. Firebase will register the new changes
15. Allow a dialog to be prompt when the user selects the add button

+ Add a card

Task Completed

complete

15. Make Calendar at Home Page

+ Add a card

Task Closed (Sprint 2)

+ Add a card

24th August 2021

User Story (Sprint 2)

complete in-progress
incomplete closed

Sprint Color Guide

in-progress:

8. As a HR employee, I want to be able to remove my leave so that I can select it again, in the event where I made a wrong submission.

in-progress:

15. As a HR executive, I want the home page calendar of the application, as well as the leaves calendar, to allow time selection for the calendar so that it is flexible for me to choose how long the MC was given and how long my leave is.

+ Add a card

Tasks in Progress

in-progress
in-progress
in-progress
in-progress
in-progress

8. Make Calendar at Leaves Page
8. Edit leaves setting
8. User can remove leave(s)
8. Firebase will register the new changes
15. Allow a dialog to be prompt when the user selects the add button

+ Add a card

Task Completed

complete

15. Add events at Home Calendar

+ Add a card

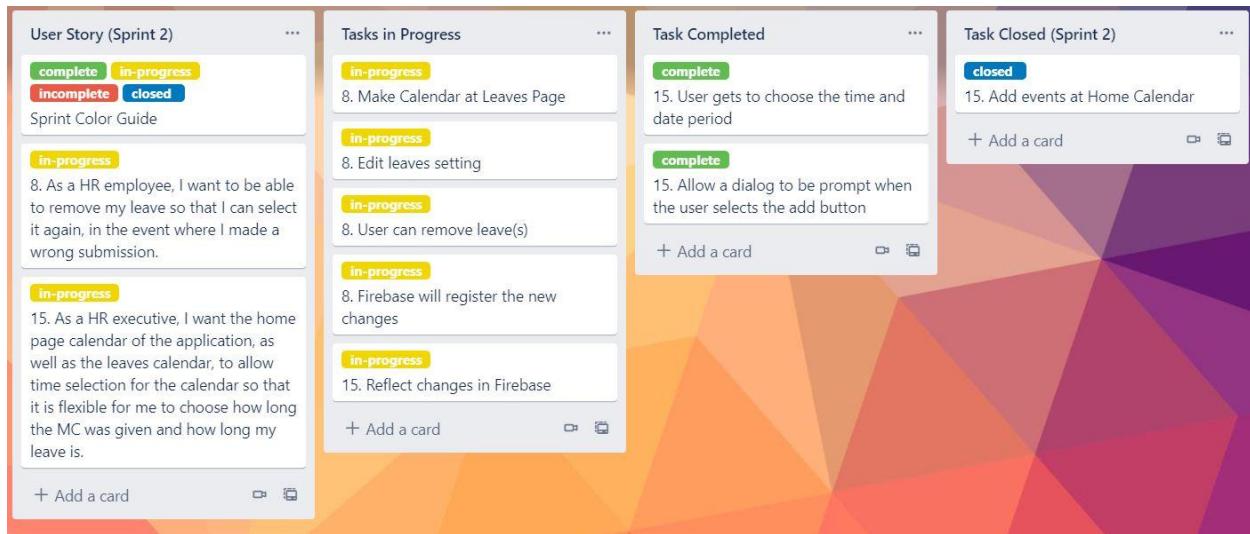
Task Closed (Sprint 2)

closed

15. Make Calendar at Home Page

+ Add a card

25th August 2021



Daily Scrum 25th August 2021:

We held another daily scrum to check on our progress. The stand-up was held mainly because the Tasks in Progress was piling up. This prompted us to come together and discuss it. Overall the meeting was held for about fifteen minutes, with teammates voicing out their difficulties faced in regards to Firebase. It was revealed that some teammates struggled to grasp the idea of the Firebase despite tutorials shown. There are many ways to display the information via Flutter and Firebase. Some teammates were more or less confused with the various methods shown, thus resulting in difficulty towards understanding the usage of Flutter and Firebase together. An example was that some tutorials use classes or using different packages or extensions by Flutter.

Debugging was also an issue with Firebase, where most of the time spent was to figure out what was the cause for the following errors. Usually, after looking through online for answers, it eventually led to more questions being asked. The team tried to solve the issue by writing down the list of errors that they have faced and tried to see if anyone has solved the issue before. Any issues that were solved were eventually written down.

In regards to the To-Do list, the team will try to host 2 daily scrums per sprint to ensure that there is consistency to the development of the To-Do list. As a week has passed, the team has an approved wireframe after a few edits. We are ready to show the stakeholders our results and get ready to present to them to have an approval, in order to get the new function into development with the other functions on sprint review.

26th August 2021

User Story (Sprint 2)

- complete**
- in progress**
- incomplete**
- closed**

Sprint Color Guide

in-progress:

- As a HR employee, I want to be able to remove my leave so that I can select it again, in the event where I made a wrong submission.
- As a HR executive, I want the home page calendar of the application, as well as the leaves calendar, to allow time selection for the calendar so that it is flexible for me to choose how long the MC was given and how long my leave is.
- As a HR executive, I want the application to view my payslip and as well as for the rest of the employees so that they can gauge or check it on their own.
- As a HR employee, I want to be able to use the application to check my remaining leaves so that I can plan for a family holiday.
- As a CEO, I want to be able to see my attendance so that I can prepare any letter of reason to the HR Executive to explain my absence.

+ Add a card

Tasks in Progress

- in-progress**

8. Make Calendar at Leaves Page
8. Edit leaves setting
8. User can remove leave(s)
8. Firebase will register the new changes
15. Reflect changes in Firebase
13. Get data from Firebase

+ Add a card

Task Completed

+ Add a card

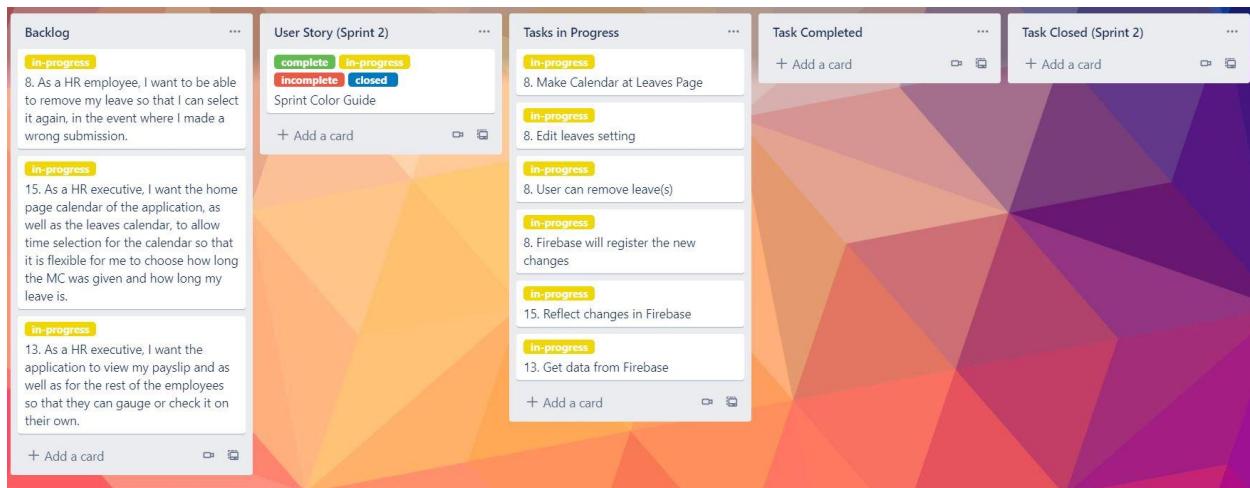
Task Closed (Sprint 2)

- closed**

6. Use firebase
6. Page that displays leaves from the database
13. Payslip page
15. User gets to choose the time and date period
15. Allow a dialog to be prompt when the user selects the add button
20. Attendance to be displayed at the payslip page, following the wireframe
20. Get data from Firebase

+ Add a card

27th August 2021



Finished Work:

The app is really coming along in terms of production. There are much more features now as compared to before. User interface is generally done with most pages up while the technical aspects need some work.

Sprint 2 Review:

Meeting was held with our stakeholders in regards to the progress and result of our second sprint. They were happier with how the application looks and we are grateful for that. We assured the stakeholders by the next meeting that we would have finished the application.

They provided us with a new requirement - since the user interface is more or less complete, the technical aspects are what they hope to see for the next sprint. They hoped that the database would be up and running the next time they arrived for the meeting. We showed the To-Do list to the stakeholders and they are happy with what they saw and how it addresses their needs. They were impressed that we came up with something that looks pretty much completed in a short span of time. They look forward to seeing it in the application.

Sprint Retrospective:

The team have discussed and agreed that the second sprint was somewhat okay. We reviewed the remaining cards on the Trello board to find three in-progress user stories and six in Tasks in Progress. We told ourselves to buck up for this so we would not lose out in the next meeting. The good thing about Sprint 2 was finishing up the pages, with some able to display information from Firebase. The obvious challenge for this sprint was Firebase for sure, everyone had a different error to debug and the time spent on watching tutorials and testing different codes was excruciating for the team. Our objective for the next sprint was to finish the application as requested by the stakeholders.

Sprint 3 (28th August 2021 - 7th September 2021)

Version log for Sprint 3 (#53 - 28th August 2021 to #01 - 7th September 2021)

| | | | | | | | | |
|----|----------|----------------|---|--|--|--|--|--|
| 1 | 0fd63e | Dan Miller | monsterdaniel97@gmail.com | fixed up a few visual elements | | | | |
| 2 | ecc0023 | Dan Miller | monsterdaniel97@gmail.com | deleted hero_dialog_template file | | | | |
| 3 | 33019a6 | Dan Miller | monsterdaniel97@gmail.com | fleshed out the scrolling on the announcement messages | | | | |
| 4 | e374ae3 | Dan Miller | monsterdaniel97@gmail.com | changed to appear visually to have it all show the same theme colour and fixing the announcement text bottom overflow issue and allowing scrolling | | | | |
| 5 | 513ae02b | rizfebriansyah | rizqi980202@gmail.com | changed overall colour to indigo | | | | |
| 6 | 8fcdf69 | rizfebriansyah | rizqi980202@gmail.com | add new files to the todo folder with firebase | | | | |
| 7 | e7f11c6 | rizfebriansyah | rizqi980202@gmail.com | deleted the previous todo folder | | | | |
| 8 | 7c994d3 | rizfebriansyah | rizqi980202@gmail.com | added todo.dart file under services folder | | | | |
| 9 | 6f41ef1 | rizfebriansyah | rizqi980202@gmail.com | added services folder | | | | |
| 10 | 16b6537 | Maeve Ng | maeve@Maeves-MacBook-Pro.local | finalisation | | | | |
| 11 | 770ed54 | Maeve Ng | maeve@Maeves-MacBook-Pro.local | finalise | | | | |
| 12 | ee2d090 | Maeve Ng | maeve@Maeves-MacBook-Pro.local | clean up | | | | |
| 13 | 472e7fa | Maeve Ng | maeve@Maeves-MacBook-Pro.local | tidy up linking between leaves and requests | | | | |
| 14 | 9fe934c | Dan Miller | monsterdaniel97@gmail.com | removed the payslip tab due to redundancy | | | | |
| 15 | 2bc8711 | Dan Miller | monsterdaniel97@gmail.com | added firebase auth to leave_editing_page.dart | | | | |
| 16 | effc6a1 | Dan Miller | monsterdaniel97@gmail.com | fixed overflow text in the announcement cards | | | | |
| 17 | f94bcf | rizfebriansyah | rizqi980202@gmail.com | Update leave_editing_page.dart | | | | |
| 18 | 57d8bb | ohecc | 88427559+ohecc@users.noreply.github.com | Merge pull request #9 from dduck123/leo | | | | |
| 19 | 1ddc682 | ohecc | lyjsoon001@mymail.sim.edu.sg | Update faq_page.dart | | | | |
| 20 | 91ada9 | rizfebriansyah | rizqi980202@gmail.com | Merge branch 'main' of https://github.com/dduck123/Delta-53 into main | | | | |
| 21 | dd6f91c | Dan Miller | monsterdaniel97@gmail.com | fixed overflow issues on the leaves and attendance tab as well as the personal data tab | | | | |
| 22 | 3e08ef7 | Dan Miller | monsterdaniel97@gmail.com | Merge branch 'main' of https://github.com/dduck123/Delta-53 | | | | |
| 23 | 0340fae | rizfebriansyah | rizqi980202@gmail.com | added a todo page | | | | |
| 24 | 1a11c67 | Maeve Ng | maeve@Maeves-MacBook-Pro.local | connect request page to leaves page as well as firebase | | | | |
| 25 | 5f50c75 | Maeve Ng | maeve@Maeves-MacBook-Pro.local | connect leaves page to database and make changes to calendar in leaves page | | | | |
| 26 | d511d7f | Dan Miller | monsterdaniel97@gmail.com | Merge branch 'main' of https://github.com/dduck123/Delta-53 | | | | |
| 27 | 7b31a76 | Rhonda | srlim003@mymail.sim.edu.sg | Employee page | | | | |

| | | | | | | | | |
|----|----------|----------------|---|---|--|--|--|--|
| 28 | 5e8683f | Dan Miller | monsterdaniel97@gmail.com | ada | | | | |
| 29 | 48779fc | ohecc | 88427559+ohecc@users.noreply.github.com | merge pull req | | | | |
| 30 | 24e41d6 | ohecc | lyjsoon001@mymail.sim.edu.sg | Update ic_launcher.png | | | | |
| 31 | d11b737 | ohecc | lyjsoon001@mymail.sim.edu.sg | Update ic_launcher.png | | | | |
| 32 | f996dad | ohecc | lyjsoon001@mymail.sim.edu.sg | Update ic_launcher.png | | | | |
| 33 | 64d3780 | ohecc | lyjsoon001@mymail.sim.edu.sg | Update ic_launcher.png | | | | |
| 34 | 2f1218f | ohecc | lyjsoon001@mymail.sim.edu.sg | Update ic_launcher.png | | | | |
| 35 | 224268a | ohecc | 88427559+ohecc@users.noreply.github.com | Merge pull request #7 from dduck123/Leo1 | | | | |
| 36 | a27486d | ohecc | lyjsoon001@mymail.sim.edu.sg | Revert "Update stats_page.dart" | | | | |
| 37 | a2aebb9 | ohecc | lyjsoon001@mymail.sim.edu.sg | Update stats_page.dart | | | | |
| 38 | aaea802 | Dan Miller | monsterdaniel97@gmail.com | added and linked leaves and attendance | | | | |
| 39 | 6acbfb13 | Dan Miller | monsterdaniel97@gmail.com | completed the p; leaves and attendance linked with database | | | | |
| 40 | 8f60e87 | ohecc | 88427559+ohecc@users.noreply.github.com | Merge pull request #6 from dduck123/Req | | | | |
| 41 | 86fb186 | ohecc | lyjsoon001@mymail.sim.edu.sg | Update drawer_navigation.dart | | | | |
| 42 | 7256000 | ohecc | lyjsoon001@mymail.sim.edu.sg | Create requests_page.dart | | | | |
| 43 | df21913 | Maeve Ng | maeve@Maeves-MacBook-Pro.local | clean up code | | | | |
| 44 | 4fd8769 | Dan Miller | monsterdaniel97@gmail.com | added use of database in payslip | | | | |
| 45 | dfedc8f | Dan Miller | monsterdaniel97@gmail.com | retrieval of database for the payslip added | | | | |
| 46 | 32c0b1d | Dan Miller | monsterdaniel97@gmail.com | removed databasemanagement.dart | | | | |
| 47 | 612bc4a | Dan Miller | monsterdaniel97@gmail.com | Merge branch 'main' of https://github.com/dduck123/Delta-53 | | | | |
| 48 | 17e7abe | rizfebriansyah | rizqi980202@gmail.com | updated directory | | | | |
| 49 | 7020cc2 | rizfebriansyah | rizqi980202@gmail.com | leave calendar | | | | |
| 50 | 4298c3f | Maeve Ng | maeve@Maeves-MacBook-Pro.local | design personal data | | | | |
| 51 | 45e7c30 | Maeve Ng | maeve@Maeves-MacBook-Pro.local | work on work stats personal data | | | | |
| 52 | dab1aaaf | Maeve Ng | maeve@Maeves-MacBook-Pro.local | personal data | | | | |
| 53 | 29d6099 | Maeve Ng | maeve@Maeves-MacBook-Pro.local | personal data | | | | |

Sprint 3 Tasks

Sprint 3 ...

complete **in-progress**
incomplete **closed**

Sprint Color Guide

in-progress

8. As a HR employee, I want to be able to remove my leave so that I can select it again, in the event where I made a wrong submission.

in-progress

13. As a HR executive, I want the application to view my payslip and as well as for the rest of the employees so that they can gauge or check it on their own.

in-progress

15. As a HR executive, I want the home page calendar of the application, as well as the leaves calendar, to allow time selection for the calendar so that it is flexible for me to choose how long the MC was given and how long my leave is.

+ Add a card ⌂ ⌂

Sprint 3 ...

incomplete

1. As a user, I want to have a To-do list so that I can see my tasks to do for the day

incomplete

2. As a user, I want my profile to be seen on the application so I can see my personal details.

incomplete

5. As a user, I want the To-Do list to show me my completed tasks so I can focus on something else.

incomplete

9. As a HR employee, I want to see the requests I have made on the application so I know if it is pending.

incomplete

10. As a HR executive, I want a FAQ section so that employees do not have to ask me often about something.

incomplete

14. As a HR executive, I want the logo design to have the same motive as the app, preferably a folder so it looks more professional.

incomplete

17. As a CEO, I want a profile page with my profile photo displayed so that people can differentiate me.

+ Add a card ⌂ ⌂

Planning of Sprint 3:

For the final sprint, we ensured that the firebase is set up correctly. As requested by the stakeholders, we will finalise the functionality of our application. We gave ourselves 10 days to finish the application to manage our time better. Reviewing our discussions, for the To-Do list, we presented the low to high wireframes for the stakeholders and then came up with two different designs for the To-Do list. The stakeholders also chose the design that matches the requirements they requested. After finishing the process, the team went on to work on this sprint.

The goal for this sprint, as we have discussed, is to properly connect our application to the firebase and add the new implemented features. We managed to finish the application in 8 days. All functions were able to access our Firebase and the To-Do list is up and running. Any later adjustments were completely minor, such as enhancing the visuals of the application. The dates above of each image will show which functions have been done properly and the daily scrums.

Sprint 3 Backlogs

29th August 2021

| User Story (Sprint 3) | Tasks in Progress | Task Completed | Task Closed (Sprint 3) |
|---|---|---|------------------------|
| <p>complete in-progress incomplete closed</p> <p>Sprint Color Guide</p> <p>in-progress 2. As a user, I want my profile to be seen on the application so I can see my personal details.</p> <p>in-progress 17. As a CEO, I want a profile page with my profile photo displayed so that people can differentiate me.</p> <p>in-progress 8. As a HR employee, I want to be able to remove my leave so that I can select it again, in the event where I made a wrong submission.</p> <p>in-progress 13. As a HR executive, I want the application to view my payslip and as well as for the rest of the employees so that they can gauge or check it on their own.</p> <p>in-progress 15. As a HR executive, I want the home page calendar of the application, as well as the leaves calendar, to allow time selection for the calendar so that it is flexible for me to choose how long the MC was given and how long my leave is.</p> <p>+ Add a card</p> | <p>in-progress 2. Display personal details such as birthday</p> <p>in-progress 8. Make Calendar at Leaves Page</p> <p>in-progress 8. Edit leaves setting</p> <p>in-progress 8. User can remove leave(s)</p> <p>in-progress 8. Firebase will register the new changes</p> <p>in-progress 13. Get data from Firebase</p> <p>in-progress 15. Reflect changes in Firebase</p> <p>in-progress 17. User profile page will display profile picture</p> <p>in-progress 17. Get data from Firebase</p> <p>+ Add a card</p> | <p>complete</p> <p>2. Make a user profile section or page</p> <p>+ Add a card</p> | <p>+ Add a card</p> |

30th August 2021

The image shows a digital whiteboard interface with four main sections arranged horizontally:

- User Story (Sprint 3)**: Contains cards for "Sprint Color Guide" (closed), "As a user, I want my profile to be seen on the application so I can see my personal details." (in-progress), "As a CEO, I want a profile page with my profile photo displayed so that people can differentiate me." (in-progress), "As a HR employee, I want to be able to remove my leave so that I can select it again, in the event where I made a wrong submission." (in-progress), "As a HR executive, I want the application to view my payslip and as well as for the rest of the employees so that they can gauge or check it on their own." (in-progress), and "As a HR executive, I want the home page calendar of the application, as well as the leaves calendar, to allow time selection for the calendar so that it is flexible for me to choose how long the MC was given and how long my leave is." (in-progress).
- Tasks in Progress**: Contains cards for "Display personal details such as birthday" (in-progress), "User can remove leave(s)" (in-progress), "Firebase will register the new changes" (in-progress), "Get data from Firebase" (in-progress), "Reflect changes in Firebase" (in-progress), "User profile page will display profile picture" (in-progress), and "Get data from Firebase" (in-progress).
- Task Completed**: Contains cards for "Make Calendar at Leaves Page" (complete) and "Edit leaves setting" (complete).
- Task Closed (Sprint 3)**: Contains cards for "Make a user profile section or page" (closed) and a placeholder "+ Add a card".

At the bottom of each column, there is a "+ Add a card" button and a trash bin icon.

2nd September 2021

User Story (Sprint 3)

- complete**
- in-progress**
- closed**

Sprint Color Guide

in-progress

- As a user, I want my profile to be seen on the application so I can see my personal details.
- As a CEO, I want a profile page with my profile photo displayed so that people can differentiate me.
- As a HR executive, I want the application to view my payslip and as well as for the rest of the employees so that they can gauge or check it on their own.
- As a HR executive, I want the home page calendar of the application, as well as the leaves calendar, to allow time selection for the calendar so that it is flexible for me to choose how long the MC was given and how long my leave is.

+ Add a card

Tasks in Progress

- in-progress**

- Display personal details such as birthday
- User can remove leave(s)
- Firebase will register the new changes
- Reflect changes in Firebase
- User profile page will display profile picture
- Get data from Firebase

+ Add a card

Task Completed

- complete**

- Get data from Firebase

+ Add a card

Task Closed (Sprint 3)

- closed**

- Make Calendar at Leaves Page
- Edit leaves setting

+ Add a card

3rd September 2021

User Story (Sprint 3)

- complete**
- in-progress**
- closed**

Sprint Color Guide

in-progress

- As a user, I want to have a To-do list so that I can see my tasks to do for the day
- As a HR employee, I want to see the requests I have made on the application so I know if it is pending
- As a CEO, I want a profile page with my profile photo displayed so that people can differentiate me
- As a HR employee, I want to be able to remove my leave so that I can select it again, in the event where I made a wrong submission
- As a HR executive, I want the home page calendar of the application, as well as the leaves calendar, to allow time selection for the calendar so that it is flexible for me to choose how long the MC was given and how long my leave is

+ Add a card

User Story (Sprint 3)

- complete**

2. As a user, I want my profile to be seen on the application so I can see my personal details.

closed

10. As a HR executive, I want a FAQ section so that employees do not have to ask me often about something.

closed

14. As a HR executive, I want the logo design to have the same motive as the app, preferably a folder so it looks more professional.

closed

13. As a HR executive, I want the application to view my payslip and as well as for the rest of the employees so that they can gauge or check it on their own.

+ Add a card

Tasks in Progress

- in-progress**

- User can remove leave(s)
- Firebase will register the new changes
- Reflect changes in Firebase
- User profile page will display profile picture
- Get data from Firebase
- Allow the user to see if the request is done
- Changes reflected in Firebase

+ Add a card

Task Completed

- complete**

- Make a requests page on the application
- Make To-Do list page
- List out work required for the day
- Add To-Do list if user wish to edit

+ Add a card

Task Closed (Sprint 3)

- closed**

- Get data from Firebase
- Folder with the words Delta 53 has been added as the app logo
- Request to ask HR executive to list out frequently asked questions
- Proper responses will be added

+ Add a card

Daily Scrum 3rd September 2021:

We made progressive steps in completing our application but there were still some problems in troubleshooting the firebase.

4th September 2021

The board displays the following sections and their contents:

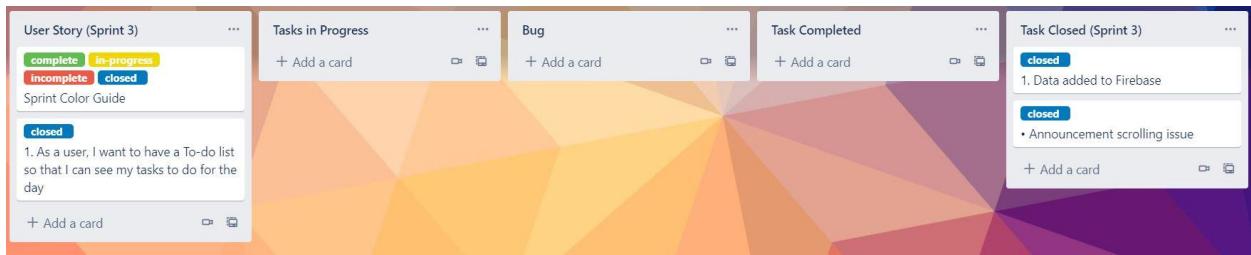
- User Story (Sprint 3)**:
 - complete, in-progress, incomplete, closed
 - Sprint Color Guide
 - 1. As a user, I want to have a To-do list so that I can see my tasks to do for the day
 - closed
 - 2. As a user, I want my profile to be seen on the application so I can see my personal details.
 - closed
 - 5. As a user, I want the To-Do list to show me my completed tasks so I can focus on something else.
 - + Add a card
- User Story (Sprint 3)**:
 - complete
 - 9. As a HR employee, I want to see the requests I have made on the application so I know if it is pending.
 - complete
 - 17. As a CEO, I want a profile page with my profile photo displayed so that people can differentiate me.
 - complete
 - 8. As a HR employee, I want to be able to remove my leave so that I can select it again, in the event where I made a wrong submission.
 - complete
 - 15. As a HR executive, I want the home page calendar of the application, as well as the leaves calendar, to allow time selection for the calendar so that it is flexible for me to choose how long the MC was given and how long my leave is.
 - + Add a card
- Tasks in Progress**:
 - + Add a card
- Bug**:
 - incomplete
 - Overflow Error for Payslip page
 - + Add a card
- Task Completed**:
 - complete
 - 8. User can remove leave(s)
 - complete
 - 8. Firebase will register the new changes
 - complete
 - 9. Allow the user to see if the request is done
 - complete
 - 9. Changes reflected in Firebase
 - complete
 - 15. Reflect changes in Firebase
 - complete
 - 17. Get data from Firebase
 - complete
 - 17. User profile page will display profile picture
 - + Add a card
- Task Closed (Sprint 3)**:
 - closed
 - 2. Display personal details such as birthday
 - closed
 - 9. Make a requests page on the application
 - closed
 - 1. Make To-Do list page
 - closed
 - 1. List out work required for the day
 - closed
 - 1. Add To-Do list if user wish to edit
 - closed
 - 5. Each card of To-Do list will have a checkbox, once ticked it is done
 - closed
 - 5. Completed section to show finished tasks
 - + Add a card

5th September 2021

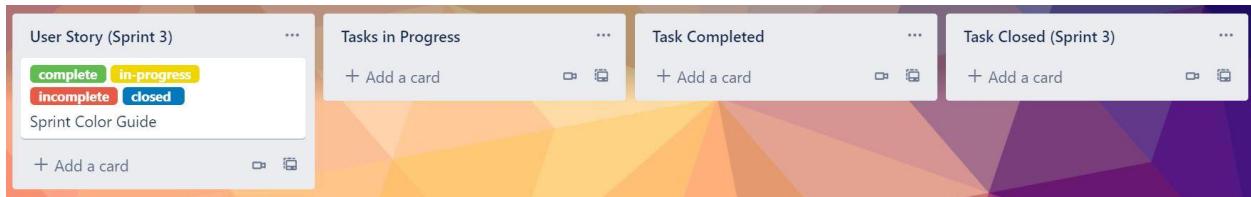
The board displays the following sections and their contents:

- User Story (Sprint 3)**:
 - complete, in-progress, incomplete, closed
 - Sprint Color Guide
 - 1. As a user, I want to have a To-do list so that I can see my tasks to do for the day
 - closed
 - 9. As a HR employee, I want to see the requests I have made on the application so I know if it is pending.
 - closed
 - 17. As a CEO, I want a profile page with my profile photo displayed so that people can differentiate me.
 - closed
 - 8. As a HR employee, I want to be able to remove my leave so that I can select it again, in the event where I made a wrong submission.
 - closed
 - 15. As a HR executive, I want the home page calendar of the application, as well as the leaves calendar, to allow time selection for the calendar so that it is flexible for me to choose how long the MC was given and how long my leave is.
 - + Add a card
- Tasks in Progress**:
 - + Add a card
- Bug**:
 - incomplete
 - Announcement scrolling issue
 - + Add a card
- Task Completed**:
 - complete
 - 1. Data added to Firebase
 - + Add a card
- Task Closed (Sprint 3)**:
 - closed
 - Overflow Error for Payslip page
 - closed
 - 2. As a user, I want my profile to be seen on the application so I can see my personal details.
 - closed
 - 5. As a user, I want the To-Do list to show me my completed tasks so I can focus on something else.
 - closed
 - 8. User can remove leave(s)
 - closed
 - 8. Firebase will register the new changes
 - closed
 - 9. Allow the user to see if the request is done
 - closed
 - 9. Changes reflected in Firebase
 - closed
 - 15. Reflect changes in Firebase
 - closed
 - 17. Get data from Firebase
 - closed
 - 17. User profile page will display profile picture
 - + Add a card

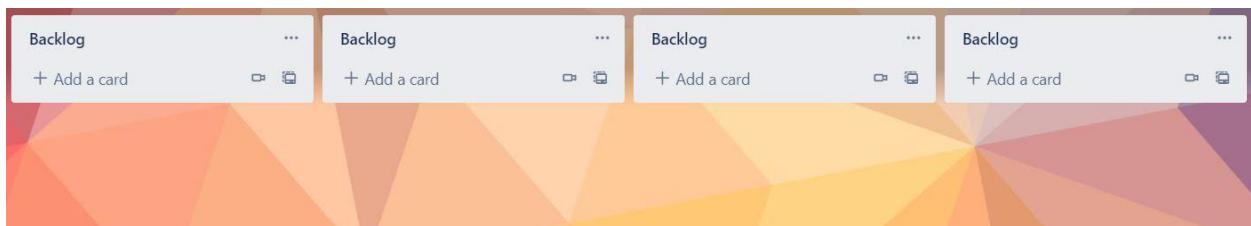
6th September 2021



7th September 2021



Backlogs Remaining:



Finished Work:

The application is complete and ready to be rolled out in the app store. The functions are all working and we are happy with how it turned out in the end.

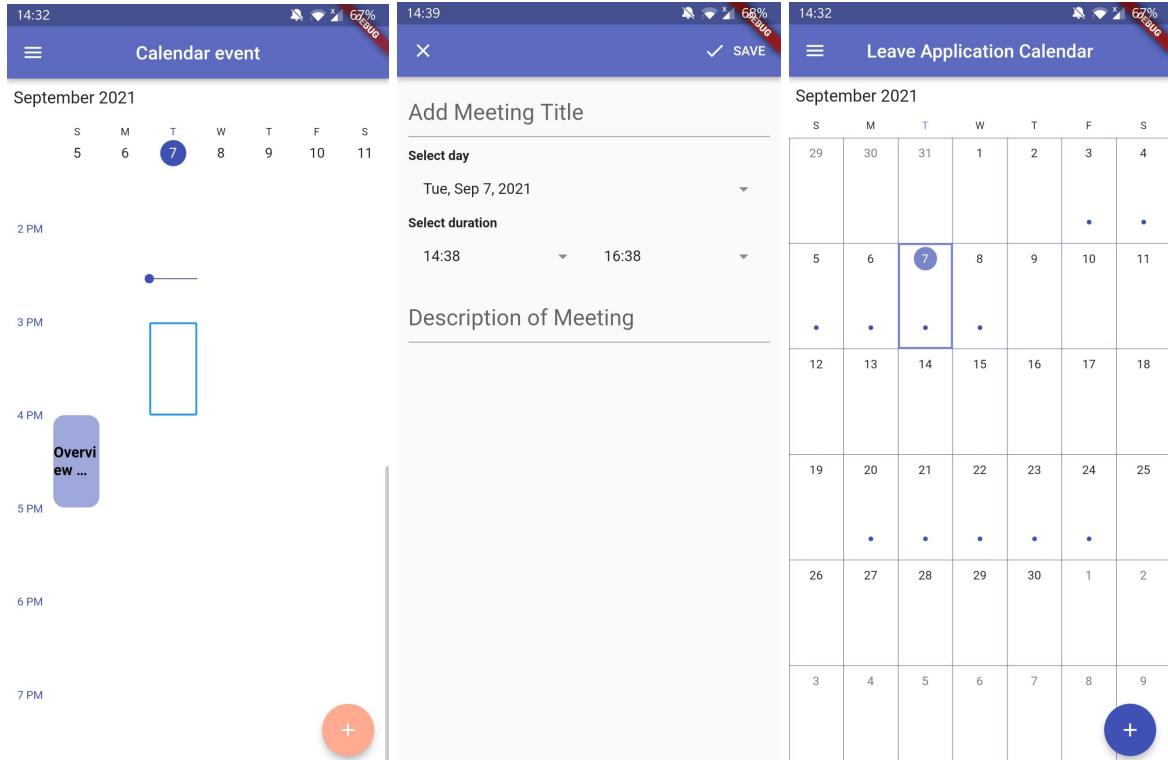
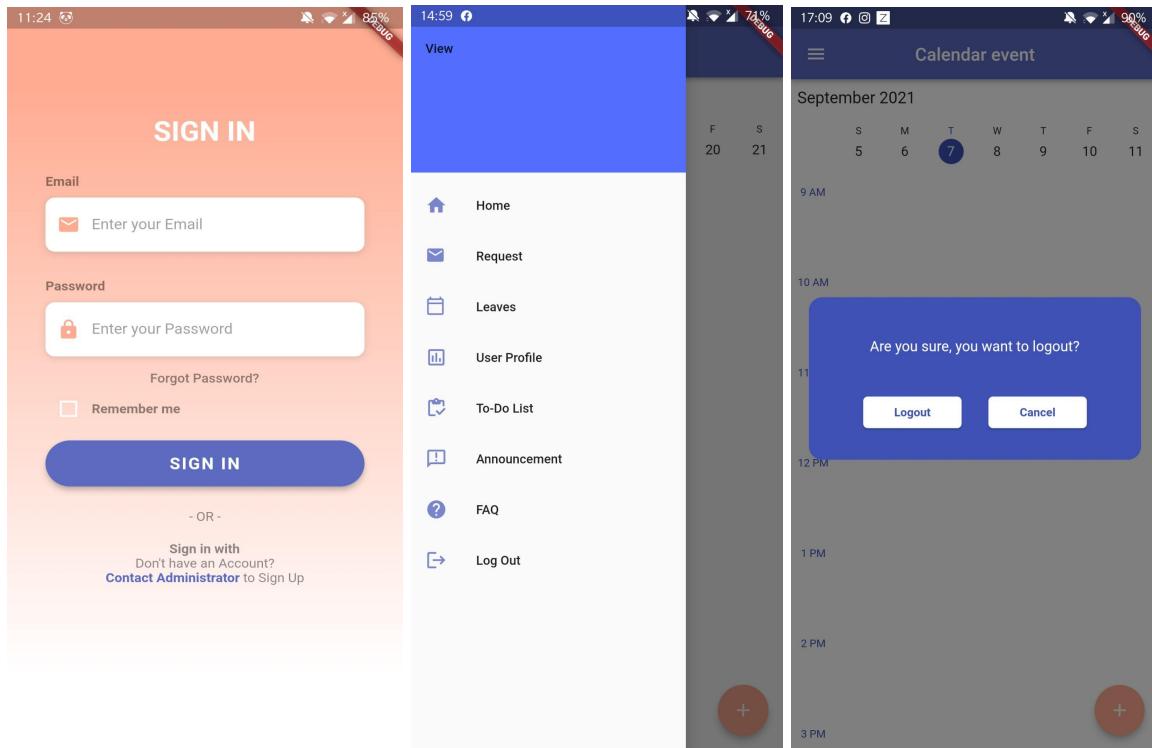
Sprint 3 Review:

Meeting was held with our stakeholders in regards to the progress and result of our final sprint. They are deeply impressed that we finished all the functions they required within 8 days instead of the given 10 days. The stakeholders praised us for our efforts and are very happy with the application.

Sprint Retrospective:

The team discussed the progress and are with how everything turned out. Right after the daily scrum on 3rd September, we managed to connect Firebase with the application. We gave ourselves 10 days for this sprint but managed to finish it in 8 days. We used the remaining 2 days to touch up the application. Issues were all resolved (mainly connecting to Firebase) and there were no user stories left to do unless specified by the stakeholders.

Results of Sprints



14:37

X ✓ SUBMIT

Type of Leave Request (Annual, ...)

FROM

Tue, Sep 7, 2021

TO

Tue, Sep 7, 2021

Description of Leave

15:00

Leave Application Calendar

September 2021

| S | M | T | W | T | F | S |
|----|----|----|----|----|----|----|
| 29 | 30 | 31 | 1 | 2 | 3 | 4 |
| 5 | 6 | 7 | 8 | 9 | 10 | 11 |
| .. | .. | .. | .. | .. | .. | .. |

7 Tuesday

| 9 AM | 10 AM | 11 AM | 12 PM | 1 PM | 2 PM | 3 PM |
|------|-------|-------|-------|------|------|------------------|
| | | | | | | Compensatory off |

14:33

Announcement

IMPORTANT :

Dear All, We are holding a meeting on 13th September at 2:00pm to discuss our progress on our client's crypto application. Your presence is essential. We plan to go

RE: October Research meeting :

We will hold our monthly research meeting in the conference room at 11:00 a.m. on Friday, 16th October. David and Jane will report on the meeting with the Marketing

Lunch Treat :

Good day Everyone, After all our hard work in the past few months, the company has decided to treat the entire department out for lunch. This is to extend their gratitude and

CHANGE OF POLICY :

We would like to bring to your notice the recent changes made by the Management in policy related with employees coming late at work. This change will take effect from

14:33

FAQ

Contact support

- How do I make an account? →
- Who do I notify to change password? →

How to use

- Does the app allow me to see my leaves and pay? Or do I need to ask the HR Executive for that matter? →
- How to use the Announcement page? →
- Oh no! I wish to edit or delete the announcement! Is that possible? →
- What does the calendar at the Home page do? →

14:34

← How do I make an account?

Press the Contact Administrator at the bottom of the login page, the HR Executive will be notified to assist you!

14:32

Requests

- Sick
Leave status: Pending
- Business Trip
Leave status: Pending
- Sick relative
Leave status: Pending
- Compensatory off
Leave status: Pending

Requests

- Sick
Leave status: Pending
- Business Trip
Leave status: Pending
- Sick relative
Leave status: Pending

Business Trip
2021-09-20 - 2021-09-24
Going to New Zealand to sign a contract and meeting
OK

Todo List

- Meeting Arrangements**
Set up a meeting with the team regarding Project Binance.
 - Define aims and objectives
- What we need to achieve
- Delegate tasks
- Crypto Trend Analysis
- Review Survey**
Review previous day survey responses, and credit each account by end of today. Come up with a conclusion.
- Work Orders**
 Ensure all work orders are posted and completed.

Do analysis regarding
 Todos ✓ Completed

Todo List

Review Survey
Review previous day survey responses, and credit each account by end of today. Come up with a conclusion.

Task completed
 Todos ✓ Completed

Profile

Work Stats Personal data

LEAVES
25/30

ATTENDANCE
79.18%

Payslip
\$2800.00

Profile

Work Stats Personal data


David Toh
Position
Boss

Personal Information

User ID: zEfB25UlcdMqZPg6u9iEatmav0T2
Email: david.toh@delta53.com
Full Name: David Toh
Date of birth: 20 June 1970
Address: 51 Bull Street S812633
Phone No: 98334510

Summative Review of Project:

We requested feedback from the stakeholders to understand how they felt about our progress and review. This turned out to be a big success. They were pleased with how we tackled issues and challenges that rose along the way as it showed that we could adapt well - for example, adding new features along the way as per their request within a short time frame.

The stakeholders mentioned that the overall design of the mobile application was very appealing and user-friendly. In the design page templates, all key aspects are easily accessible. Material is neatly organised using white space, visual components and alignment. The navigation bar is properly identified, regularly arranged, and allows the user to quickly go from one page to another page flawlessly.

Lastly, the stakeholders mentioned that this would not be possible if we did not have good communication skills among our team members. We delegated tasks according to each members' individual strengths and made sure everyone was on the right track. Our scrum meeting conversations were generally handled in a manner that did not disrupt others and were largely focused on the project and what is required to complete the task at hand.

Overall, the stakeholders were highly satisfied with our application as it fulfilled all their requirements.

Technologies and External Packages used:

External Packages: Dart, Flutter, Syncfusion

Database used: Firebase

The Core Components of the System:

1. Authentication System (Log in)

Below image is the code for the login function which calls the firebase authentication to ensure that the user enters the correct login details. If successful, the user will be able to login and the application will navigate to the main page of the application, in our case, calling the Calendar MainPage class.

```
//Log in function
} _login() async {
{
  setState(() {
    isInProgress = true;
  });
  final status = await FirebaseAuthHelper().login(
    email: myControllerEmail.text.trim(),
    pass: myControllerPass.text.trim());
  setState(() {
    isInProgress = false;
  });
  if (status == AuthResultStatus.successful) {
    Navigator.of(context).pushReplacement(
      MaterialPageRoute(builder: (context) => Calendar MainPage()));
  } else {
}
```

In the case where the user enters the wrong login details, an alertDialog will pop up to notify the user of the error message returned based on what the user actions are.

```
} else {
  final errorMsg = AuthExceptionHandler.generateExceptionMessage(status);

  showDialog(
    context: context,
    builder: (BuildContext context) => AlertDialog(
      backgroundColor: Colors.indigo,
      title: Center(
        child: Text('Alert',
          style: TextStyle(
            color: Colors.white,
            fontSize: 24,
            fontWeight: FontWeight.bold,
          )), // TextStyle, Text, Center
      content: Row(
        mainAxisAlignment: MainAxisAlignment.center,
        crossAxisAlignment: CrossAxisAlignment.center,
        children: <Widget>[
          Expanded(
            child: Text(
              errorMsg,
              textAlign: TextAlign.center,
              style: TextStyle(
                color: Colors.white,
```

Below are the codes where we handle the exception for different case scenarios and generate the right error message to alert and guide the users.

```
class AuthExceptionHandler {
    static handleException(e) {
        print(e.code);
        var status;
        switch (e.code) {
            case "invalid-email":
                status = AuthResultStatus.invalidEmail;
                break;
            case "wrong-password":
                status = AuthResultStatus.wrongPassword;
                break;
            case "weak-password":
                status = AuthResultStatus.weakPassword;
                break;
            case "user-not-found":
                status = AuthResultStatus.userNotFound;
                break;
            case "user-disabled":
                status = AuthResultStatus.userDisabled;
                break;
            case "email-already-in-use":
                status = AuthResultStatus.emailAlreadyExists;
                break;
            case "unknown":
                status = AuthResultStatus.nothing;
                break;
            default:
                status = AuthResultStatus.undefined;
        }
        return status;
    }

    static generateExceptionMessage(exceptionCode) {
        String errorMessage;
        switch (exceptionCode) {
            case AuthResultStatus.invalidEmail:
                errorMessage = "Your email address appears to be malformed.";
                break;
            case AuthResultStatus.wrongPassword:
                errorMessage = "Your password is wrong.";
                break;
            case AuthResultStatus.weakPassword:
                errorMessage = "Your password is too weak. It must have at least 6 characters.";
                break;
            case AuthResultStatus.userNotFound:
                errorMessage = "User with this email doesn't exist.";
                break;
            case AuthResultStatus.userDisabled:
                errorMessage = "User with this email has been disabled.";
                break;
            case AuthResultStatus.emailAlreadyExists:
                errorMessage =
                    "The email has already been registered. Please login or reset your password.";
                break;
            case AuthResultStatus.nothing:
                errorMessage =
                    "Enter your details correctly.";
                break;
            default:
                errorMessage = "Please try again.";
        }
        return errorMessage;
    }
}
```

2. Mainpage (Work timeline)

After successful user authentication login, the Calendar MainPage class is called. This sets the EventProvider as the context and calls the class MainPage to show the calendar as well as the floating button to add events to the calendar timeline.

```
class Calendar MainPage extends StatelessWidget {
    static final String title = "Calendar event";

    @override
    Widget build(BuildContext context) => ChangeNotifierProvider(
        create: (context) => EventProvider(),
        child: MaterialApp(
            debugShowCheckedModeBanner: false,
            title: title,
            theme: ThemeData(
                brightness: Brightness.light,
                primaryColor: Colors.indigo[300],
            ), // ThemeData
            home: MainPage(),
        ), // MaterialApp
    ); // ChangeNotifierProvider
}
```

MainPage:

```
class MainPage extends StatelessWidget {
    Widget build(BuildContext context) => Scaffold(
        appBar: AppBar(
            title: Text(Calendar MainPage.title),
            centerTitle: true,
        ), // AppBar
        body: CalendarWidget(),
        floatingActionButton: FloatingActionButton(
            child: Icon(
                Icons.add,
                color: Colors.white,
            ), // Icon
            backgroundColor: Colors.indigo,
            onPressed: () => Navigator.of(context).push(
                MaterialPageRoute(builder: (context) => EventEditingPage())),
        ),
        drawer: MyDrawer(),
    ); // Scaffold
}
```

For adding and editing events and displaying them onto the calendar timeline in the main page, we successfully made use of firebase to achieve CRUD on our data. The eventEditingPage is where we create and update events to be stored to firebase. The eventViewingPage as well as CalendarWidget is where we read the data from firebase.

Getting user id and connecting to the right collection in firebase:

```
final currentUserID = FirebaseAuth.instance.currentUser!.uid;  
CollectionReference employees =  
  FirebaseFirestore.instance.collection('Employees');
```

Meeting object:

```
//To create a event inside the app  
class Meeting {  
  final String title;  
  final String description;  
  final DateTime from;  
  final DateTime to;  
  final Color backgroundColor;  
  final bool isAllDay;  
  
  const Meeting({  
    required this.title,  
    required this.description,  
    required this.from,  
    required this.to,  
    this.backgroundColor = Colors.indigo,  
    this.isAllDay = false,  
  });  
}
```

Creating Meeting object to be submitted into firebase:

```
final event = Meeting(  
  title: titleController.text,  
  description: descriptionController.text,  
  from: fromDate,  
  to: toDate,  
  isAllDay: false,  
)
```

Adding events to firebase in EventEditingPage class:

```
provider.addEvent(event);
int index = provider.getIndexEvent(event);
employees
    .doc(currentUserID)
    .collection("DayEvents")
    .doc(index.toString())
    .set(
{
    'title': titleController.text,
    'description': descriptionController.text,
    'from': fromDate,
    'to': toDate
},
SetOptions(merge: true),
)
.then((value) => print("Day event Added"))
.catchError((error) => print("Failed to add day event: $error"));
```

Editing events in firebase based on selected event in EventEditingPage class:

```
final isEditing = widget.meeting != null;
final provider = Provider.of<EventProvider>(context, listen: false);

if (isEditing) {
    int index = provider.getIndexEventOld(event, widget.meeting!);
    provider.editEvent(event, widget.meeting!);
    employees
        .doc(currentUserID)
        .collection("DayEvents")
        .doc(index.toString())
        .update({
            'title': titleController.text,
            'description': descriptionController.text,
            'from': fromDate,
            'to': toDate
        })
        .then((value) => print("Day event Updated"))
        .catchError((error) => print("Failed to update day event: $error"));
}
```

Deleting events in firebase based on selected event in EventViewingPage class:

```
IconButton(
  icon: Icon(Icons.delete),
  onPressed: () {
    final provider = Provider.of<EventProvider>(context, listen: false);
    provider.deleteEvent(event);
    int index = provider.getIndexEvent(event);
    provider.deleteEvent(event);
    employees
      .doc(currentUserID)
      .collection("DayEvents")
      .doc(index.toString())
      .delete()
      .then((value) => print("Event Deleted"))
      .catchError((error) => print("Failed to delete event: $error"));
    Navigator.of(context).pop();
```

In eventProvider, we can get the events and by calling functions in it and store them locally.

```
class EventProvider extends ChangeNotifier{
  final List<Meeting> _events = [];

  List<Meeting> get events => _events;

  DateTime _selectedDate = DateTime.now();
  DateTime get selectedDate => _selectedDate;

  void setDate(DateTime date) => _selectedDate = date;

  List<Meeting> get eventsOfSelectedDate => _events;
```

Since this EventProvider list of events will be wiped every time the application does a hot restart, we made it such that we will access the firebase to retrieve any events stored in it and append it to the list List<Meeting> _events. In the meantime, we also re-index the document id in firebase in the case of any deleted event so as to keep the events in order.

In calendarWidget class, store firebase events into _events:

```
final provider = Provider.of<EventProvider>(context);

if (provider.events.isEmpty) {
    print("Connected to database for main page event calendar");
    employees
        .doc(currentUserID)
        .collection("DayEvents")
        .get()
        .then((QuerySnapshot querySnapshot) {
            int counter = 0;
            querySnapshot.docs.forEach((doc) {
                DateTime myDateTimeTo = doc["to"].toDate();
                DateTime myDateTimeFrom = doc["from"].toDate();
                final event = Meeting(
                    title: doc["title"],
                    description: doc["description"],
                    from: myDateTimeFrom,
                    to: myDateTimeTo,
                    isAllDay: false,
                );
                if (counter == int.parse(doc.id)) {
                    employees
                        .doc(currentUserID)
                        .collection("DayEvents")
                        .doc(counter.toString())
                        .set({
                            "title": doc["title"],
                            "description": doc["description"],
                            "from": myDateTimeFrom,
                            "to": myDateTimeTo,
                        });
                }
            });
        });
}
```

Re-indexing of document ID:

```
    } else {
        employees
            .doc(currentUserID)
            .collection("DayEvents")
            .doc(counter.toString())
            .set({
                "title": doc["title"],
                "description": doc["description"],
                "from": myDateTimeFrom,
                "to": myDateTimeTo,
            });
        doc.reference.delete();
    }

    provider.events.add(event);
    counter += 1;
});
```

3. User Profile (Work statistic and personal information)

In the User Profile page, we made two tabs for viewing information related to the login user. The first tab is Work Stats and the other is Personal Data. The stats class is called which will call both Leaves class and Employee class.

In stats class, the two other classes are called:

```
tabs: [
  Tab(
    icon: Icon(Icons.list_alt_outlined),
    text: "Work Stats",
  ), // Tab
  Tab(
    icon: Icon(Icons.person),
    text: "Personal data",
  ), // Tab
],
), // TabBar
title: const Text('Profile'),
), // AppBar
body: const TabBarView(
  children: [
    // Payslip(),
    Leaves(),
    Employee(),
  ],
)
```

Both classes connect to the same firebase document to retrieve different information and display it on their own respective tab scaffold.

```
final currentUserID = FirebaseAuth.instance.currentUser!.uid;
CollectionReference employees =
  FirebaseFirestore.instance.collection('Employees');
```

Storing document data in the variable data to be used later for leave and employee class:

```
child: FutureBuilder<DocumentSnapshot>(
  future: employees.doc(currentUserID).get(),
  builder:
    (BuildContext context, AsyncSnapshot<DocumentSnapshot> snapshot) {
      if (snapshot.connectionState == ConnectionState.done) {
        Map<String, dynamic> data =
          snapshot.data!.data() as Map<String, dynamic>;
```

In Leaves class, the leaves widget::

```
- CircularPercentIndicator(  
    radius: 120.0,  
    lineWidth: 15.0,  
    animation: true,  
    animationDuration: 600,  
    percent: (data['leaves'] / 30),  
    center: new Text('${data['leaves']} / 30',  
        style: TextStyle(fontSize: 17)), // Text  
    progressColor: Colors.tealAccent[400],  
    backgroundColor: Colors.grey[500],  
) // CircularPercentIndicator
```

In Leave class, the attendance widget:

```
- CircularPercentIndicator(  
    radius: 120.0,  
    lineWidth: 15.0,  
    percent: (data['attendance'] / 365),  
    center: new Text(  
        ((data['attendance'] / 365 * 100)  
            .toStringAsFixed(2) +  
        "%"),  
        style: TextStyle(fontSize: 17)), // Text
```

In Leave class, the payslip widget:

```
- ListTile(  
    title: Text(  
        'Payslip',  
        style: TextStyle(fontSize: 50),  
        textAlign: TextAlign.center,  
) // Text  
    subtitle: Text(  
        '\$' + (data['payslip']).toStringAsFixed(2),  
        textAlign: TextAlign.center,  
        style: TextStyle(fontSize: 50),  
) // Text  
) // ListTile
```

In employee class, a few widget examples in code snippets are for avatar, name and position. The rest of the details are of similar code structure.

In employee class, avatar widget:

```
- CircleAvatar(
    radius: 50,
    child: ClipOval(
        child: Image.network("${data['photo']}"),
    )), // ClipOval, CircleAvatar
```

In employee class, Text widget to display name:

```
- Text(
    "${data['name']} ${data['surname']}",
    style: TextStyle(
        fontSize: 25.0,
        color: Colors.white,
        fontWeight: FontWeight.bold,
    ), // TextStyle
    textAlign: TextAlign.center,
), // Text
```

In employee class, Text widget to display position:

```
- Text(
    "Position",
    style: TextStyle(
        color: Colors.indigo,
        fontSize: 24.0,
    ), // TextStyle
), // Text
- SizedBox(
    height: 5.0,
), // SizedBox
- Text(
    "${data['position']}",
    style: TextStyle(
        fontSize: 28.0,
        color: Colors.redAccent,
        fontWeight: FontWeight.bold,
    ), // TextStyle
), // Text
```

4. Announcement board

For the announcement page, as the announcement is meant for all users to see the same thing, instead of querying the documents, we query the collection snapshots and display all the documents in a grid list for viewing.

The below image is where we query the whole announcement collection snapshot:

```
CollectionReference announcements =
    FirebaseFirestore.instance.collection('Announcements');
```

```
body: StreamBuilder(
    stream: announcements.snapshots(),
    builder: (context, AsyncSnapshot<QuerySnapshot> snapshot) {
        if (!snapshot.hasData) {
            return Center(child: Text("Loading..."));
        }
        return GridView.count(
```

Also, editing and deleting the announcement is done through pressing the selected buttons from a pop up menu. Different action will be called respectively:

```
PopupMenuButton(itemBuilder: (context) {
    return [
        PopupMenuItem(
            value: "Edit",
            child: Text("Edit"),
        ), // PopupMenuItem
        PopupMenuItem(
            value: "Delete",
            child: Text("Delete"),
        ) // PopupMenuItem
    ];
}, onSelected: (String value) {
```

Delete:

```
if (value == "Delete") {  
    announce.reference.delete();
```

Edit:

```
if (value == "Edit") {
    textTitle.text = announce["title"];
    textMessage.text = announce["message"];
    showDialog(
        context: context,
        builder: (context) {
            return Dialog(
                shape: RoundedRectangleBorder(
                    borderRadius:
                        BorderRadius.circular(40)), // RoundedRectangleBorder
                elevation: 16,
                child: Container(
                    child: ListView(
                        shrinkWrap: true,
                        children: <Widget>[
                            SizedBox(height: 20),
                            Center(
                                child: Text(
                                    'Update announcement')), // Text, Center
                            SizedBox(height: 20),
                            TextField(
                                decoration: InputDecoration(
                                    border: InputBorder.none,
                                    contentPadding:
                                        EdgeInsets.only(
                                            top: 14.0), // EdgeInsets.only
                                    labelText: 'Title',
                                ), // InputDecoration
                                controller: textTitle,
                            ), // TextField
                            TextField(
                                decoration: InputDecoration(
                                    border: InputBorder.none,
                                    contentPadding:
                                        EdgeInsets.only(
                                            top: 14.0), // EdgeInsets.only
                                    labelText: 'Message',
                                ), // InputDecoration
                                controller: textMessage,
                            ), // TextField
                            RaisedButton(
                                child: Text("Update"),
                                onPressed: () {
                                    announce.reference
                                        .update({
                                        'title': textTitle.text,
                                        'message':
```

Add announcement:

```
floatingActionButton: FloatingActionButton(
  child: Icon(Icons.add),
  backgroundColor: Colors.indigo,
  onPressed: () {
    showDialog(
      context: context,
      builder: (context) {
        return Dialog(
          shape: RoundedRectangleBorder(
            borderRadius: BorderRadius.circular(40)), // RoundedRectangleBorder
          elevation: 16,
          child: Container(
            child: ListView(
              shrinkWrap: true,
              children: <Widget>[
                SizedBox(height: 20),
                Center(child: Text('Add new announcement')),
                SizedBox(height: 20),
                TextField(
                  decoration: InputDecoration(
                    border: InputBorder.none,
                    contentPadding: EdgeInsets.only(top: 14.0),
                    labelText: 'Title',
                  ), // InputDecoration
                  controller: textControllerTitle,
                ), // TextField
                TextField(
                  decoration: InputDecoration(
                    border: InputBorder.none,
                    contentPadding: EdgeInsets.only(top: 14.0),
                    labelText: 'Message',
                  ), // InputDecoration
                  controller: textControllerMessage,
                ), // TextField
                RaisedButton(
                  child: Text("Add"),
                  onPressed: () {
                    announcements.add({
                      'title': textControllerTitle.text,
                      'message': textControllerMessage.text,
                    });
                    textControllerTitle.clear();
                    textControllerMessage.clear();
                    Navigator.pop(context, announcements);
                  },
                ),
              ],
            ),
          ),
        );
      },
    );
  },
);
```

5. Leave applying

For the leave applying days, it is almost similar to the main page calendar. The only difference is that we can select multiple consecutive days for the leaves and that there isn't any time choosing since the leaves for applying will be a full day type of leaves. Also, there is an additional field called status which will be accessed and used for the request viewing page.

The leave object:

```
class Leave {  
    final String status;  
    final String title;  
    final String description;  
    final DateTime from;  
    final DateTime to;  
    final Color backgroundColor;  
    final bool isAllDay;  
  
    const Leave({  
        required this.status,  
        required this.title,  
        required this.description,  
        required this.from,  
        required this.to,  
        this.backgroundColor = Colors.indigo,  
        this.isAllDay = true,  
    });  
}
```

One example of querying and adding the leave document in the firebase in LeaveEditingPage:

```
provider.addLeave(event);  
int index = provider.getIndexLeave(event);  
leaves  
    .doc(currentUserID)  
    .collection("Leaves")  
    .doc(index.toString())  
    .set(  
    {  
        "status": "Pending",  
        'title': titleController.text,  
        'description': descriptionController.text,  
        'from': fromDate,  
        'to': toDate  
    },  
    SetOptions(merge: true),  
)  
    .then((value) => print("Leaves Added"))  
    .catchError((error) => print("Failed to add leave: $error"));
```

6. Request status

For the request, it is to access the Leaves Collection under the Employees collection for the current login user document to display all leaves and to show the status. The request status can be modified by the HR through the firebase itself.

Accessing the firebase Employee collection:

```
CollectionReference requests =
    FirebaseFirestore.instance.collection('Employees');
final currentUserID = FirebaseAuth.instance.currentUser!.uid;
```

Getting the snapshot of all the leaves application:

```
_body: StreamBuilder(
    stream:
        requests.doc(currentUserID).collection("Leaves").snapshots(),
    builder: (context, AsyncSnapshot<QuerySnapshot> snapshot) {
```

List tile widget of all leaves including title of leaves and the status in Requests class:

```
child: ListTile(
    title: Transform.translate(
        offset: Offset(0, 2),
        child: Text(
            requests['title'],
            style: TextStyle(
                fontSize: 20.0,
                color: Colors.black,
            ), // TextStyle
        ), // Text
    ), // Transform.translate
    subtitle: Transform.translate(
        offset: Offset(0, 4),
        child: Text(
            "Leave status: " + requests['status'],
            style: TextStyle(
                fontSize: 15.0,
                color: Colors.black,
            ), // TextStyle
        ), // Text
    ), // Transform.translate
```

Pressing the list tile will prompt a pop up alert dialogue to show the title, dates of leaves and also the description of leaves:

```
onTap: () {
    DateTime myDateTimeTo = requests["to"].toDate();
    DateTime myDateTimeFrom = requests["from"].toDate();
    showDialog(
        context: context,
        builder: (BuildContext context) => AlertDialog(
            backgroundColor: Colors.white,
            title: Center(
                child: Text(requests['title'],
                    style: TextStyle(
                        color: Colors.black,
                        fontSize: 18,
                        fontWeight: FontWeight.bold,
                    )), // TextStyle, Text, Center
            content: Column(
                mainAxisSize: MainAxisSize.min,
                children: <Widget>[
                    Text(
                        myDateTimeFrom
                            .toString()
                            .substring(0, 10) +
                        " - " +
                        myDateTimeTo
                            .toString()
                            .substring(0, 10),
                    style: TextStyle(
                        fontSize: 12,
                        color: Colors.black,
                    ), // TextStyle
                ], // Text
            ), // Column
        ), // AlertDialog
    ); // showDialog
}, // onTap
```

```
    Text(
        requests['description'],
        style: TextStyle(
            color: Colors.black,
        ), // TextStyle
    ), // Text
], // <Widget>[]
), // Column
actions: <Widget>[
    RaisedButton(
        color: Colors.indigo[100],
        onPressed: () =>
            Navigator.pop(context, 'OK'),
        child: const Text('OK'),
    ),
]
```

7. FAQ page

For the frequently asked question page, we did not store the questions and answers in the firebase but instead used a list of {question, answer and category} to store our data. We made use of a group list view to group by the category of the type of questions and display them in a list tile to be tapped to navigate to the DetailScreen page.

Example of the list in FAQ class:

```
List _elements = [
  {
    'qns': 'How to use the Announcement page?',
    'ans':
      'The announcement is a place where employees can be notified of any new announcements. To make an announcement, press the button at the top right of the screen and enter your message. You can also edit or delete existing announcements by clicking on the three dots at the side of each announcement and then hitting update!',
    'cat': 'How to use'
  },
  {
    'qns':
      'Oh no! I wish to edit or delete the announcement! Is that possible?',
    'ans':
      'Yes it is possible! Please click the three dots at the side to edit or delete the announcement you have made and then hit update!',
    'cat': 'How to use'
  }
]
```

Grouped list view in FAQ class arranged by alphabetical order:

```
-- body: GroupedListView<dynamic, String>(
  elements: _elements,
  groupBy: (element) => element['cat'],
  groupComparator: (value1, value2) => value1.compareTo(value2),
  itemComparator: (item1, item2) =>
    item1['qns'].compareTo(item2['qns']),
  order: GroupedListOrder.ASC,
```

List tile widget to display the question and when tapped, will call the DetailScreen class:

```
-- child: ListTile(
  tileColor: Colors.blue[100],
  contentPadding:
    EdgeInsets.symmetric(horizontal: 20.0, vertical: 10.0),
  leading: Icon(Icons.account_circle),
  title: Text(element['qns']),
  trailing: Icon(Icons.arrow_forward),
  onTap: () {
    Navigator.push(
      context,
      MaterialPageRoute(
        builder: (context) =>
          DetailScreen(element['qns'], element['ans']),
    ), // MaterialPageRoute
```

DetailScreen class to display the questions and answers:

```
class DetailScreen extends StatelessWidget {
    final String qns;
    final String ans;
    DetailScreen(this.qns, this.ans);

    @override
    Widget build(BuildContext context) {
        return Scaffold(
            appBar: AppBar(
                title: Text(qns),
            ), // AppBar
            body: Padding(
                padding: const EdgeInsets.all(16.0),
                child: Text(ans),
            ), // Padding
        ); // Scaffold
    }
}
```

8. To-Do List Page

```
abstract class BaseTodo {  
    Future<List<Todo>> retrieveTodo();  
    Future insertTodo(Todo todo);  
    Future updateTodo(Todo todo);  
    Future removeTodo(Todo todo);  
}
```

We create an abstract class that `TodoFirebase` will implement.

The 4 abstract methods are `retrieveTodo()`, `insertTodo(Todo todo)`, `updateTodo(Todo todo)` and `removeTodo(Todo todo)` and the return type is **Future** as they are asynchronous operations (database interactions)

```
class TodoFirebase implements BaseTodo {  
    final FirebaseAuth _firebaseAuth = FirebaseAuth.instance;
```

Here we define a new class that makes use of, or implement, the `BaseTodo` abstract class defined earlier.

```
Future<List<Todo>> retrieveTodo() async {  
    var ref = await databaseReference.collection("Employees").doc(_auth.currentUser!.uid).collection("MyTodos").get();  
  
    List<Todo> todos = [];  
    ref.docs.forEach((element) {  
        var data = element.data();  
        if (data['user'] == _auth.currentUser!.uid) {  
            var t = new Todo(  
                createdTime: DateTime.parse(data['createdTime']),  
                id: element.id,  
                title: data['title'],  
                isDone: data["isDone"] == "true" ? true : false,  
                description: data['description']);  
            todos.add(t);  
        }  
    });  
  
    return todos;  
}
```

`retrieveTodo()` retrieves data from firebase, under collection “MyTodos” which is under “Employees”. We loop through and create a `Todo` object and pass in the respective data. In each loop we add the `Todo` object into a list of `Todos`.

```
Future removeTodo(Todo todo) async {
  var ref = await databaseReference.collection("Employees").doc(_auth.currentUser!.uid).collection("MyTodos");
  ref.doc(todo.id).delete();
}
```

removeTodo(Todo todo) removes data from firebase, under collection “MyTodos” which is under “Employees”. It removes the data based on the randomly generated unique id when we first inserted data into the firebase. This unique id generation is handled by firebase.

```
Future insertTodo(Todo todo) async {
  var ref = await databaseReference.collection("Employees").doc(_auth.currentUser!.uid).collection("MyTodos");

  ref.add({
    "id": todo.id,
    "title": todo.title,
    "description": todo.description,
    "createdTime": todo.createdTime.toString(),
    "user": _auth.currentUser!.uid,
    "isDone":todo.isDone.toString(),
  });
}
```

insertTodo(Todo todo) inserts data to firebase, under collection “MyTodos” which is under “Employees”. We simply pass in the data by accessing the todo object we passed in via the argument.

```
Future updateTodo(Todo todo) async {
  var ref = await databaseReference.collection("Employees").doc(_auth.currentUser!.uid).collection("MyTodos");
  ref.doc(todo.id).update({
    "id": todo.id,
    "title": todo.title,
    "description": todo.description,
    "createdTime": todo.createdTime.toString(),
    "user": _auth.currentUser!.uid,
    "isDone":todo.isDone.toString(),
  });
}
```

updateTodo(Todo todo) updates existing data in firebase, under collection “MyTodos” which is under “Employees”. It looks for the correct existing data based on the unique id.

```

class TodoListWidget extends StatelessWidget {
  @override
  Widget build(BuildContext context) {
    final provider = Provider.of<TodosProvider>(context);

    return FutureBuilder<List<Todo>>(
      future: provider.todos,

      builder: (BuildContext context, snapshot) {
        if (snapshot.hasData && snapshot.data!.length > 0) {

          return ListView.separated(
            physics: BouncingScrollPhysics(),
            padding: EdgeInsets.all(16),
            separatorBuilder: (context, index) => Container(height: 8),
            itemCount: snapshot.data!.length,
            itemBuilder: (context, index) {
              final todo = snapshot.data![index];

              return TodoWidget(todo: todo);
            },
          ); // ListView.separated
        } else {

          return Center(
            child: Text(
              'No todos.',
              style: TextStyle(fontSize: 20),
            ); // Text, Center
        }
      }, // FutureBuilder
    );
  }
}

class TodoListWidget extends StatelessWidget {
  @override
  Widget build(BuildContext context) {
    final provider = Provider.of<TodosProvider>(context);

    return FutureBuilder<List<Todo>>(
      future: provider.todos,

      builder: (BuildContext context, snapshot) {
        if (snapshot.hasData && snapshot.data!.length > 0) {

          return ListView.separated(
            physics: BouncingScrollPhysics(),
            padding: EdgeInsets.all(16),
            separatorBuilder: (context, index) => Container(height: 8),
            itemCount: snapshot.data!.length,
            itemBuilder: (context, index) {
              final todo = snapshot.data![index];

              return TodoWidget(todo: todo);
            },
          ); // ListView.separated
        } else {
          return Center(
            child: Text(
              'No todos.',
              style: TextStyle(fontSize: 20),
            ); // Text, Center
        }
      }, // FutureBuilder
    );
  }
}

```

We use FutureBuilder as database data retrieval is asynchronous and we used Future for that.

Discussion of results

Overall, the app meets our overall objectives - we aimed to create a user-based application for employees and HR managers to be more efficient in employee-related tasks and overall communication. Employees can view tasks, meetings, and personal details on the go anytime. They can also keep track of ongoing tasks and schedules, and be reminded when they are on leave.

Looking at this we will discuss our evaluation of the functionalities below.

| Functionality | Does it meet our objectives? |
|---------------------------------|--|
| Personal work calendar | <p>Once the user has been authenticated, the user is able to get a daily view of his tasks.</p> <p>He is able to create an event, view and edit it.</p> <p><i>This meets our objectives</i> in easy viewing of scheduled tasks. Users can rely on this app to provide a consolidated view of all outstanding tasks. Rather than relying on physical notes to keep track of them, the app is able to consolidate all ongoing and future events, with all relevant details included.</p> |
| View personal details | <p>The user is able to view all of his/her personal details - this includes payslip details, leaves and attendance, personal information.</p> <p><i>This meets our objectives</i> in showing the user relevant information regarding themselves and is consolidated and presented in an easy-to-read way.</p> |
| Apply for leaves/ view requests | <p>The user can make a request (e.g. leaves) anytime on the go. They can view the number of leaves they have taken and apply for one accordingly. This results in lesser human error where HR have to personally keep track of leave and absence records of their employees.</p> <p><i>This meets our objectives</i> of on-the-go leave application and application status. Instead of having to make a call to the company or wait for a weekday to schedule a meeting with HR personnel, the user can simply log on to the application to view his/her application status. It is hassle free and can be done independently by the user.</p> |
| To-Do List | The user can add a list of tasks which need to be done. A to-do list serves as a memory aid. Only a few pieces |

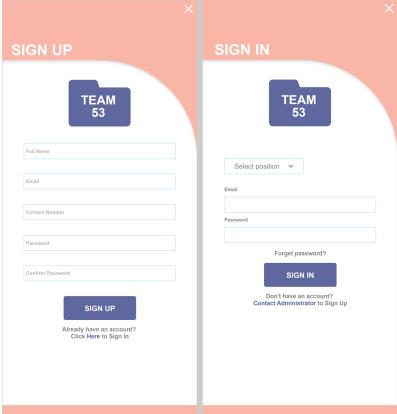
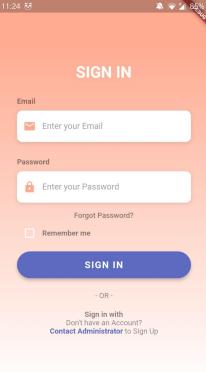
| | |
|--------------------|--|
| | <p>of information may be held in our head at any given moment. Having a to-do list will help users to keep track of everything. This to-do list will also serve to reinforce the information, making it less likely that employee/HR will forget something important.</p> <p><i>This meets our objectives</i> as it increases productivity. Having a to-do list helps employees/HR to prioritise their most important tasks. This implies that employees/HR won't waste time on things that don't need immediate attention. This to-do list will assist them in staying focused on the most crucial tasks. Both employees/HR can cross a task off their list once they have completed it. This offers employees/HR a sense of accomplishment and progress, which they won't get if they are continually racing from one activity to the next. A sense of accomplishment encourages and inspires them to continue going forward.</p> |
| View announcements | <p>To facilitate better and less error free communication, any announcements from the company can be uploaded onto the app.</p> <p><i>This meets our objectives</i> of more efficient and error-free communication. Traditional communication would be physical passing of messages from one person to the next, and messages may lose their integrity over time, resulting in miscommunication.</p> |
| FAQ | <p>The FAQ message serves to help users navigate the app in the event they are unsure of the functionalities.</p> <p><i>This meets our objectives</i> in providing hassle-free help should the employees face any problem.</p> |

Evaluation

Original proposal & plan vs results

Within the short time frame, we can proudly say that our app is fully functioning, and all the core functions are working as intended. However, not everything in our proposal has been implemented.

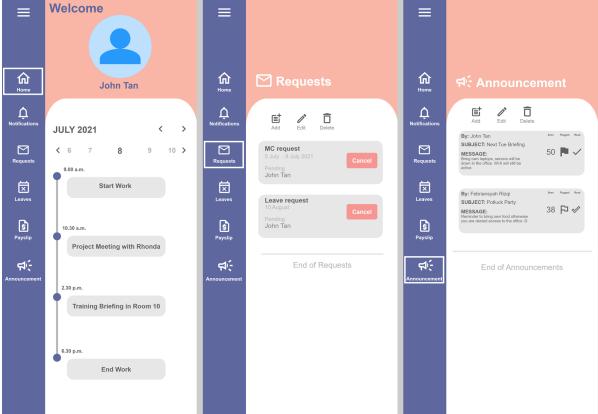
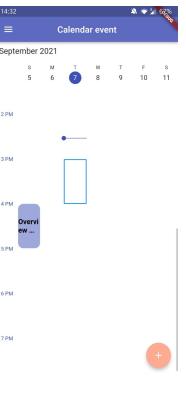
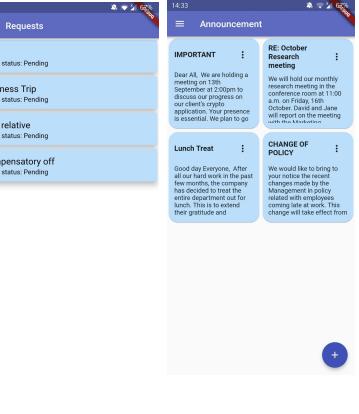
(Side by side comparison - photos of each)

| Original proposal | Our app |
|--|--|
|  |  |

Results (what we achieved and what we did not achieve):

In our original proposal, our aim was to build one single application, which will allow users to login as an employee or HR, depending on their positions. Having login to either position will give different access and functionalities. This position can be selected on the login page itself. However, for our current app, we have created the app solely based on employees in mind.

Our aim was also to allow users of our app to sign up to create a new account on their own. However, this was not achieved.

| | | |
|---|--|---|
|  |  |  |
|---|--|---|

Results (what we achieved and what we did not achieve): We have achieved the following according to our proposal - the main page, requests and announcement pages.

In our main page, we opted to have a weekly view rather than a daily view. However, all tasks are still displayed.

Looking at the above comparison, we have achieved the design for both our request and announcement page as per the proposal. However, for the request page, after making a request, the user will not be able to edit it anymore.

The image displays four mobile application screens side-by-side:

- Notifications Screen:** Shows a list of notifications from users Rizqi, Maeve, and others, each with a small icon and a brief description.
- Payslip Screen:** Displays a summary of employee attendance and a payslip amount of \$4000.00.
- Profile Screen (Left):** Shows a circular progress bar for 'LEAVES' (25/30), a circular progress bar for 'ATTENDANCE' (79.18%), and a total 'Payslip' amount of \$2800.00.
- Profile Screen (Right):** Shows a detailed profile for 'David Toh' with the title 'Boss'. It includes personal information such as User ID, Email, Full Name, Date of birth, Address, and Phone Number.

Results (what we achieved and what we did not achieve):

Looking at the above comparison, we have achieved the payslip page, however it is under the user profile. It displays the same information in the same design.

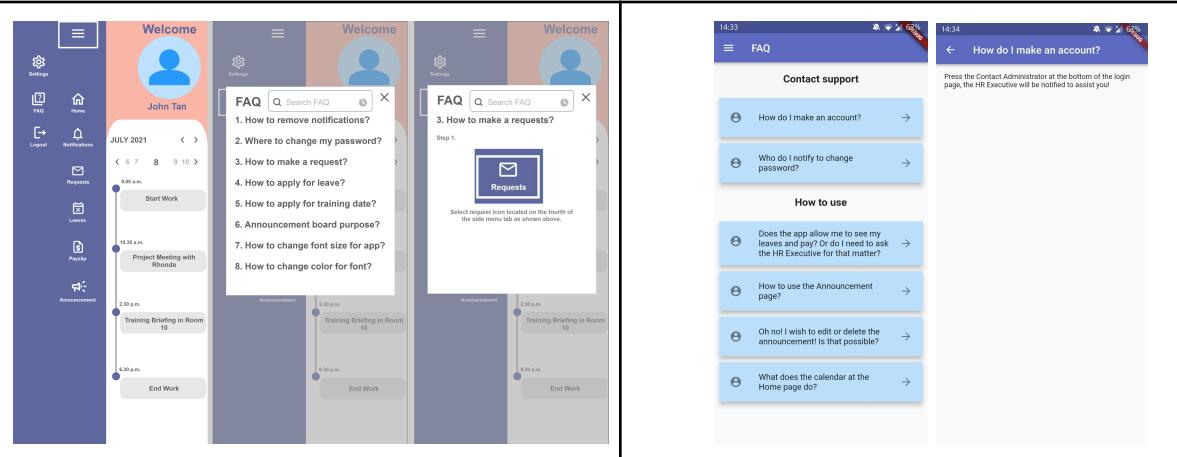
We have, however, left out the notification page.

The image displays four mobile application screens side-by-side:

- Leaves Screen:** Shows a weekly calendar for July 2021 with specific dates highlighted in blue.
- Leave Application Screen:** Shows a form to apply for leave, including fields for start date, end date, and file attachment, with 'Approve' and 'Cancel' buttons.
- Leaves Screen (with Confirmation):** Shows a confirmation message: 'Thank you for applying!' and 'Please wait for your approval'.
- Leave Application Calendar Screen:** Shows a monthly calendar for September 2021 with specific dates highlighted in blue, used for scheduling leave requests.

Results (what we achieved and what we did not achieve):

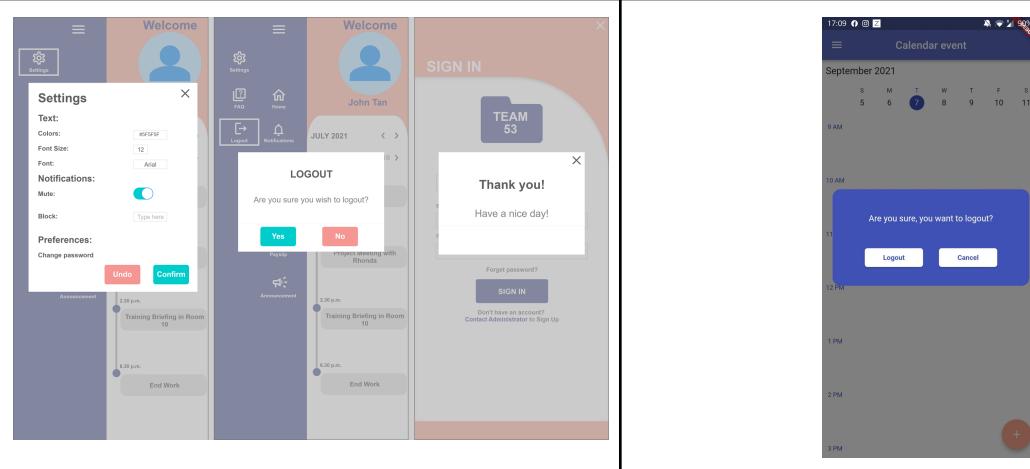
Looking at the comparison above, we have achieved our calendar leave functionality. Users are able to apply for a leave and can also view details of the leave (type, date etc.)



Results (what we achieved and what we did not achieve):

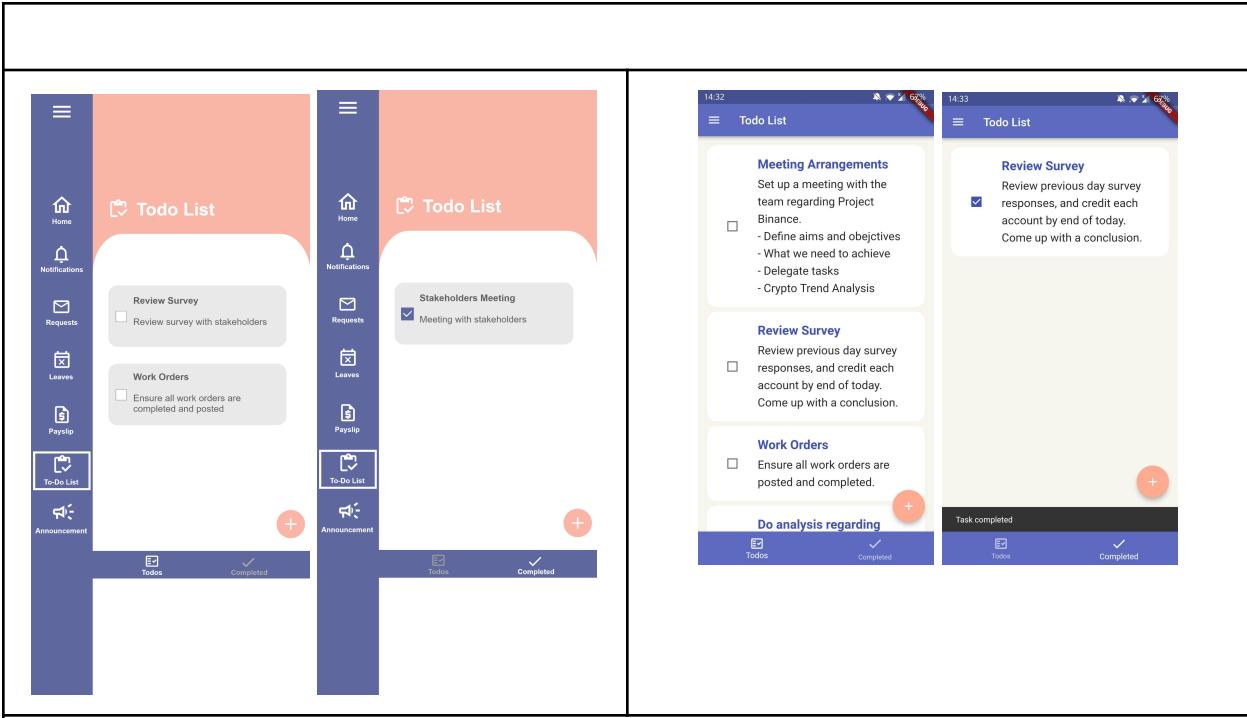
Looking at the comparison above, we have achieved our FAQ functionality, where users are able to click on an FAQ.

However, we do not have a search bar where users can search regarding a particular question. Users need to scroll down and glance through each FAQ and see which FAQ is the one they are looking for.



Results (what we achieved and what we did not achieve): Looking at the comparison above, we have achieved the functionality to give the user the option to logout from the app.

However, we did not implement the Settings page, where users can customise the appearance of the text to their preference, giving the users the option to mute or block notifications as well as changing their password.



Additional feature We added a to-do list function, accessible on the drawer bar itself because we feel that our users will greatly benefit from it. This to-do list page serves as a platform for users to organise their tasks during the day. Users can ensure that all of their tasks and responsibilities are written down in one place and that they do not overlook anything essential and important by keeping a list as such. They may also arrange the sequence in which they will complete things by prioritising them, so they know what requires immediate attention and what can wait.

Analysis of our products vs others in the market:

In our first report, we analysed existing big players in the market. In this section, we will compare our software with theirs.

| Functionalities | DELTA | Workday | ConnectTeam | Monday.com WorkOS |
|---------------------------------------|-----------------------------|--|---|---|
| Security | Email and password login | Biometric identification | Email and password login | Email and password login |
| Communication methods | None | Employee Portal Internal chat integration | Live-chat groups | Able to leave comments on specific projects which can be found in the inbox section |
| Broadcasting methods | Announcement page | Announcements page | Announcement Newsletters | Changes to the projects show on the notifications page. |
| Employee feedback methods | None | Talent Management Feedback Management | Feedback surveys Live polls Suggestion box | No native feedback but able to connect to other apps such as SurveyMonkey |
| Time tracking | None | Time tracking | Time tracking for clock-in and clock-out Time sheets to calculate overtime/double time | Time tracking widget. |
| Leave and attendance request/tracking | Leave and attendance status | Absence Management | Absence and paid time off application notifications to make approvals and requests | none |
| Scheduling methods | Personal work calendar | Calendar Sync | Employee scheduling | Scheduling is available on the |

| | | | | |
|---------------------------------|---|--|-----------------------------------|---|
| | Leave calendar | | | projects to provide assigned users with a deadline. Time stamps are also available for notifications. |
| Salary and personal information | User profile page that displays personal information. | Personal information page displays information such as home, birth, gender etc. and can be edited. | Personal information is displayed | No salary displayed. Personal information is available and admins may access and change them. |

Things to improve:

Looking at the above comparison with the other players in the market, more can be improved regarding the functionalities of DELTA. Whilst our app is able to provide basic features, it can be seen that we are lacking in providing advanced capabilities (in the table above). As we are a newcomer to this market, more research and development will be needed down the road for us to push out more functionalities to cater the growing needs and demands of consumers.

Push notifications

First thing first, our app lacks the functionality of a push notification. We strongly feel that push notifications will improve the user's experience significantly. They are less invasive than other types of user interaction, and they may successfully generate traffic and assist users by giving relevant information. They let your users stay up to date on the newest news, important announcements, and approval of leave requests. However, push notifications will be disabled if individuals do not want to hear from you. Unlike text messaging, this mode of communication leaves the user in total control. So it is highly customisable for the users themselves, what notifications they would want to see in their lock screen.

Enhanced security through biometrics

To increase the overall security of our app, users can use biometric authentication to login instead of the traditional username and password. Biometrics, as opposed to passwords, are a much-needed advancement. Hacking passwords is a breeze. A hacker may just require a person's date of birth and the name of their pet in some cases. On the other hand, biometrics are far more difficult to obtain. The quick, convenient, and simple user experience is one of the main advantages of biometric authentication systems. It is considerably easier to utilise biometric authentication since it does not need users to recall something they have or something they know. Whether the system involves scanning a face or a fingerprint, the process is simple and quick, resulting in a user-friendly identification process.

Time-in Time-out

As employees are now mostly on the move (The Rise Of Deskless Workforce, 2018), we would want to implement a time-in/time-out process based on their location, and they are able to take attendance whenever and wherever they are. The app collects geolocation data from all mobile punches and check-ins. This information will be available in the employee profiles and reports. To better verify the information, all punches and check-ins can be accompanied with pictures. This will avoid the problem of buddy punching and inaccurately reported payroll hours. HR can also set up geofences so that employees may only clock in and out from specific locations. Allowances and radius can be customised for each location. This is to make certain that the payroll department only pays for hours that are supposed to be paid for.

Learning On the Move

Mobile learning is the most rapidly growing area of HR technology. As a result, employees no longer have to sit through long, boring traditional training sessions; instead, they can learn on their own time and on their own terms, whenever and wherever they want. Employees

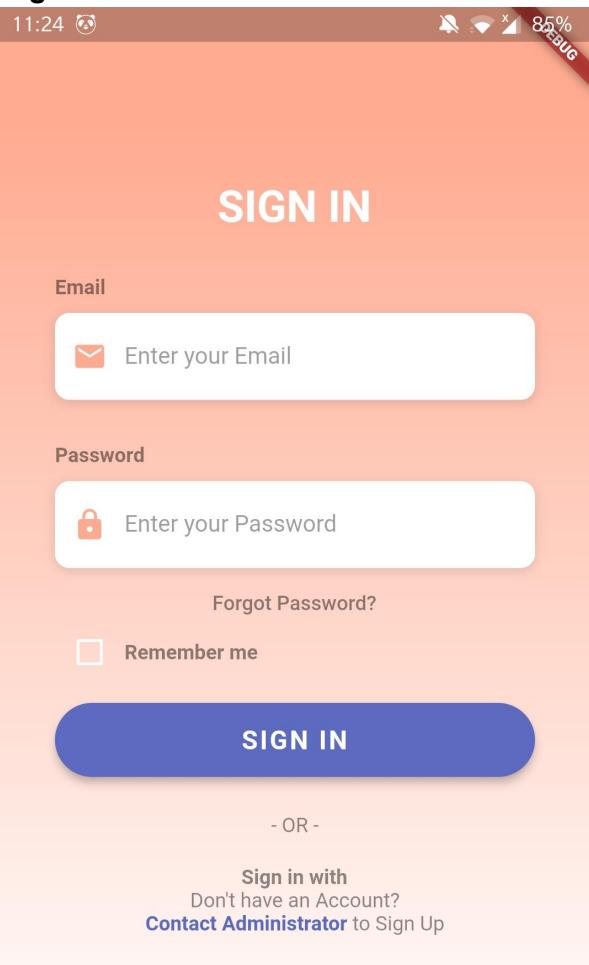
appreciate the flexibility that e-learning provides because it allows them to spend less time away from the office and their families. Employees can also take charge of their own learning and participate in training at their leisure, allowing them to better manage their work/life balance. Videos, games, quizzes, and tutorials are all stored in the cloud in order to keep employees engaged while also providing ongoing knowledge enhancement. The HR department can utilise the mobile app to administer quizzes and other forms of exams after employees have completed an online course and earned their certificate. In turn, training courses become simpler, allowing the team to concentrate on their daily tasks.

Advanced Recruiting Tools

A simplified recruitment process implemented in our app will be helpful. We need to streamline the hiring process if we want future applicants to stick around and be hired. Candidate experience is improved when managers have access to an advanced and feature-rich recruiting tool. A sophisticated and feature-rich recruiting tool makes the hiring process easier for both candidates and managers. Candidates can fill out the mobile form and apply while on the go. The simple forms ensure that potential candidates do not have to fill out all of the information. Only the information needed for quick processing or scanning is requested. The candidate list is sent to management after the details have been scanned and shortlisted based on a pre-defined algorithm. The HR team can use the information gleaned from candidate research and analysis to help top management decide whether or not to hire these individuals. The mobile apps will provide a healthy method of conducting recruitments and will make the process easier for both candidates and employers.

User Guide & Documentation

Login



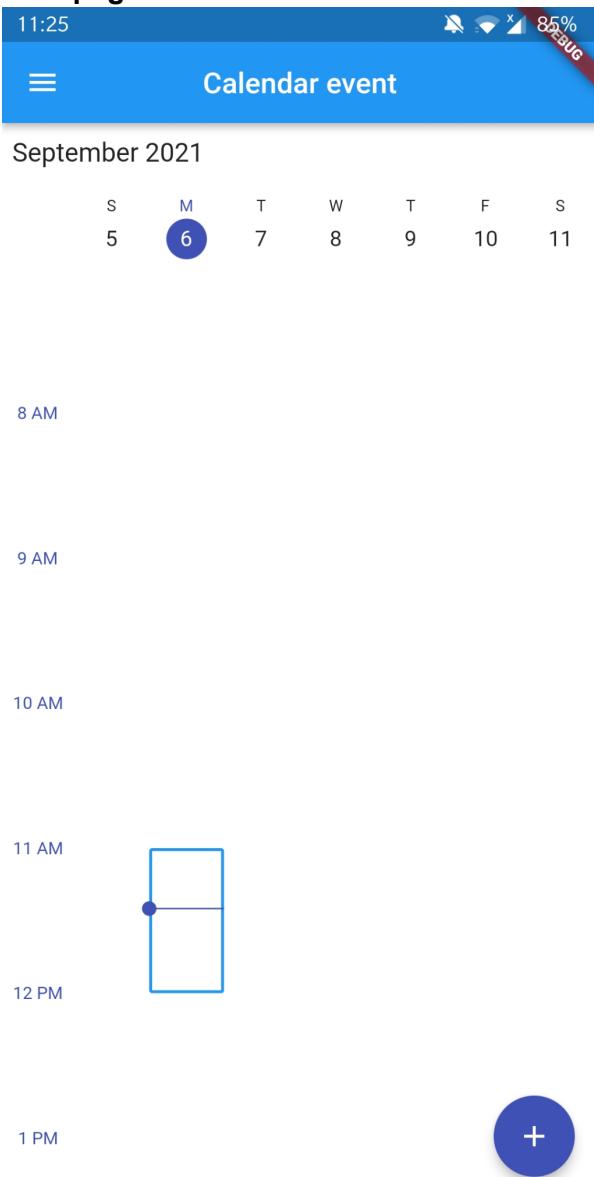
1. To login

- Key in *employee email
- Key in *password
- Click on 'SIGN IN' button

Once the user has been authenticated, the user will be brought to the day timeline on the main page.

*All employees are given a company-registered email as well as a password.

Main page



The main page shows a calendar with the overall view of the week.

Dates that have been marked out for an event will be highlighted in green with the name of the event.

1. To add an event

Click on the '+' button located at the bottom right-hand corner on the screen. This button will navigate the user to the event form page to fill event details (name, day, date, description).

2. To view details of the event

Select the event on the main calendar page. The details of the selected event will be shown on the event form page with all the details key-ed in upon creating the event.

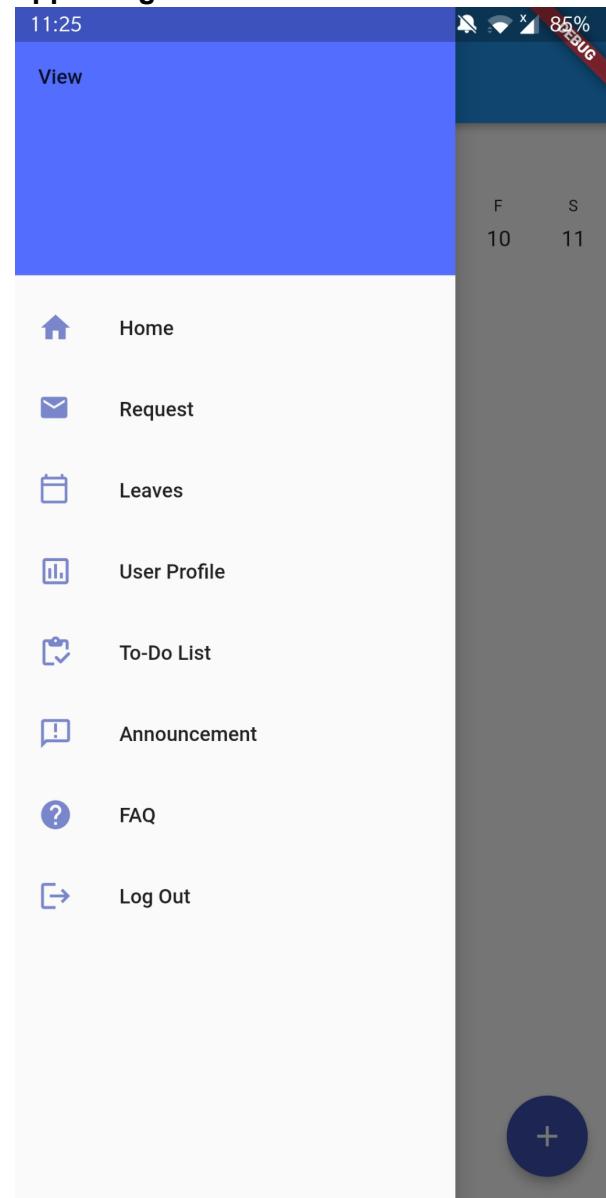
3. To edit an event

Select the event on the main calendar page. Click on the lower left pencil icon located at the top right hand corner to edit the selected event.

4. To delete an event

Select the event on the main calendar page. Click on the trash icon located at the top right hand corner of the app bar to delete the selected event.

App Navigation - Menu Drawer



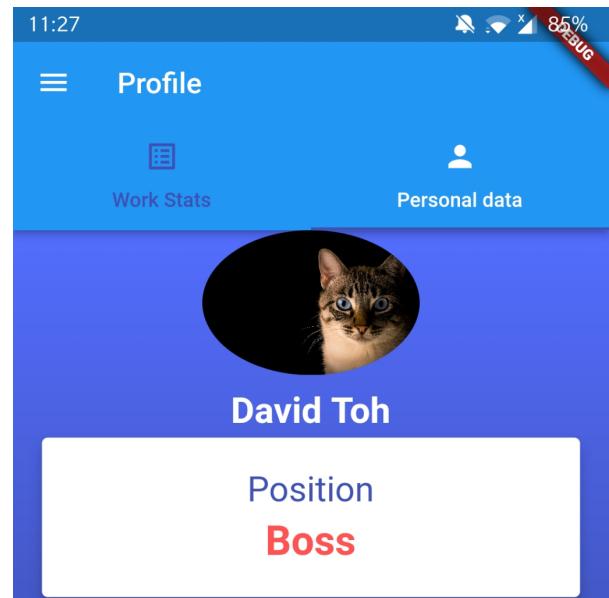
The user can navigate to any page on the application by using the menu drawer.

1. To get to other pages within the app

Click on the '≡' icon located on the left hand side of the application bar at the top.

This provides an overall view to all the pages in which the user can navigate to in the application.

User Profile - Personal Data



This page is for viewing for the user's personal information. The user can navigate between their work statistics page and personal data.

Personal Information

User ID: zEfB25UlcdMqZPg6u9iEatmav0T2

Email: david.toh@delta53.com

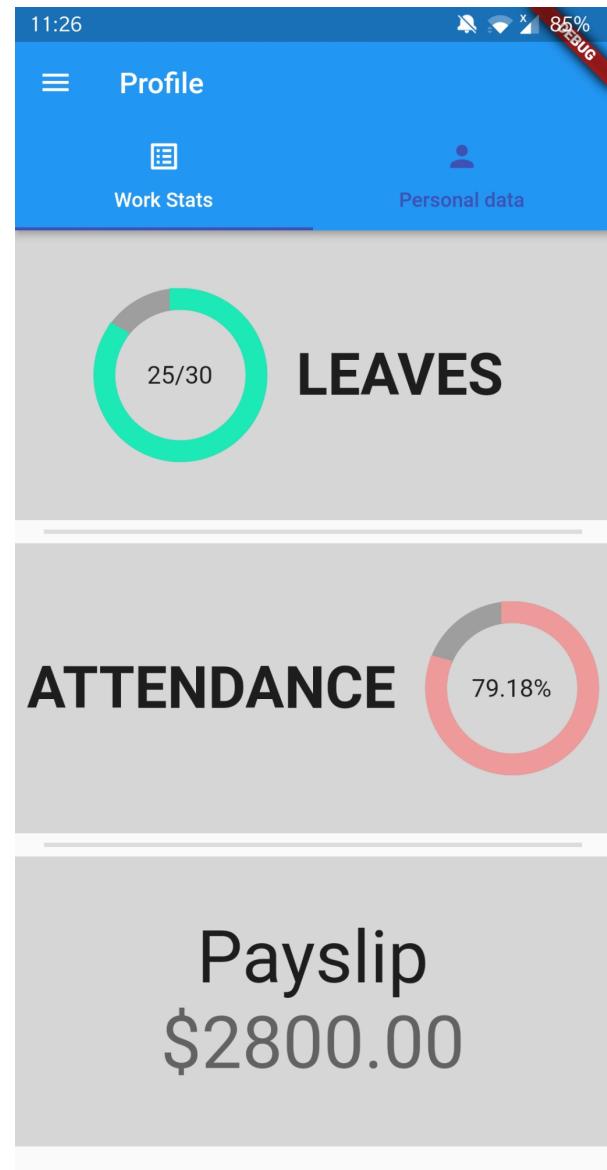
Full Name: David Toh

Date of birth: 20 June 1970

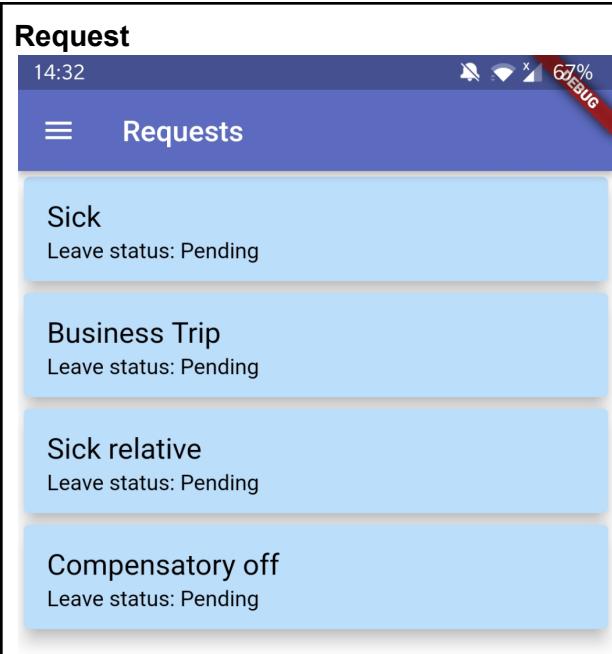
Address: 51 Bull Street S812633

Phone No: 98334510

User Profile - Work Stats



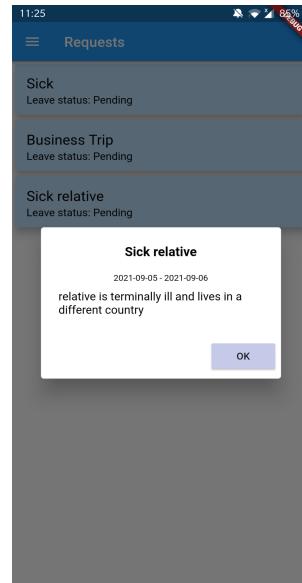
The user can view and track their leave attendance record as well as their payslip all under the 'Work Stats' tab.



The request page shows the status of all requests made on the 'Leaves' page.

1. To view a request and its details

Select the request on the page. A pop up will show all the details of the request.



Leaves



This is the page where users can apply their leave. It will show a calendar with a monthly view.

1. To request a leave

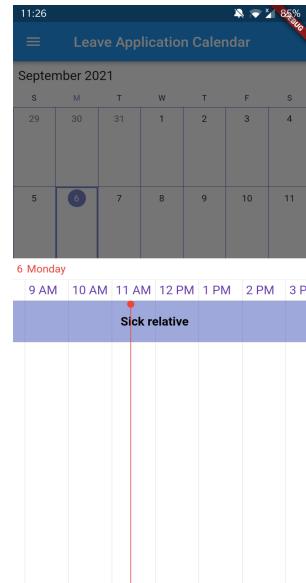
Click on the '+' button located at the bottom right hand corner of the screen. This button will navigate the user to the request form page to fill leave request details (name, day, description).

*Successful submission of the leave request will be indicated by a small dot on the calendar.

2. To view a leave request

Long press on the selected date (dates that have a request will be indicated with a tiny blue dot). This will show up with a timeline with the title of the leave request.

*To otherwise view the status of the request, navigate to the ‘Request’ page using the menu drawer.

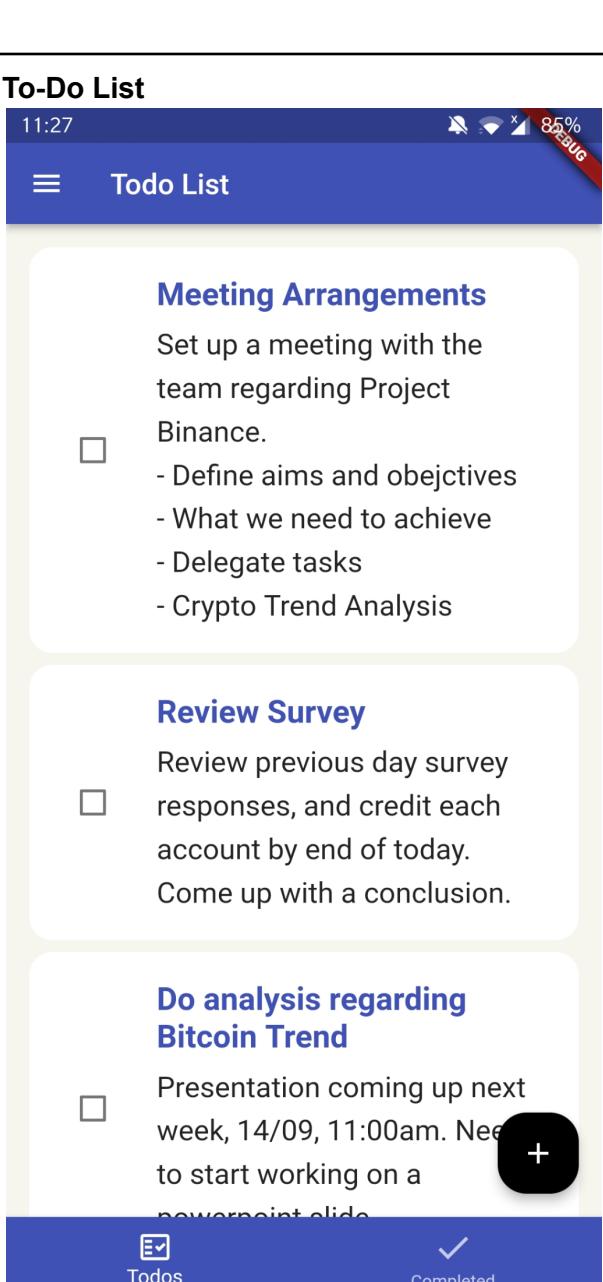


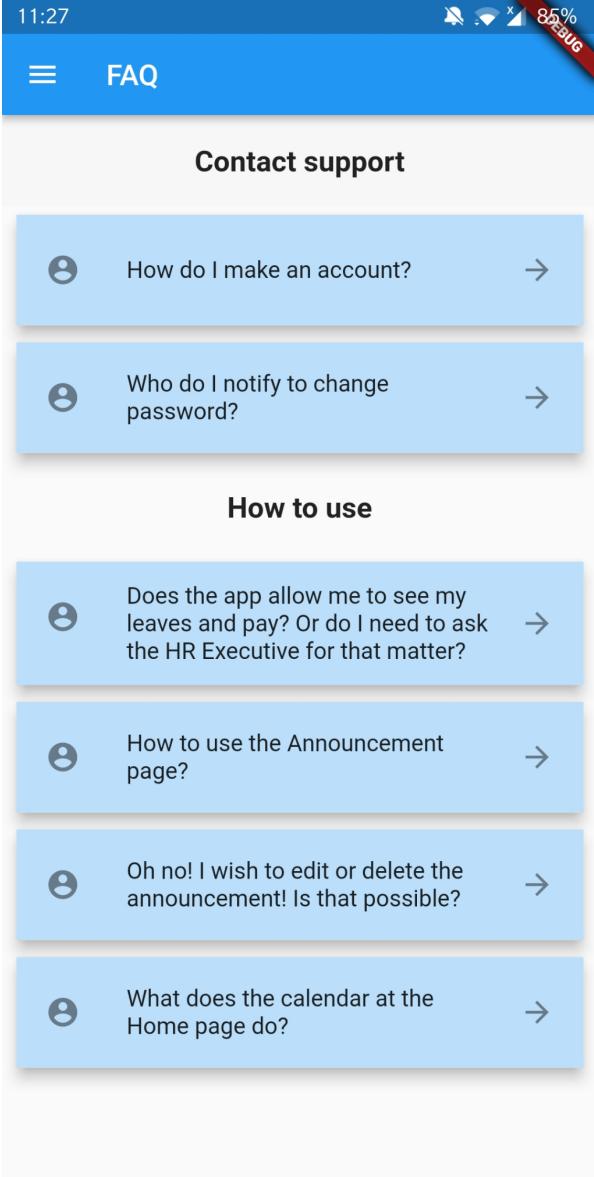
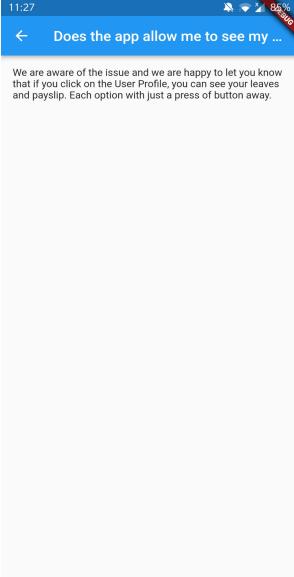
3. To edit a request

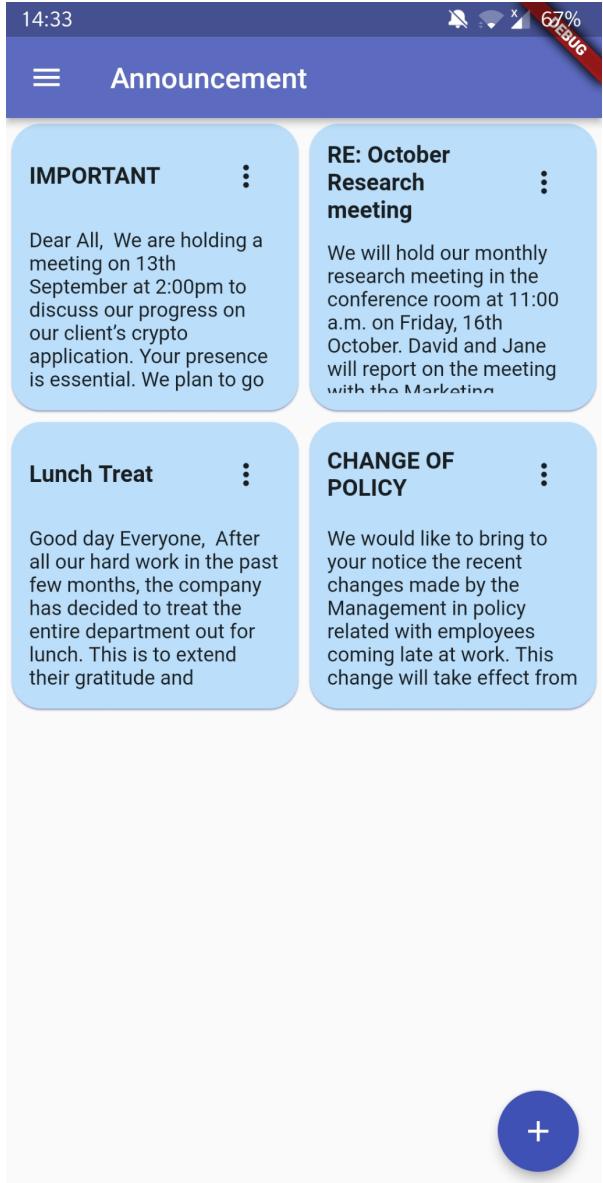
Long press on the selected date. Click on the title of the event that shows up on the timeline. This will navigate to the request form page previously filled up by the user. Click on the lower left pencil icon located at the top right hand corner to edit the selected request.

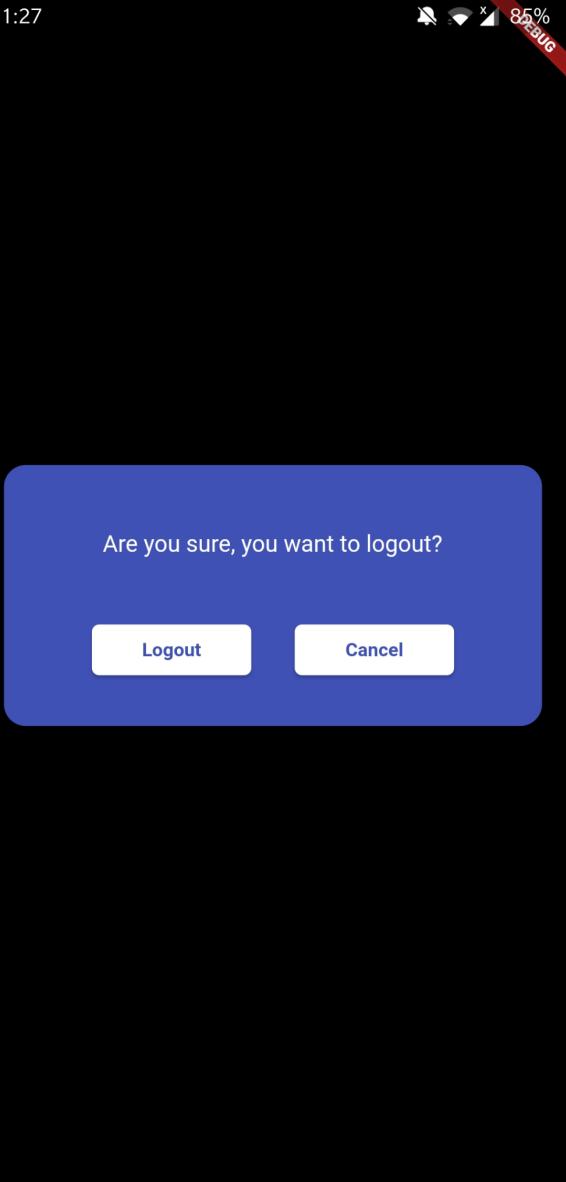
4. To delete the request

Repeat step 3 above, but instead, select the trash icon located at the top right hand corner to delete the request.

| | |
|---|---|
|  <p>To-Do List</p> <p>11:27 85% DEBUG</p> <p>Todo List</p> <p>Meeting Arrangements Set up a meeting with the team regarding Project Binance.</p> <ul style="list-style-type: none"> <input type="checkbox"/> - Define aims and objectives <input type="checkbox"/> - What we need to achieve <input type="checkbox"/> - Delegate tasks <input type="checkbox"/> - Crypto Trend Analysis <p>Review Survey Review previous day survey responses, and credit each account by end of today. Come up with a conclusion.</p> <p>Do analysis regarding Bitcoin Trend Presentation coming up next week, 14/09, 11:00am. Need to start working on a powerpoint slide.</p> <p>+ Todos Completed</p> | <p>The user can make a list of tasks which need to be done.</p> <ol style="list-style-type: none"> To add a new to-do task Click on the '+' button located at the bottom right hand corner of the screen. This button will navigate the user to the to-do form page to fill to-do task details (title, description). *Once saved, it will appear on the 'Todos' tab. To mark a task as completed Select the checkbox located at the left hand side of each to-do task. To view a completed task Click on the tab with the '✓' icon located at the bottom. To unmark a completed task Users can unmark a completed task by unmarking the checkbox. The task will automatically be transferred back to the 'Todos' tab. To edit/delete a task Select the task. This will bring the user to the editing page in which they can save or delete it. To delete a task, click on the trash icon located at the top right hand corner of the app bar. **Another way would be swiping right on a particular task to edit it, where a green edit icon will appear, and swiping left to delete a task, where a red bin icon will come into view. |
|---|---|

| | |
|--|--|
| | |
| <p>FAQ</p>  | <p>Consider a FAQ page to be a simplified version of a customer support page or a product handbook. This is where users can independently find solutions to app-related questions on their own.</p> <p>1. To view the FAQ</p> <p>Select on the question. Users will be brought to the page with the relevant answers.</p>  |

| | |
|--|--|
| | |
| <p>Announcements</p>  | <ol style="list-style-type: none"> To add an announcement: Click on the ‘+’ button located at the bottom right-hand corner on the screen. This button will navigate the user to the announcement form page to fill announcement details (title, message). To edit/delete an announcement Click on the ‘⋮’ icon for the selected announcement to be edited/deleted. Edit: Users will be navigated to the previously-filled announcement page where users can update the details. Delete: The announcement will no longer appear once it has been deleted. |

| | |
|---|---|
| | |
| <p>Logout</p>  A screenshot of a mobile application interface. At the top, there is a black header bar with white icons for time (11:27), signal strength, battery level (85%), and a red 'DEBUG' tag. Below the header is a large blue rectangular dialog box with rounded corners. Inside the dialog, the text "Are you sure, you want to logout?" is displayed in white. At the bottom of the dialog are two white buttons with black text: "Logout" on the left and "Cancel" on the right. | <p>By clicking on the logout button, this allows the user to terminate and leave the app, and they will be welcomed with a sign in page once successfully logged out.</p> |

| | |
|--|--|
| | |
|--|--|

Test Case

Check the Login Functionality

| Test Scenario ID | Login-1 | | Test Case ID | Login-1A | | | | | | |
|--|---|---|--|--|-----------------------|-------------|---|--|--|--|
| Test Case Description | Login – Positive test case | | Test Priority | High | | | | | | |
| Prerequisite | A valid user account | | Post-Requisite | NA | | | | | | |
| Test Execution Steps: | | | | | | | | | | |
| <ul style="list-style-type: none"> • Launch the application • Sign in with a valid user account | | | | | | | | | | |
| S.No | Action | Inputs | Expected Output | Actual Output | Test Emulator Android | Test Result | Test Comments | | | |
| 1 | Launch application | | Login Screen will appear | Login Screen will appear | Pixel 4 API 27 2 | Pass | [Rhonda 12/08/2021 11:45 AM]: Launch successful | | | |
| 2 | Clicking on the sign in button after entering a valid registeredE mail & Password | Email id : david.toh@delta53.com Password: ***** | Login success. Directed to the home page | Login success. Directed to the home page | Pixel 4 API 27 2 | Pass | [Rhonda 12/08/2021 11:46 AM]: Login successful | | | |
| 3 | Clicking on the sign in button after entering a | Email id : delta-53@g mail.com Password: | Login success. Directed to the home page | Login success. Directed to the home | Pixel 4 API 27 2 | Pass | [Rhonda 12/08/2021 11:47 AM]: Login | | | |

| | | | | | | | |
|--|--|-------|--|------|--|--|------------|
| | valid registered Email & Password | ***** | | page | | | successful |
|--|--|-------|--|------|--|--|------------|

| Test Scenario ID | Login-1 | | Test Case ID | Login-1B | | | |
|--|---|---|---|---|-----------------------|-------------|---|
| Test Case Description | Login – Negative test case | | Test Priority | High | | | |
| Prerequisite | NA | | Post-Requisite | NA | | | |
| Test Execution Steps: <ul style="list-style-type: none"> • Launch Application • Sign in with an invalid user account | | | | | | | |
| S.No | Action | Inputs | Expected Output | Actual Output | Test Emulator Android | Test Result | Test Comments |
| 1 | Launch application | | Login Screen will appear | Login Screen will appear | Pixel 4 API 27 2 | Pass | [Rhonda 13/08/2021 11:50 AM]: Launch successful |
| 2 | Enter invalid Email & invalid Password and click the sign in button | Email id : invalid@xyz.com Password: ***** | Alert: Your email address appears to be malformed | Alert: Your email address appears to be malformed | Pixel 4 API 27 2 | Pass | [Rhonda 13/08/2021 11:51 AM]: Invalid login attempt stopped |
| 3 | Enter valid Email & incorrectPassword and click the sign in button | Email id : david.toh@delt a-53 Password: ***** | Alert: Your password is wrong | Alert: Your password is wrong | Pixel 4 API 27 2 | Pass | [Rhonda 13/08/2021 11:52 AM]: Invalid login attempt stopped |

| | | | | | | | |
|---|--|--|---|---|------------------|------|---|
| 4 | Email and password are left blank & click the Sign in button | | Alert: Your email address appears to be malformed | Alert: Your email address appears to be malformed | Pixel 4 API 27 2 | Pass | [Rhonda 13/08/2021 11:55 AM]: Invalid login attempt stopped |
|---|--|--|---|---|------------------|------|---|

Check the Main Page - Navigation Menu

| Test Scenario ID | DrawerMenu-1 | | Test Case ID | DrawerMenu-1 A | | | |
|---|--|---|--|--|-----------------------|-------------|---|
| Test Case Description | DrawerMenu – Positive test case | | Test Priority | Medium | | | |
| Prerequisite | A valid user account | | Post-Requisite | NA | | | |
| Test Execution Steps: | | | | | | | |
| <ul style="list-style-type: none"> ● Launch the application ● Sign in with a valid user account ● Check if the drawer navigation is working as intended ● Will lead to a new page when we click one of the buttons | | | | | | | |
| S.No | Action | Inputs | Expected Output | Actual Output | Test Emulator Android | Test Result | Test Comments |
| 1 | Launch application | | Login Screen will appear | Login Screen will appear | Pixel 4 API 27 2 | Pass | [Maeve 18/08/2021 10:00 AM]: Launch successful |
| 2 | Clicking on the sign in button after entering a valid registeredEmail & Password | Email id : david.toh@delta53.com Password: ***** | Login success. Directed to the home page with a calendar view | Login success. Directed to the home page with a calendar view | Pixel 4 API 27 2 | Pass | [Maeve 19/08/2021 10:02 AM]: Login successful |
| 3 | Clicking on the Navigation Button on the top left hand corner | Click the Navigation Button | A series of different pages will appear | A series of pages will appear | Pixel 4 API 27 2 | Pass | [Maeve 19/08/2021 10:02 AM]: Attempt of displaying different pages successful |
| 4 | Clicking on the Home button | | Directed back to the Home page, which shows the calendar will appear | Directed back to the Home page, which shows the calendar will appear | Pixel 4 API 27 2 | Pass | [Maeve 19/08/2021 10:02 AM]: Attempt of displaying home page successful |

| | | | | | | | |
|---|---|--|---|---|------------------|------|--|
| 5 | Clicking on the Leaves button | | Directed to Leave application page, which shows a monthly calendar will appear | Directed to Leave application page, which shows a monthly calendar will appear | Pixel 4 API 27 2 | Pass | [Maeve 19/08/2021 10:02 AM]: Attempt of displaying leave page successful |
| 6 | Clicking on the Request button | | Directed to Request page, which shows a list of leave requests will appear | Directed to Request page, which shows a list of leave requests will appear | Pixel 4 API 27 2 | Pass | [Maeve 19/08/2021 10:03 AM]: Attempt of displaying leave request page successful |
| 7 | Clicking on the User Profile button | | Directed to User Profile page, which leads to the work stats of the user, displaying the number of leaves remaining, attendance and payslip | Directed to User Profile page, which leads to the work stats of the user, displaying the number of leaves remaining, attendance and payslip | Pixel 4 API 27 2 | Pass | [Maeve 19/08/2021 10:03 AM]: Attempt of displaying user profile page and work stats successful |
| 8 | While on the User Profile Page, click on Personal Data | | Directed to the Personal Data of the user, displaying the photo, position, user ID, email, full name, date of birth, address and phone number | Directed to User Profile page, which leads to the work stats of the user, displaying the number of leaves remaining, attendance and payslip | Pixel 4 API 27 2 | Pass | [Maeve 19/08/2021 10:03 AM]: Attempt of displaying Personal Data page successful |
| 9 | Clicking on the Navigation Button on the top left hand corner, and then click on the To-Do List | | A series of different pages will appear and upon clicking the To-Do List button, user will be directed to To-Do Lists | A series of different pages will appear and upon clicking the To-Do List button, user will be directed to To-Do Lists | Pixel 4 API 27 2 | Pass | [Maeve 19/08/2021 10:03 AM]: Attempt of displaying To-Do List page successful |

| | Page | | Page, displaying a set of tasks that need to be completed | Page, displaying a set of tasks that need to be completed | | | |
|----|-------------------------------------|--|--|--|------------------|------|---|
| 10 | Clicking on the Announcement button | | Directed to Announcement page, where it will show a list of important announcements or messages that need to be taken note of | Directed to Announcement page, where it will show a list of important announcements or messages that need to be taken note of | Pixel 4 API 27 2 | Pass | [Maeve 19/08/2021 10:03 AM]: Attempt of displaying Announcement page successful |
| 11 | Clicking on the FAQ button | | Directed to FAQ page, where it will show a list of Frequently asked questions, so the user is able to navigate through the app | Directed to FAQ page, where it will show a list of Frequently asked questions, so the user is able to navigate through the app | Pixel 4 API 27 2 | Pass | [Maeve 19/08/2021 10:05 AM]: Attempt of displaying FAQ page successful |
| 12 | Clicking on the Log Out button | | Prompted with a message: Are you sure you want to log out? Logout or Cancel | Prompted with a message: Are you sure you want to log out? Logout or Cancel | Pixel 4 API 27 2 | Pass | [Maeve 19/08/2021 10:05 AM]: Attempt of displaying Log out page successful |

Check Calendar Add Event Page

| Test Scenario ID | CalendarEvent-1 | | Test Case ID | CalendarEvent -1A | | | |
|---|---|---|---|---|------------------------|-------------|--|
| Test Case Description | CalendarEvent – Positive test case | | Test Priority | Medium | | | |
| Prerequisite | A valid user account | | Post-Requisite | NA | | | |
| Test Execution Steps: <ul style="list-style-type: none"> • Launch the application • Sign in with a valid user account • Check if the Calendar Add Event/Meeting is working as intended | | | | | | | |
| S.No | Action | Inputs | Expected Output | Actual Output | Test Emulator Andro id | Test Result | Test Comments |
| 1 | Launch application | | Login Screen will appear | Login Screen will appear | Pixel 4 API 27 2 | Pass | [Rhonda 26/08/2021 4:20 PM]: Launch successful |
| 2 | Clicking on the sign in button after entering a valid registered Email & Password | Email id : david.toh@delta53.com Password: ***** | Login success. Directed to the home page with a calendar view | Login success. Directed to the home page with a calendar view | Pixel 4 API 27 2 | Pass | [Rhonda 26/08/2021 4:20 PM]: Login successful |
| 3 | Clicking on the '+' Button on the bottom right hand corner | | Directed to Add Meeting Title Page, where it shows Meeting Title, Date and Duration as well as Description of Meeting | Directed to Add Meeting Title Page, where it shows Meeting Title, Date and Duration as well as Description of Meeting | Pixel 4 API 27 2 | Pass | [Rhonda 26/08/2021 4:21 PM]: Attempt of pressing '+' button successful |
| 4 | Keying in the Meeting Title, Choosing the Date and Time, | | All of the empty fields (Meeting Title and Description of Meeting) can be filled in. Day | All of the empty fields (Meeting Title and Description of Meeting) | Pixel 4 API 27 2 | Pass | [Rhonda 26/08/2021 4:22 PM]: Attempt of keying in all of the |

| | | | | | | | |
|---|---|--|---|---|------------------|------|---|
| | and adding a short description | | and Time can be selected as well | can be filled in. Day and Time can be selected as well | | | information necessary successful |
| 5 | Clicking on the Save button | | Directed to Home page again, where the Meeting/Event the user just entered, appears on the calendar | Directed to Home page again, where the Meeting/Eve nt the user entered, appears on the calendar | Pixel 4 API 27 2 | Pass | [Rhonda 26/08/2021 4:22 PM]: Attempt of displaying event on calendar successful |
| 6 | Clicking on the Event/Meeting itself on the calendar | | It will show a detailed description of what event it is, a short description and the duration (date and time) | It will show a detailed description of what event it is, a short description and the duration (date and time) | Pixel 4 API 27 2 | Pass | [Rhonda 26/08/2021 4:25 PM]: Attempt of displaying detailed info of event successful |
| 7 | Clicking on the Edit button on the top right hand corner and saving it afterwards | | Directed to a page where users can edit the event description and once saved, user will be redirected back to the calendar with the updated event | Directed to a page where users can edit the event description and once saved, a new event is created instead | Pixel 4 API 27 2 | Fail | [Rhonda 26/08/2021 4:25 PM]: Attempt of editing event unsuccessful |
| 8 | Clicking on the Event/Meeting itself on the calendar | | It will show a detailed description of what event it is, a short description and the duration (date and time) | It will show a detailed description of what event it is, a short description and the duration (date and time) | Pixel 4 API 27 2 | Pass | [Rhonda 26/08/2021 4:26 PM]: Attempt of displaying detailed information of event successful |
| 9 | Clicking on the Delete button on the top right hand corner | | Event/Meeting will be deleted. Automatically directed back to the calendar | Event/Meetin g will be deleted. Automatically directed back | Pixel 4 API 27 2 | Fail | [Rhonda 26/08/2021 4:25 PM]: Attempt of deleting |

| | | | | | | | |
|--|--|--|--|--|--|--|------------------|
| | | | view and the event is no longer in sight | to the calendar view and the event is no longer in sight | | | event successful |
|--|--|--|--|--|--|--|------------------|

| Test Scenario ID | | CalendarEvent-1 | | Test Case ID | CalendarEvent -1B | | |
|--|---|---|---|---|------------------------|-------------|--|
| Test Case Description | | CalendarEvent – Negative test case | | Test Priority | Medium | | |
| Prerequisite | | A valid user account | | Post-Requisite | NA | | |
| Test Execution Steps: | | | | | | | |
| ● Launch the application | | | | | | | |
| ● Sign in with a valid user account | | | | | | | |
| ● Check if the Calendar Add Event/Meeting is working as intended | | | | | | | |
| S.N o | Action | Inputs | Expected Output | Actual Output | Test Emulator Andro id | Test Result | Test Comments |
| 1 | Launch application | | Login Screen will appear | Login Screen will appear | Pixel 4 API 27 2 | Pass | [Rhonda 27/08/2021 4:30 PM]: Launch successful |
| 2 | Clicking on the sign in button after entering a valid registered Email & Password | Email id : david.toh@delta53.com Password: ***** | Login success. Directed to the home page with a calendar view | Login success. Directed to the home page with a calendar view | Pixel 4 API 27 2 | Pass | [Rhonda 27/08/2021 4:30 PM]: Login successful |

| | | | | | | | |
|---|---|--|---|---|------------------|------|---|
| 3 | Clicking on the '+' Button on the bottom right hand corner | | Directed to Add Meeting Title Page | Directed to Add Meeting Title Page | | Pass | [Rhonda 27/08/2021 4:31 PM]: Attempt of pressing '+' button successful |
| 4 | Leaving both the Meeting Title and Description of Meeting blank | | 'Title cannot be empty. Please fill in description of Meeting. Cannot be left blank' messages will appear | 'Title cannot be empty. Please fill in description of Meeting. Cannot be left blank' messages will appear | Pixel 4 API 27 2 | Pass | [Rhonda 27/08/2021 4:32 PM]: Invalid Adding a new event attempt stopped |
| 5 | Filling in the Meeting Title but leaving the Description of Meeting blank | | 'Please fill in description of Meeting. Cannot be left blank' message will appear. | 'Please fill in description of Meeting. Cannot be left blank' message will appear | Pixel 4 API 27 2 | Pass | [Rhonda 27/08/2021 4:32 PM]: Invalid Adding a new event attempt stopped |
| 6 | Filling in the Description of Meeting but leaving the Meeting Title blank | | 'Title cannot be empty' message will appear. | 'Title cannot be empty' message will appear. | Pixel 4 API 27 2 | Pass | [Rhonda 27/08/2021 4:32 PM]: Invalid Adding a new event attempt stopped |

Check the Leave Application Page

| Test Scenario ID | LeaveRequest-1 | | Test Case ID | LeaveRequest -1A | | | |
|---|---|---|---|---|-----------------------|-------------|---|
| Test Case Description | LeaveRequest – Positive test case | | Test Priority | Medium | | | |
| Prerequisite | A valid user account | | Post-Requisite | NA | | | |
| Test Execution Steps: | | | | | | | |
| <ul style="list-style-type: none"> ● Launch the application ● Sign in with a valid user account ● Check if the Leave Application is working as intended | | | | | | | |
| S.No | Action | Inputs | Expected Output | Actual Output | Test Emulator Android | Test Result | Test Comments |
| 1 | Launch application | | Login Screen will appear | Login Screen will appear | Pixel 4 API 27 2 | Pass | [Rizqi 05/09/2021 05:00 PM]: Launch successful |
| 2 | Clicking on the sign in button after entering a valid registered Email & Password | Email id : david.toh@delta53.com Password: ***** | Login success. Directed to the home page with a calendar view | Login success. Directed to the home page with a calendar view | Pixel 4 API 27 2 | Pass | [Rizqi 05/09/2021 05:00 PM]: Login successful |
| 3 | Clicking on the Navigation Button on the top left hand corner | Click the Navigation Button | A series of different pages will appear | A series of pages will appear | Pixel 4 API 27 2 | Pass | [Rizqi 05/09/2021 05:00 PM]: Attempt of displaying different pages successful |
| 4 | Clicking on the Leaves Button | | Directed to the Leave Application Calendar, with a monthly view. Today's date will be circled in purple | Directed to the Leave Application Calendar, with a monthly view. Today's date will be | Pixel 4 API 27 2 | Pass | [Rizqi 05/09/2021 05:00 PM]: Attempt of keying displaying leave application |

| | | | | | | | |
|---|---|--|--|--|------------------|------|--|
| | | | | circled in purple | | | successful |
| 3 | Clicking on the '+' Button on the bottom right hand corner | | Directed to Leave Application Page. Showing the user to key in Type of Leave Request, Choosing the duration (date and time) and a Short Description of Leave | Directed to Leave Application Page. Showing the user to key in Type of Leave Request, Choosing the duration (date and time) and a Short Description of Leave | Pixel 4 API 27 2 | Pass | [Rizqi 05/09/2021 05:01 PM]: Attempt of pressing '+' button successful |
| 4 | Filling in the Type of Leave Request, Duration of Leave, and adding a Short Description | | All of the empty fields (Type of Leave Request & Description) can be filled in. Date and Time can be selected as well. | All of the empty fields (Type of Leave Request & Description) can be filled in. Date and Time can be selected as well. | Pixel 4 API 27 2 | Pass | [Rizqi 05/09/2021 05:02 PM]: Attempt of keying in all of the information successful |
| 5 | Clicking on the Submit button | | Directed to Leave Application Page, where the Requested Leave appears on the calendar. This will be indicated by little dots. | Directed to Leave Application Page, where the Requested Leave appears on the calendar. This will be indicated by little dots. | Pixel 4 API 27 2 | Pass | [Rizqi 05/09/2021 05:03 PM]: Attempt of displaying Leave Request on calendar successful |
| 6 | Click and Long Press on the Leave Application itself on the calendar | | It will give a brief information regarding the Leave Request, showing the day and time in a horizontal view. | It will give a brief information regarding the Leave Request, showing the day and time in a horizontal | Pixel 4 API 27 2 | Pass | [Rizqi 05/09/2021 05:03 PM]: Attempt of displaying detailed info of Leave Request successful |

| | | | | | | | |
|----|---|--|---|---|------------------|------|---|
| | | | | view. | | | |
| 7 | Clicking on the selected Leave Request | | It will show a detailed information of what Leave it is, a short Description of the Leave and the duration (date and time) | It will show a detailed information of what Leave it is, a short Description of the Leave and the duration (date and time) | Pixel 4 API 27 2 | Pass | [Rizqi 05/09/2021 05:04 PM]: Attempt of displaying detailed information of event successful |
| 8 | Clicking on the Edit button on the top right hand corner and saving it afterwards | | Directed to a page where users can edit the Leave description and once saved, user will be redirected back to the calendar with the updated Leave Request | Directed to a page where users can edit the Leave description and once saved, user will be redirected back to the calendar with the updated Leave Request | Pixel 4 API 27 2 | Pass | [Rizqi 05/09/2021 05:04 PM]: Attempt of editing Leave Request successful |
| 9 | Clicking on the Delete button on the top right hand corner | | The Selected Leave Request will be deleted. Automatically directed back to the calendar view and the Leave Request is no longer in sight | The Selected Leave Request will be deleted. Automaticall y directed back to the calendar view and the Leave Request is no longer in sight | Pixel 4 API 27 2 | Pass | [Rizqi 05/09/2021 05:05 PM]: Attempt of deleting Leave Request successful |
| 10 | Clicking on the Calendar and Scrolling left or right | | Displays a different month every time user swipes left or right | Displays a different month every time user swipes left or right | Pixel 4 API 27 2 | | [Rizqi 05/09/2021 05:05 PM]: Attempt of displaying different months of the |

| | | | | | | | |
|--|--|--|--|--|--|--|---------------------|
| | | | | | | | calendar successful |
|--|--|--|--|--|--|--|---------------------|

| Test Scenario ID | LeaveRequest-1 | | Test Case ID | LeaveRequest -1B | | | | | |
|---|---|---|---|---|-----------------------|-------------|--|--|--|
| Test Case Description | LeaveRequest – Negative test case | | Test Priority | Medium | | | | | |
| Prerequisite | A valid user account | | Post-Requisite | NA | | | | | |
| Test Execution Steps: | | | | | | | | | |
| <ul style="list-style-type: none"> ● Launch the application ● Sign in with a valid user account ● Check if the Leave Application is working as intended | | | | | | | | | |
| S.No | Action | Inputs | Expected Output | Actual Output | Test Emulator Android | Test Result | Test Comments | | |
| 1 | Launch application | | Login Screen will appear | Login Screen will appear | Pixel 4 API 27 2 | Pass | [Rizqi 06/09/2021 05:15 PM]: Launch successful | | |
| 2 | Clicking on the sign in button after entering a valid registered Email & Password | Email id : david.toh@delta53.com Password: ***** | Login success. Directed to the home page with a calendar view | Login success. Directed to the home page with a calendar view | Pixel 4 API 27 2 | Pass | [Rizqi 06/09/2021 05:16 PM]: Login successful | | |

| | | | | | | | |
|---|--|-----------------------------|--|--|------------------|------|--|
| 3 | Clicking on the Navigation Button on the top left hand corner | Click the Navigation Button | A series of different pages will appear | A series of pages will appear | Pixel 4 API 27 2 | Pass | [Rizqi 06/09/2021 05:16 PM]: Attempt of displaying different pages successful |
| 4 | Clicking on the Leaves Button | | Directed to the Leave Application Calendar, with a monthly view. Today's date will be circled in purple | Directed to the Leave Application Calendar, with a monthly view. Today's date will be circled in purple | Pixel 4 API 27 2 | Pass | [Rizqi 06/09/2021 05:16 PM]: Attempt of keying displaying leave application successful |
| 5 | Clicking on the '+' Button on the bottom right hand corner | | Directed to Leave Application Page. Showing the user to key in Type of Leave Request, Choosing the duration (date and time) and a Short Description of Leave | Directed to Leave Application Page. Showing the user to key in Type of Leave Request, Choosing the duration (date and time) and a Short Description of Leave | Pixel 4 API 27 2 | Pass | [Rizqi 06/09/2021 05:17 PM]: Attempt of pressing '+' button successful |
| 6 | Leaving both the Type of Leave Request Title and Description of Leave Request Meeting blank | | 'Please fill in TYPE OF LEAVE. Please fill in description of leave. Cannot be left blank' message will appear | 'Please fill in TYPE OF LEAVE. Please fill in description of leave. Cannot be left blank' message will appear | Pixel 4 API 27 2 | Pass | [Rizqi 06/09/2021 05:17 PM]: Invalid Adding a new event attempt stopped |

| | | | | | | | |
|---|---|--|--|---|------------------------|------|--|
| 7 | Filling in the Type of Leave Request but leaving the Descriptio n of Leave blank | | 'Please fill in description of Leave. Cannot be left blank' message will appear. | 'Please fill in description of Leave. Cannot be left blank' message will appear. | Pixel 4 API 27 2 | Pass | [Rizqi 06/09/2021 05:17 PM]: Invalid Adding a new event attempt stopped |
| 8 | Filling in the Descriptio n of Leave but leaving the Type Of Leave Request blank | | 'Please fill in TYPE OF LEAVE. Cannot be left blank' message will appear. | 'Please fill in TYPE OF LEAVE. Cannot be left blank' message will appear. | Pixel 4 API 27 2 | Pass | [Rizqi 06/09/2021 05:18 PM]: Invalid Adding a new event attempt stopped |

Check the Request Page

| Test Scenario ID | RequestPage-1 | | Test Case ID | RequestPage-1A | | | |
|--|---|---|--|---|-----------------------|-------------|---|
| Test Case Description | RequestPage – Positive test case | | Test Priority | Medium | | | |
| Prerequisite | A valid user account | | Post-Requisite | NA | | | |
| Test Execution Steps: | | | | | | | |
| <ul style="list-style-type: none"> ● Launch the application ● Sign in with a valid user account ● Check if the Request Page is working as intended | | | | | | | |
| S.No | Action | Inputs | Expected Output | Actual Output | Test Emulator Android | Test Result | Test Comments |
| 1 | Launch application | | Login Screen will appear | Login Screen will appear | Pixel 4 API 27 2 | Pass | [Leonard 04/09/2021 08:00 PM]: Launch successful |
| 2 | Clicking on the sign in button after entering a valid registered Email & Password | Email id : david.toh@delta53.com Password: ***** | Login success. Directed to the home page with a calendar view | Login success. Directed to the home page with a calendar view | Pixel 4 API 27 2 | Pass | [Leonard 04/09/2021 08:00 PM]: Login successful |
| 3 | Clicking on the Navigation Button on the top left hand corner | Click the Navigation Button | A series of different pages will appear | A series of pages will appear | Pixel 4 API 27 2 | Pass | [Leonard 04/09/2021 08:00 PM]: Attempt of displaying different pages successful |
| 4 | Clicking on the Request Button | | Directed to the Request Page. Showing all the Leave Requests made and their status | Directed to the Request Page. Showing all the Leave Requests available and their status | Pixel 4 API 27 2 | Pass | [Leonard 04/09/2021 08:00 PM]: Attempt of keying displaying Request Page |

| | | | | | | | |
|---|--------------------------------------|--|---|---|------------------|------|--|
| | | | | | | | successful |
| 5 | Clicking on a selected Leave Request | | Displays a pop-out message regarding the Type of Leave, Duration of Leave and Description of Leave. | Displays a pop-out message regarding the Type of Leave, Duration of Leave and Description of Leave. | Pixel 4 API 27 2 | Pass | [Leonard 04/09/2021 08:00 PM]: Attempt of displaying more information regarding Leave successful |
| 6 | Clicking the OK button | | Directs back to the Request Page | Directs back to the Request Page | Pixel 4 API 27 2 | Pass | [Leonard 04/09/2021 08:00 PM]: Attempt of clicking OK button is successful |

Check the User Profile Page

| Test Scenario ID | UserProfilePage-1 | | Test Case ID | UserProfile-1A | | | |
|--|---|---|---|---|-----------------------|-------------|--|
| Test Case Description | UserProfilePage – Positive test case | | Test Priority | Medium | | | |
| Prerequisite | A valid user account | | Post-Requisite | NA | | | |
| <p>Test Execution Steps:</p> <ul style="list-style-type: none"> • Launch the application • Sign in with a valid user account • Check if the User Profile Page is working as intended | | | | | | | |
| S.No | Action | Inputs | Expected Output | Actual Output | Test Emulator Android | Test Result | Test Comments |
| 1 | Launch application | | Login Screen will appear | Login Screen will appear | Pixel 4 API 27 2 | Pass | [Daniel 01/09/2021 06:00 PM]: Launch successful |
| 2 | Clicking on the sign in button after entering a valid registered Email & Password | Email id : david.toh@delta53.com Password: ***** | Login success. Directed to the home page with a calendar view | Login success. Directed to the home page with a calendar view | Pixel 4 API 27 2 | Pass | [Daniel 01/09/2021 06:00 PM]: Login successful |
| 3 | Clicking on the Navigation Button on the top left hand corner | Click the Navigation Button | A series of different pages will appear | A series of pages will appear | Pixel 4 API 27 2 | Pass | [Daniel 01/09/2021 06:01 PM]: Attempt of displaying different pages successful |
| 4 | Clicking on the User Profile button | | Directed to User Profile page, which leads to the work stats of the user, displaying the number of leaves | Directed to User Profile page, which leads to the work stats of the user, displaying the number of leaves | Pixel 4 API 27 2 | Pass | [Daniel 01/09/2021 06:02 PM]: Attempt of displaying user profile page and work stats |

| | | | | | | | |
|---|---|--|---|--|------------------------|------|--|
| | | | remaining, attendance and payslip | remaining, attendance and payslip | | | successful |
| 5 | While on the User Profile Page, click on Personal Data | | Directed to the Personal Data of the user, displaying the photo, position, user ID, email, full name, date of birth, address and phone number | Directed to User Profile page, which leads to the work stats of the user, displaying the number of leaves remaining, attendance and payslip | Pixel 4 API 27 2 | Pass | [Daniel 01/09/2021 06:02 PM]: Attempt of displaying Personal Data page successful |

Check the To-Do List Page

| Test Scenario ID | ToDoListPage-1 | | Test Case ID | ToDoListPage-1A | | | |
|--|---|---|--|--|-----------------------|-------------|---|
| Test Case Description | ToDoListPage – Positive test case | | Test Priority | Medium | | | |
| Prerequisite | A valid user account | | Post-Requisite | NA | | | |
| <p>Test Execution Steps:</p> <ul style="list-style-type: none"> • Launch the application • Sign in with a valid user account • Check if the To-Do List Page is working as intended | | | | | | | |
| S.No | Action | Inputs | Expected Output | Actual Output | Test Emulator Android | Test Result | Test Comments |
| 1 | Launch application | | Login Screen will appear | Login Screen will appear | Pixel 4 API 27 2 | Pass | [Rizqi 03/09/2021 07:30 AM]: Launch successful |
| 2 | Clicking on the sign in button after entering a valid registered Email & Password | Email id : david.toh@delta53.com Password: ***** | Login success. Directed to the home page with a calendar view | Login success. Directed to the home page with a calendar view | Pixel 4 API 27 2 | Pass | [Rizqi 03/09/2021 07:30 AM]: Login successful |
| 3 | Clicking on the Navigation Button on the top left hand corner | Click the Navigation Button | A series of different pages will appear | A series of pages will appear | Pixel 4 API 27 2 | Pass | [Rizqi 03/09/2021 07:30 AM]: Attempt of displaying different pages successful |
| 4 | Clicking on the To-Do List button | | Directed to To-Do Lists Page, displaying a set of tasks that need to be completed. Shows two tabs on the | Directed to To-Do Lists Page, displaying a set of tasks that need to be completed. Shows two | Pixel 4 API 27 2 | Pass | [Rizqi 03/09/2021 07:30 AM]: Attempt of displaying To-Do List Page successful |

| | | | | | | | |
|---|--|--|--|--|------------------|------|--|
| | | | bottom, Todos & Completed | tabs on the bottom, Todos & Completed | | | |
| 5 | While on the To-Do List Page, click on 'Completed' Tab on the bottom | | Directs to a new page where it shows a list of completed tasks. If there are no completed tasks, a small message will appear on the white background 'No completed tasks.' | Directs to a new page where it shows a list of completed tasks. If there are no completed tasks, a small message will appear on the white background 'No completed tasks.' | Pixel 4 API 27 2 | Pass | [Rizqi 03/09/2021 07:30 AM]: Attempt of displaying 'Completed' To-Do List Data page successful |
| 6 | While on the To-Do List Page, click on 'Todos' Tab on the bottom | | Directs back to the page where it shows a list of uncompleted tasks. If there are no tasks being added, a small message will appear on the while background 'No todos'. | Directs back to the page where it shows a list of uncompleted tasks. If there are no tasks being added, a small message will appear on the while background 'No todos'. | Pixel 4 API 27 2 | | [Rizqi 03/09/2021 07:30 AM]: Attempt of displaying 'Todos' To-Do List Data page successful |
| 7 | Marking a task completed, by clicking on the box situated just beside the task | | Task will be moved to the 'Completed' tab and a small pop up message will appear in the bottom, 'Task Completed' | Task will be moved to the 'Completed' tab and a small pop up message will appear in the bottom, 'Task Completed' | Pixel 4 API 27 2 | | [Rizqi 03/09/2021 07:31 AM]: Attempt of checking off a completed task successful |
| 8 | Clicking on the ticked box of a completed | | Task will be transferred back to the 'Todos' tab | Task will be transferred back to the 'Todos' tab | Pixel 4 API 27 2 | | [Rizqi 03/09/2021 07:32 AM]: Attempt of |

| | | | | | | | |
|----|--|--|---|---|------------------|--|---|
| | task in the 'Completed' tab | | and a small pop up message will appear in the bottom, 'Task Marked Incomplete' | and a small pop up message will appear in the bottom, 'Task Marked Incomplete' | | | checking off an incomplete task successful |
| 9 | Clicking on the '+' button on the bottom right hand side | | Directs to a new page to add a new todo/task. Two empty fields to key in title and description of task | Directs to a new page to add a new todo/task. Two empty fields to key in title and description of task | Pixel 4 API 27 2 | | [Rizqi 03/09/2021 07:35 AM]: Attempt of clicking the '+' button successful |
| 10 | Clicking on the save button | | A new task will be added to the 'Todos' tab and will appear instantly | A new task will be added to the 'Todos' tab and will appear instantly | Pixel 4 API 27 2 | | [Rizqi 03/09/2021 07:37 AM]: Attempt of clicking on save button Successful and a new task was added |
| 11 | Clicking on a selected task | | Directs users to a page where they are able to navigate to the previous page (< button), edit by clicking save or delete the task | Directs users to a page where they are able to navigate to the previous page (< button), edit by clicking save or delete the task | Pixel 4 API 27 2 | | [Rizqi 03/09/2021 07:38 AM]: Attempt of clicking on a selected task successful |
| 12 | Clicking on the Save Button | | Selected task will be updated accordingly and user will be directed back automatically to the To-Do List Page, showing the | Selected task will be updated accordingly and user will be directed back automatically to the To-Do List Page, showing the | Pixel 4 API 27 2 | | [Rizqi 03/09/2021 07:40 AM]: Attempt of saving a task successful |

| | | | updated task | updated task | | | |
|----|-------------------------------------|--|--|--|------------------|--|--|
| 13 | Clicking on the Bin Button (Delete) | | Selected task will be deleted accordingly and user will be directed back automatically to the To-Do List Page, with the deleted task nowhere to be found | Selected task will be deleted accordingly and user will be directed back automatically to the To-Do List Page, with the deleted task nowhere to be found | Pixel 4 API 27 2 | | [Rizqi 03/09/2021 07:40 AM]: Attempt of deleting a task successful |
| 14 | Swiping right on a selected task | | A green Edit button will appear, giving users the option to edit the task | A green Edit button will appear, giving users the option to edit the task | Pixel 4 API 27 2 | | [Rizqi 03/09/2021 07:40 AM]: Attempt of swiping right and a green Edit button emerged was successful |
| 15 | Swiping left on a selected task | | A red Delete button will appear, giving users the option to delete the task | A red Delete button will appear, giving users the option to delete the task | Pixel 4 API 27 2 | | [Rizqi 03/09/2021 07:40 AM]: Attempt of swiping right and a red Delete button emerged was successful |

| Test Scenario ID | ToDoListPage-1 | | Test Case ID | ToDoListPage-1B | | | |
|--|---|---|---|---|-----------------------|-------------|---|
| Test Case Description | ToDoListPage– Negative test case | | Test Priority | Medium | | | |
| Prerequisite | A valid user account | | Post-Requisite | NA | | | |
| Test Execution Steps: <ul style="list-style-type: none"> • Launch the application • Sign in with a valid user account • Check if the To-Do List Page is working as intended | | | | | | | |
| S.No | Action | Inputs | Expected Output | Actual Output | Test Emulator Android | Test Result | Test Comments |
| 1 | Launch application | | Login Screen will appear | Login Screen will appear | Pixel 4 API 27 2 | Pass | [Rizqi 05/09/2021 07:30 AM]: Launch successful |
| 2 | Clicking on the sign in button after entering a valid registered Email & Password | Email id : david.toh@delta53.com Password: ***** | Login success. Directed to the home page with a calendar view | Login success. Directed to the home page with a calendar view | Pixel 4 API 27 2 | Pass | [Rizqi 05/09/2021 08:50 AM]: Login successful |
| 3 | Clicking on the Navigation Button on the top left hand corner | Click the Navigation Button | A series of different pages will appear | A series of pages will appear | Pixel 4 API 27 2 | Pass | [Rizqi 05/09/2021 08:51 AM]: Attempt of displaying different pages successful |

| | | | | | | | |
|---|--|--|--|--|------------------|------|---|
| 4 | Clicking on the To-Do List button | | Directed to To-Do Lists Page, displaying a set of tasks that need to be completed. Shows two tabs on the bottom, Todos & Completed | Directed to To-Do Lists Page, displaying a set of tasks that need to be completed. Shows two tabs on the bottom, Todos & Completed | Pixel 4 API 27 2 | Pass | [Rizqi 05/09/2021 08:52 AM]: Attempt of displaying To-Do List Page successful |
| 5 | Clicking on the '+' button on the bottom right hand side | | Directed to a page to add a new todo/task. Two empty fields to key in title and description of task | Directed to a page to add a new todo/task. Two empty fields to key in title and description of task | Pixel 4 API 27 2 | | [Rizqi 05/09/2021 08:52 AM]: Attempt of displaying page to key in task successful |
| 6 | Leaving the two fields EMPTY (Title & Description) and clicking the Save button afterwards | | 'The title cannot be empty' message will appear | 'The title cannot be empty' message will appear | Pixel 4 API 27 2 | | [Rizqi 05/09/2021 08:55 AM]: Invalid Adding a new task attempt stopped |

Check the Announcement Page

| Test Scenario ID | Announcement-1 | | Test Case ID | Announcement-1A | | | |
|--|---|---|--|---|-----------------------|-------------|--|
| Test Case Description | Announcement – Positive test case | | Test Priority | Medium | | | |
| Prerequisite | A valid user account | | Post-Requisite | NA | | | |
| Test Execution Steps: <ul style="list-style-type: none"> • Launch the application • Sign in with a valid user account • Check if the Announcement Page is working as intended | | | | | | | |
| S.No | Action | Inputs | Expected Output | Actual Output | Test Emulator Android | Test Result | Test Comments |
| 1 | Launch application | | Login Screen will appear | Login Screen will appear | Pixel 4 API 27 2 | Pass | [Maeve 25/08/2021 9:30 PM]: Launch successful |
| 2 | Clicking on the sign in button after entering a valid registered Email & Password | Email id : david.toh@delta53.com Password: ***** | Login success. Directed to the home page with a calendar view | Login success. Directed to the home page with a calendar view | Pixel 4 API 27 2 | Pass | [Maeve 25/08/2021 9:30 PM]: Login successful |
| 3 | Clicking on the Navigation Button on the top left hand corner | Click the Navigation Button | A series of different pages will appear | A series of pages will appear | Pixel 4 API 27 2 | Pass | [Maeve 25/08/2021 9:30 PM]: Attempt of displaying different pages successful |
| 4 | Clicking on the Announcement button | | Directed to Announcement page, where it will show a list of important announcements or messages that | Directed to Announcement page, where it will show a list of important announcements or messages | Pixel 4 API 27 2 | Pass | [Maeve 25/08/2021 9:31 PM]: Attempt of displaying Announcement page successful |

| | | | need to be taken into account | that need to be taken into account | | | |
|----|--|--|--|--|------------------|------|--|
| 5 | Clicking on the '+' Button on the bottom right hand corner | | Directed to Add New Announcement pop-up page, where user can key in Title and Message | Directed to Add New Announcement pop-up page, where user can key in Title and Message | Pixel 4 API 27 2 | Pass | [Maeve 25/08/2021 9:35 PM]: Attempt of pressing '+' button successful |
| 6 | Keying in Title and Description | | Both the empty fields can be filled in | Both the empty fields can be filled in | Pixel 4 API 27 2 | Pass | [Maeve 25/08/2021 9:37 PM]: Attempt of keying in all of the information necessary successful |
| 8 | Clicking on the Add button | | Directed back to the Announcement page again, and the newly added announcement will be visible | Directed back to the Announcement page again, and the newly added announcement will be visible | Pixel 4 API 27 2 | Pass | [Maeve 25/08/2021 9:38 PM]: Attempt of displaying announcement successful |
| 8 | Clicking on the '...' show menu button of an announcement | | Revealing two options, whether to edit or delete the selected announcement | Revealing two options, whether to edit or delete the selected announcement | Pixel 4 API 27 2 | Pass | [Maeve 25/08/2021 9:38 PM]: Attempt of displaying announcement successful |
| 9 | Clicking edit | | Directed to the update announcement pop-up page, where users can update the title and message | Directed to the update announcement pop-up page, where users can update the title and message | Pixel 4 API 27 2 | Pass | [Maeve 25/08/2021 9:45 PM]: Attempt of pressing the edit button successful |
| 10 | Clicking update | | Selected announcement | Selected announcement | Pixel 4 API | Pass | [Maeve |

| | | | | | | | |
|----|-----------------|--|--|--|------------------|------|--|
| | | | t will be updated and the newly updated announcement will be shown on the announcement page | nt will be updated and the newly updated announcement will be shown on the announcement page | 27 2 | | 25/08/2021 9:46 PM]: Attempt of pressing the update button successful |
| 11 | Clicking delete | | Selected announcement will be deleted, and it will no longer appear on the announcement page | Selected announcement will be deleted, and it will no longer appear on the announcement page | Pixel 4 API 27 2 | Pass | [Maeve 25/08/2021 9:47 PM]: Attempt of pressing the delete button successful |

Check the FAQ Page

| Test Scenario ID | FAQ-1 | | Test Case ID | FAQ-1A | | | | | | |
|--|---|---|---|---|------------------------|-------------|--|--|--|--|
| Test Case Description | FAQ – Positive test case | | Test Priority | Medium | | | | | | |
| Prerequisite | A valid user account | | Post-Requisite | NA | | | | | | |
| Test Execution Steps: | | | | | | | | | | |
| <ul style="list-style-type: none"> ● Launch the application ● Sign in with a valid user account ● Check if the FAQ Page is working as intended | | | | | | | | | | |
| S.No | Action | Inputs | Expected Output | Actual Output | Test Emulator Andro id | Test Result | Test Comments | | | |
| 1 | Launch application | | Login Screen will appear | Login Screen will appear | Pixel 4 API 27 2 | Pass | [Rhonda 13/08/2021 11:45 AM]: Launch successful | | | |
| 2 | Clicking on the sign in button after entering a valid registered Email & Password | Email id : david.toh@delta53.com Password: ***** | Login success. Directed to the home page with a calendar view | Login success. Directed to the home page with a calendar view | Pixel 4 API 27 2 | Pass | [Rhonda 13/08/2021 11:45 AM]: Login successful | | | |
| 3 | Clicking on the Navigation Button on the top left hand corner | Click the Navigation Button | A series of different pages will appear | A series of pages will appear | Pixel 4 API 27 2 | Pass | [Rhonda 13/08/2021 11:46 AM]: Attempt of displaying different pages successful | | | |
| 4 | Clicking on the FAQ button | | Directed to FAQ page, where it will show a list of Frequently asked questions, so the user will be able | Directed to FAQ page, where it will show a list of Frequently asked questions, so the user will be able | Pixel 4 API 27 2 | Pass | [Rhonda 13/08/2021 11:46 AM]: Attempt of displaying FAQ page successful | | | |

| | | | | | | | |
|---|--|--|--|--|------------------|--|---|
| | | | choose a question pertaining to his/her concerns. FAQ has been divided into two, Contact Support & How To Use | choose a question pertaining to his/her concerns. FAQ has been divided into two, Contact Support & How To Use | | | |
| 5 | Clicking on a question | | The question along with the answer to that question will be displayed | The question along with the answer to that question will be displayed | Pixel 4 API 27 2 | | [Rhonda 13/08/2021 11:46 AM]: Attempt of clicking on any of the FAQs successful |
| 6 | Clicking the (<) back button on the top left hand corner | | Directed back to the main FAQ page | Directed back to the main FAQ page | Pixel 4 API 27 2 | | [Rhonda 13/08/2021 11:46 AM]: Attempt of clicking on the back button successful |

Check the Logout Page

| Test Scenario ID | Logout-1 | | Test Case ID | Logout-1A | | | | | | |
|---|---|---|--|---|------------------------|-------------|---|--|--|--|
| Test Case Description | Logout – Positive test case | | Test Priority | Medium | | | | | | |
| Prerequisite | A valid user account | | Post-Requisite | NA | | | | | | |
| Test Execution Steps: | | | | | | | | | | |
| <ul style="list-style-type: none"> ● Launch the application ● Sign in with a valid user account ● Check if the Logout Page is working as intended | | | | | | | | | | |
| S.No | Action | Inputs | Expected Output | Actual Output | Test Emulator Andro id | Test Result | Test Comments | | | |
| 1 | Launch application | | Login Screen will appear | Login Screen will appear | Pixel 4 API 27 2 | Pass | [Daniel 18/08/2021 3:45 PM]: Launch successful | | | |
| 2 | Clicking on the sign in button after entering a valid registered Email & Password | Email id : david.toh@delta53.com Password: ***** | Login success. Directed to the home page with a calendar view | Login success. Directed to the home page with a calendar view | Pixel 4 API 27 2 | Pass | [Daniel 18/08/2021 3:45 PM]: Login successful | | | |
| 3 | Clicking on the Navigation Button on the top left hand corner | Click the Navigation Button | A series of different pages will appear | A series of pages will appear | Pixel 4 API 27 2 | Pass | [Daniel 18/08/2021 3:46 PM]: Attempt of displaying different pages successful | | | |
| 4 | Clicking on the Log Out button | | Prompted with a message: 'Are you sure you want to log out?' User can select either Logout or Cancel | Prompted with a message: 'Are you sure you want to log out?' User can select either | Pixel 4 API 27 2 | Pass | [Daniel 18/08/2021 3:48 PM]: Attempt of displaying Logout page successful | | | |

| | | | | Logout or Cancel | | | |
|---|--------------------|--|--|--|------------------|--|--|
| 5 | Clicking on Cancel | | Directed back to the previous page user was in | Directed back to the previous page user was in | Pixel 4 API 27 2 | | [Daniel 18/08/2021 3:49 PM]: Attempt of clicking on the cancel button successful |
| 6 | Clicking on Logout | | Directed back to the main Login screen | Directed back to the main Login screen | Pixel 4 API 27 2 | | [Daniel 18/08/2021 3:50 PM]: Attempt of clicking on the back button successful |

Conclusion and Future Work

Delta was created to assist an organisation in HR and employee related tasks as well as facilitate efficient communication between the two parties. It is able to serve its role as an all-in-one management system with many functionalities that can support the roles of both.

In the future, Delta can be further developed to provide an even more customised experience for an organisation - for example, a reward system or a communication system that can further boost efficiencies in an organisation.

At some point in the future, we would like our app to boost employee engagement. Our mobile app will serve as a channel for connecting employees with the HR department. Using our app, HR can better meet the needs of their employees if HR knows what their employees want. It can assist HR in obtaining feedback by having employees complete questionnaires and surveys, as well as providing them with an environment in which they can participate more fully through the use of activities. As a result, HR can measure employee engagement and make an effort to engage those who are still disconnected.

Also, employees can use mobile applications to track their progress toward their goals and receive regular and ongoing feedback. We offer the option of rewarding a colleague for exceptional collaboration with a badge entitlement or a vote rating. We can also give the option of rewarding a colleague for exceptional collaboration with a vote rating or badge entitlement. Employees can express gratitude, appreciation, and recognition for each other's hard work in this manner, positively affecting each other's performance results and quickly straightening out the team work at no cost.

As we all know, the workforce is always on the move, and in a global operation, it could be located anywhere. Subsequently, we aim to have more effective communication internally. HR managers can contact their employees at any time and from any location by using a mobile app. Employers can keep them informed of what is going on, reach out to them via chat features in mobile HR software, and encourage them to communicate if there are any issues. In general, mobile applications enable us to communicate at any time, and employers will no longer have to wait for their employees to log in to their computer or laptop in order for them to connect with one another. Employers can send messages to employees while they are on the road or at work using our HR mobile app. It is critical to empower employees in order to improve employee engagement. An engaged employee and an HR professional who is willing to listen and solve problems will always assist the company in becoming profitable.

Our app is targeted towards SMEs or small businesses who are starting to venture out into the open. As their businesses grow, their number of employees will grow as well. We will be rolling out our app in both iOS and Android, which will be available to download in both AppStore and Google Store, and it will be free of charge. This will give an equal playing field for all forms and kinds of businesses who would want to try to automate and simplify basic employee and HR

processes. In the process, we would collect feedback from our consumers and we will add more features along the way, and provide timely updates to fix bugs and issues encountered.

In terms of functionalities, we may lose out compared to those big players such as WorkDay, Connecteam, Monday.com, who have dominated the market for quite some time now, but as mentioned above, we are targeting our app for small businesses to improve their overall HR & employees processes. As we are small scale and not backed by any investors, we will be more willing to listen to feedback from our users, gather what they have to say and make the necessary improvements along the way. This includes adding in more features, enhancing the security of our users' data and fixing bugs to smoothen the overall user experience.

As more people use their smartphones for all aspects of their lives, HR professionals will recognise that mobile technology can have a huge impact and significantly improve their services to employees while also freeing up their time to focus on other more crucial issues like planning and implementing innovative projects that ensure a competitive edge for their company.

Mobile technology is reshaping the way people connect with one another, as well as the way businesses are formed, organised, and managed. The digitalisation of mobile apps provides new opportunities for businesses to reform and improve basic HR processes, making them more functional by increasing productivity and improving employee satisfaction, resulting in a giant leap forward in human resource management.

In conclusion, we have created Delta as a simple and on-the-go mobile platform for employees and the HR team in an organisation. We have come up with this app to achieve our aim and are proud to present you with our final product.

Personal Reflection

It has indeed been a roller coaster journey for me. Having to work on an app for the past two months has been an eye-opening experience. I am a slow learner, and programming is not my forte, especially when it's rushed. Learning a new language and trying to get a grasp of it within a short time frame, was challenging indeed. Watching lots of video tutorials on flutter/dart and slowly getting the hang of it. I was mainly working on two functions, however, I was stuck and lost on how to get the data to be stored in the firebease. In the end, as time was running out, Maeve jumped in to help and she was able to get it done in no time. Data was successfully stored in firebase. I also managed to finish up my last to-do list function and got it working, and most importantly it was able to do CRUD (create, read, update, delete) operations of firebase, and I manage to do it on my own. Overall, my teammates have been helpful and enjoyable to work with. Maeve was extremely skillful in coding, and she always finds a way to debug our codes when there are issues. Leonard, being the main designer, is always on the ball, trying his best to do tasks assigned to him, and did not fail to deliver at the end. Rhonda is always the one who takes initiative and makes sure that our tasks are done on time before each sprint, and is always willing to help. Daniel was a lot better this time round, compared to the first coursework, where now he takes the initiative, participates in discussion, finished his tasks and did a bit of the debugging. Overall, it has been a great journey working with this team.

References

Desklessworkforce2018.com. 2018. The Rise Of Deskless Workforce. [online] Available at: <<http://desklessworkforce2018.com/>> [Accessed 2 September 2021].