IM3080 Design and Innovation Project (AY20/21 Semester 1) Project Report

Title: The Friendly Planner
Github: https://github.com/ChenZengYao/LifeBalance-flutter
Submitted by: DIP Group 6
Supervisor: Yakoob Siyal

1. Background and Motivation

1.1 Preliminary Problem Identification

Initially, our group wanted to look into how we can promote an inclusive culture when it comes to studying and working on school projects. We noticed that Singapore's education system highly values individual achievement, and less is done to promote the spirit of working together to achieve success. From the personal experiences of our members, we feel that more can be done to promote a collectivist culture in Singapore where we all strive to work together, share responsibilities, and achieve rewards together. To integrate the theme of 'education' into our project, we decided to implement this idea to be based on Singapore's education context. From here, we came up with the goal of our project, which was to create a friendly environment which encourages students to study together and collaborate with one another, thereby promoting the spirit of teamwork and collectivism in students alike.

1.2 Evaluating Disadvantages of Current Applications

With the initial problem identification, we decided to work on a calendar application that can improve collaboration. We first evaluated the shortcomings of popular calendar applications in the market, such as Google Calendar, Outlook, Monday.com etc. We found out that the main problem with these calendars was that only people with the link or people added through email can view and/or edit the calendar. Sharing function is limited as it is unable to reach out to a wide range of people. Also, there may also be Additionally, the notifications for Google Calendar are not "in-app". When a calendar is being shared with you, you have to accept it through gmail and cannot access it within the app itself.

1.3 Further Motivations and Expansion on Target Group

After further discussions, our group decided to further expand on the motivation (section 1.1) to reach a wider target audience. Building on the problem of a lack of collaboration between students, we realised that there is potential to improve on collaboration and schedule management for other users who might face similar problems, but in different contexts. Within the current application market, we identified several shortcomings that lie within these products in promoting collaboration between users.

Firstly, there is no centralised database/list of calendars where users can view and source for themselves. For example, if a user is interested to go hiking around Singapore with a group of enthusiasts, there is no centralised platform where users can know who is going and when these activities are happening. In this case, having a platform where people can publicly view a shared calendar on hiking activities can help enthusiasts to plan and schedule their activities properly.

Secondly, current calendar applications do not support the integration of multiple calendars in a single application. This does not allow users to view concurrent events happening at the same time or share their schedules with one another efficiently. For example, when planning a common schedule to work on a project which involves many collaborators, users have to go through the hassle of checking each other's schedule to reach a properly planned project timeline.

Lastly, there is a lack of social interaction and the 'friendly factor' in calendar scheduling applications. Current calendar applications focus solely on task management or solely on event scheduling. There are no existing social interaction capabilities to allow users to interact with each other. With this lack of social interaction, it is hard to inculcate a spirit of collaboration between users who share the same interest. In addition, we also feel that the lack of socialising capabilities also poses a lack of incentive to use the applications as there is no sense of "fear of missing out".

2. Objective & Solution : The Friendly Planner

With the above mentioned problems in mind, our group therefore decided to work on a social calendar application with the aim of helping users to *improve on collaboration* with each other, and *inculcate a friendly, inclusive environment* by having a *common platform* for them to view and *share their calendars and events*. This is in addition to serving as a personal scheduler just like a standard calendar application.

We derived that Calendars currently available in the market have the potential in spreading key dates and events publicly. However, the lack of social functions limits companies from sharing dates and events publicly across the internet. We see the potential in using this sharing function for friends to share and create interest groups based on their hobbies, By having various personal and public calendars, users can share events and milestones publicly.

Our application will mainly be used as a collaborative tool for better project management and community interaction for users to find common shared calendars and events. With these factors in mind, we hope to encourage a more socially active, organised and balanced lifestyle to our users.

3. Review of Literature/Technology

3.1 Technological Stacks

3.1.1 Figma: Application Design and Wireframing

We utilised Figma for our initial design and prototype. Through Figma, we planned out the initial stages and functions of each button before proceeding to link each page in the prototype and conduct user testing. After gathering user feedback for UX testing, we gained insight from the user's experience about the key buttons and features, and made changes accordingly. We then proceeded to refine the UI and develop the application accordingly using the theme.

While coding the application using flutter, we also made changes to improve and develop the UI accordingly. With reference to our original theme in the clickable prototype, we proceeded to make changes and code the front-end aspect in Flutter using Android Studios.

Besides Figma, we also tried using Adobe XD, due to the screen to code conversion in flutter. Figma was mainly used due to the centralised collaborative space for the entire team through the web link, where we all could edit in real time. Hence, we decided to stick to Figma in the end.

3.1.2 Flutter: Application Development

Flutter is an open-source UI software development kit created by Google. It is used to develop applications for Android, iOS, the web, ect. from a single codebase using Dart programming language. Flutter's ui toolkits that can support various platforms, including Android and iOS.

3.1.3 Firebase: Cloud Storage

Firebase Storage is a storage service built for Google scale and is accessible for free. The Firebase SDKs for Cloud Storage service ensures Google security to file uploads and downloads for Firebase apps. The SDKs can be used to store images, audio, video, or any other type of user-generated content. The server can be used to access the same files.

3.1.4 Android Studio/Visual Studio Code: Code Editing

Android Studio is the official integrated development environment (IDE) for Google's Android operating system, built on JetBrains' IntelliJ IDEA software and designed specifically for Android development.

Visual Studio Code is a free source-code editor made by Microsoft for Windows, Linux and macOS. Features include support for debugging, syntax highlighting, intelligent code completion, snippets, code refactoring, and embedded Git. Users can change the theme, keyboard shortcuts, preferences, and install extensions that add additional functionality.

3.2 Product Management Tools

3.2.1 Wix: Website Design

Wix is a software that provides cloud-based web development services, allowing users to create HTML5 websites and mobile sites through the use of online drag and drop tools. Wix enabled us to have a fuss-free process of creating our application website.

3.2.2 Adobe Premiere Pro: Video Editing

Adobe Premiere Pro is the leading timeline-based video editing software application. Using Premiere Pro allowed us to produce the video for our application easily.

3.3 Collaboration Platforms

3.3.1 Github

GitHub provides hosting for software development and version control using Git. It offers the distributed version control and source code management which are crucial in our collaboration efforts. Using the Dart plugin by Flutter, we utilised a shared GitHub repository and coded the codes using Android studios and Visual Studios. Through GitHub, we were able to collaborate and merge codes made by each other. We were also able to reference and retrieve previous versions of code.

3.3.2 Google Drive

We used Google Drive (docs and slides) as our main goto platform to share important information and resources such as ideation documents, git and flutter resources, meeting minutes, tasks and actions, as well as work on reports and presentation slides in real time.

3.3.3 Zoom

Zoom is a cloud-based video communications app that allows users to set up virtual video and audio conferencing, live chats, screen-sharing, and other collaborative capabilities. Using Zoom, we were able to share our codes with each other through screen-sharing which enabled us to solve problems quicker with many people viewing the screen and helping to resolve the issue at the same time.

3.3.4 MS Teams

Microsoft Teams is a business communication platform developed by Microsoft. Teams offers workspace chat and video conferencing, file storage and application integration. We initially experimented with using MS Teams as our mode of communication due to the file sharing and storage capabilities.

4. Design and Implementation

4.1 Design Consideration / Choice of components

4.1.1 Deciding type of programming language

The first thing our group did was to spend some time researching the different programming language we can use for our mobile application, as well as a flexible database that will suit our needs. We were also on the lookout to find one that supports cross-platform for mobile development, ease of development while maintaining visual consistency across platforms.

After comparing various languages, we decided to go with dart (Flutter), which have multiple advantages, such as supporting same UI and business logic in all platforms and able to reduce code development time with the use of Flutter's "hot reload" feature to show applied changes almost instantly without even losing the current application state. This makes our Flutter app development several times faster due to the increased development speed.

Flutter development framework also functions quicker than its alternatives. The main reason is that our group does not have to write any platform-specific code to achieve the desired visuals in our application as Flutter is able to operate on both Android and iOS devices. We also found out that Flutter allows us to customize anything we see on the screen regardless of how complex it is, making everything easier for the UI team. Additionally, with the rising popularity of flutter SDK, it provides our group with fresh and up to date tutorials on the internet that we can utilise.

4.1.2 Choosing Database integration

After choosing Flutter as the programming language to code, we proceed to choose Firebase as our database due to the real-time data scalability, as well as ensuring data is well synchronized within the application environment. Having a NoSQL database instead of a relational database is also proven to be more effective for mobile application-based workloads.

We have taken some relational databases into considerations such as Moor - a library that allows users to work with Flutter's SQLite database fluently and in pure dart. Using Moor in flutter will allow us to create a database easily without writing code for SQL tables. Moor itself uses SQL as its backend, therefore in flutter, we can directly create a Table using Dart.

4.1.3 Selecting calendar API

We initially used Clean Calendar API for our application, however after progressing further into the project we noticed there were limitations to Clean Calendar. There were limitations in adding and storing events for the clean calendar package as it was hard to track the events that were stored. We also had issues implementing the scheduling of tasks in the clean calendar.

After going back to the drawing board, we researched for another calendar API to be used, the options were Google or Flutter, although Google Calendar API provides us with a lot of inbuilt functions, we ultimately chose Flutter Calendar API as it has high compatibility and ease of use.

4.1.4 Number of features

We started off designing our scheduling application to focus on NTU students. However, throughout the weeks and receiving feedback from professors, we realized that we should not limit ourselves, and make the application more accessible and useful to the public by revamping it to a project management app.

We started brainstorming during the recess week and came up with multiple ideas to make our app more unique and creative. The "shared calendar" function allows users to create or join a calendar that is created by their friends in their friend list and create new calendar events specifically there. We also came up with the "community" function to allow users to interact with each other by adding friends, messaging one another and being able to see all shared calendars created by their friends.

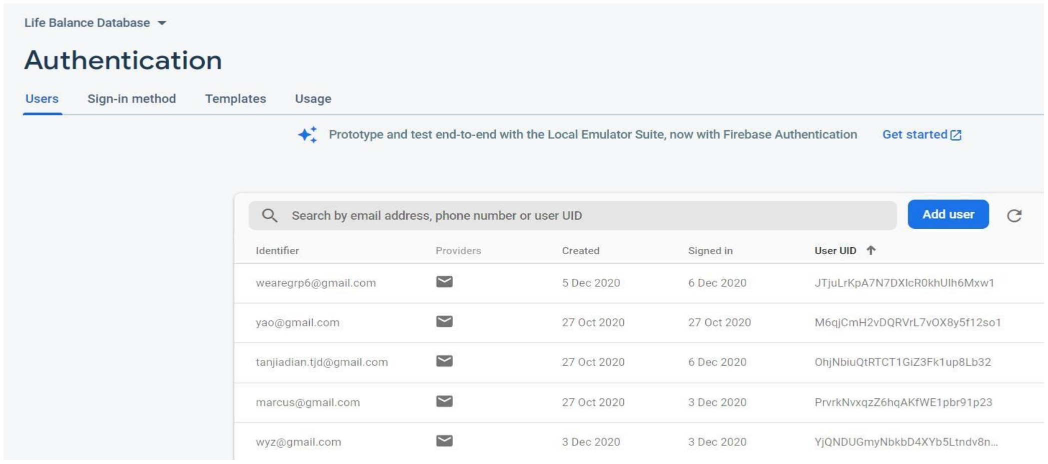
4.2 Final Design (with block diagrams)

See Appendix A

4.3 Implementation

4.3.1 Flutter with Firebase

After setting the basic configuration of Firebase with our project, we proceed to test it out with the login screen we have created. Firebase is able to track the date of which the account is created, when is the latest date the user signed in, as well as giving all users a unique UID for identification purposes (see Figure 4.1)


Life Balance Database

Authentication

Users Sign-in method Templates Usage

Prototype and test end-to-end with the Local Emulator Suite, now with Firebase Authentication Get started

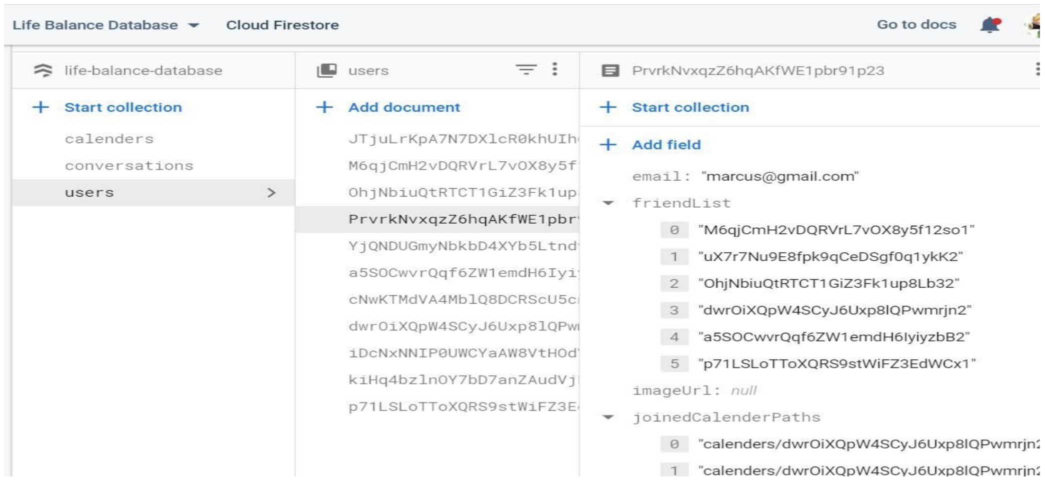
Search by email address, phone number or user UID

Add user

Identifier	Providers	Created	Signed in	User UID
wearegrp6@gmail.com	✉	5 Dec 2020	6 Dec 2020	JTjuLrKpA7N7DXlcR0khUlh6Mxw1
yao@gmail.com	✉	27 Oct 2020	27 Oct 2020	M6qjCmH2vDQRVrL7vOX8y5f12so1
tanjiadian.tjd@gmail.com	✉	27 Oct 2020	6 Dec 2020	OhjNbiuQtRTCT1GiZ3Fk1up8Lb32
marcus@gmail.com	✉	27 Oct 2020	3 Dec 2020	PrvrkNvxqzZ6hqAKfWE1pbr91p23
wyz@gmail.com	✉	3 Dec 2020	3 Dec 2020	YjQNDUGmyNbkbD4XYb5Ltndv8n...

Figure 4.1: *Firebase authentication tab*

The users that signed up will have their details stored under the "users" collection in Cloud Firestore. Each document inside will contain the unique UID of the user, followed by multiple details (fields) such as their email address, friend list and the list of shared calendars they have joined (see Figure 4.2).


Life Balance Database Cloud Firestore Go to docs

life-balance-database

Start collection

calenders
conversations
users

users

Add document

JTjuLrKpA7N7DXlcR0khUIh
M6qjCmH2vDQRVrL7vOX8y5f
OhjNbiuQtRTCT1GiZ3Fk1up
PrvrkNvxqzZ6hqAKfWE1pbr
YjQNDUGmyNbkbD4XYb5Ltnd
a5SOCwvrQqf6ZW1emdH6Iyi
cNwKTMdVA4MblQ8DCRScU5c
dwrOiXQpW4SCyJ6Uxp8lQPw
iDcNxNNIP0UWCYaAW8VtHOd
kiHq4bzlnOY7bD7anZAudVj
p71LSLoTToXQRS9stWiFZ3E

PrvrkNvxqzZ6hqAKfWE1pbr91p23

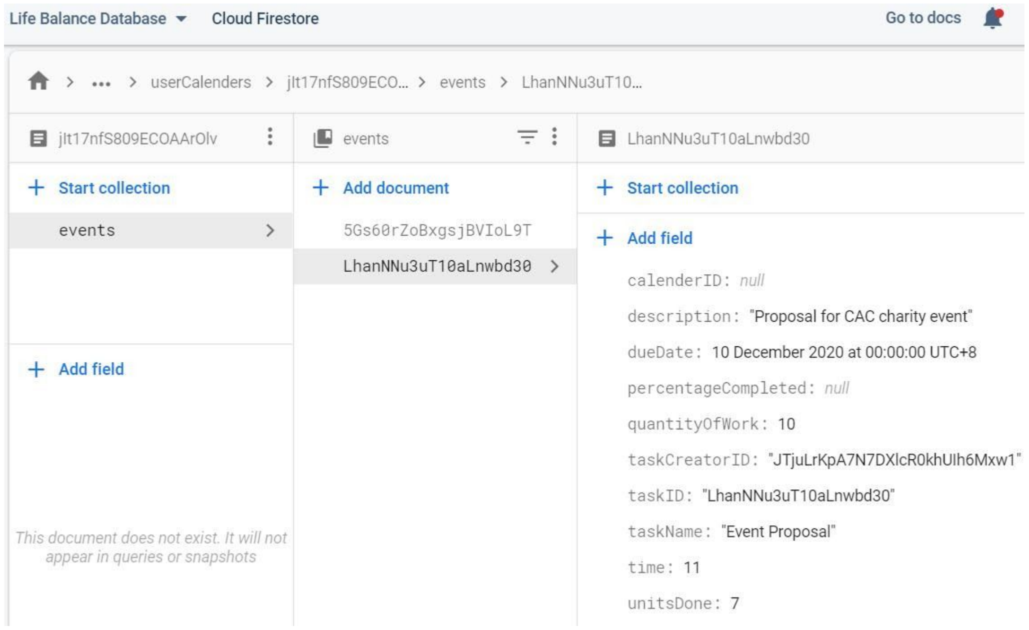
Start collection

Add field

```
email: "marcus@gmail.com"
friendList
    0 "M6qjCmH2vDQRVrL7vOX8y5f12so1"
    1 "uX7r7Nu9E8fpk9qCeDSgf0q1ykK2"
    2 "OhjNbiuQtRTCT1GiZ3Fk1up8Lb32"
    3 "dwrOiXQpW4SCyJ6Uxp8lQPwmrjn2"
    4 "a5SOCwvrQqf6ZW1emdH6lyiyzbB2"
    5 "p71LSLoTToXQRS9stWiFZ3EdWCx1"
imageUrl: null
joinedCalenderPaths
    0 "calenders/dwrOiXQpW4SCyJ6Uxp8lQPwmrjn
    1 "calenders/dwrOiXQpW4SCyJ6Uxp8lQPwmrjn
```

Figure 4.2: *Cloud Firestore Users Collection*

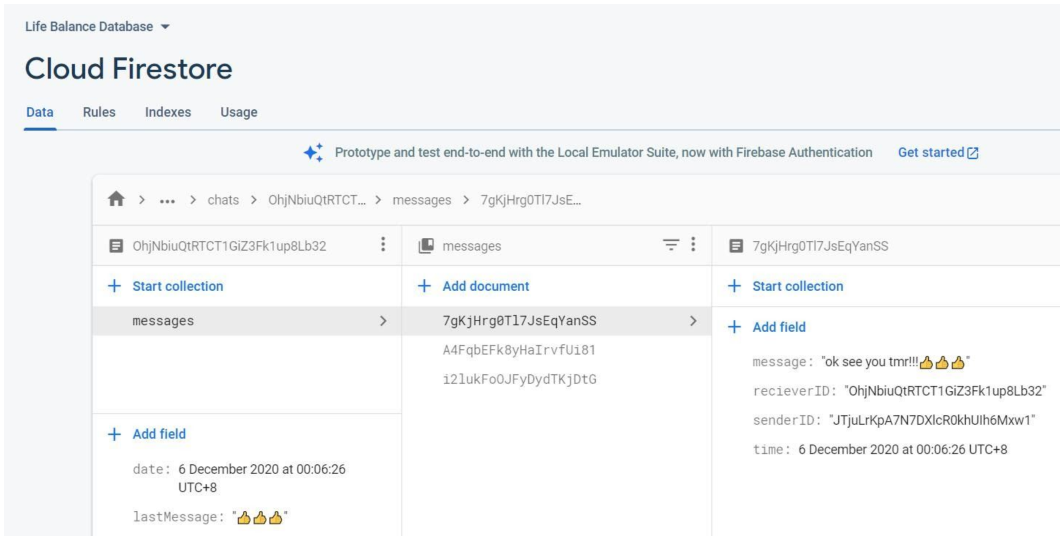
In the "calendars" collection, we are able to see the list of events created by a particular user either in their personal calendar or shared calendar. There are multiple fields stored into the database, such as description of event, due date, task name, event UID, etc. (see Figure 4.3).



```
calenderID: null
description: "Proposal for CAC charity event"
dueDate: 10 December 2020 at 00:00:00 UTC+8
percentageCompleted: null
quantityOfWork: 10
taskCreatorID: "JTjuLrKpA7N7DXlcR0khUlh6Mxw1"
taskID: "LhanNNu3uT10aLnwbd30"
taskName: "Event Proposal"
time: 11
unitsDone: 7
```

Figure 4.3: *Cloud Firestore Calendars Collection*

Lastly, all messages sent or received will be stored in the “Conversations” collection. Fields include Sender UID, Receiver UID, timestamp of message and the contents itself (see Figure 4.4).



```
date: 6 December 2020 at 00:06:26 UTC+8
lastMessage: "👍👍👍"
```

```
message: "ok see you tmr!!!👍👍👍"
recieverID: "OhjNbiuQtRTCT1GiZ3Fk1up8Lb32"
senderID: "JTjuLrKpA7N7DXlcR0khUlh6Mxw1"
time: 6 December 2020 at 00:06:26 UTC+8
```

Figure 4.4: *Cloud Firestore Conversations Collection*

4.3.2 How does personal and shared calendar work?

There are two types of calendar in our app, the user's personal calendar and the shared calendar. When the app starts up by default, the app will show the calendar in private mode (personal). In this mode, all events created will be stored in the user's private calendar with no other users having access to them (see Figure 4.5 and Figure 4.6).

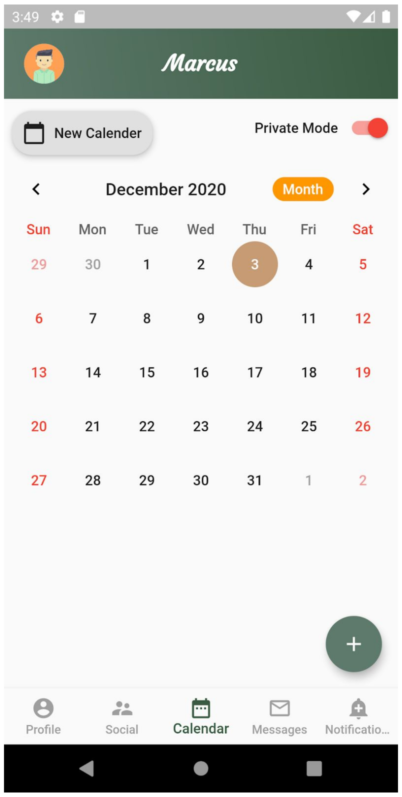


Figure 4.5: *Personal Calendar*

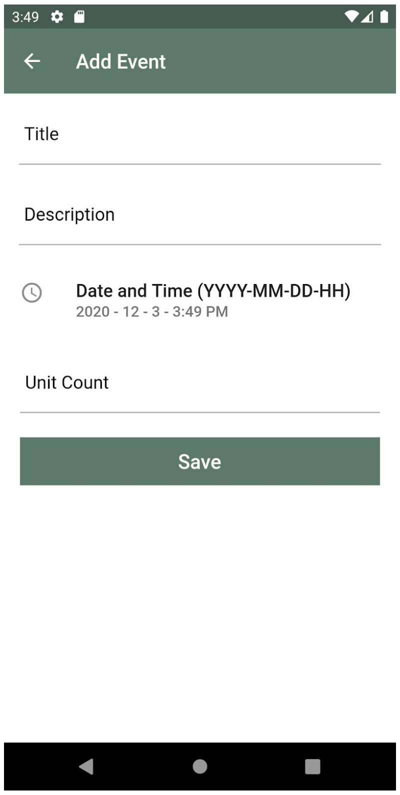


Figure 4.6: *Add Events*

To access the shared calendar function, start off by clicking on "New Calendar" at the top left of the screen and you will be prompted to write the calendar title and a short description (see Figure 4.7). After turning off the private mode, users will be able to see the available shared calendar they have created (not the ones others created. To view all the shared calendars created by others and yourself, head over to the social bottom navigation bar and the list of calendars will be shown under the "Calendars" tab (see Figure 4.8).

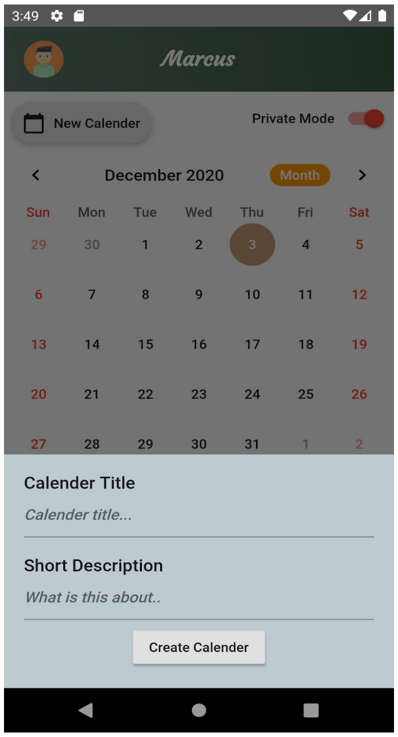
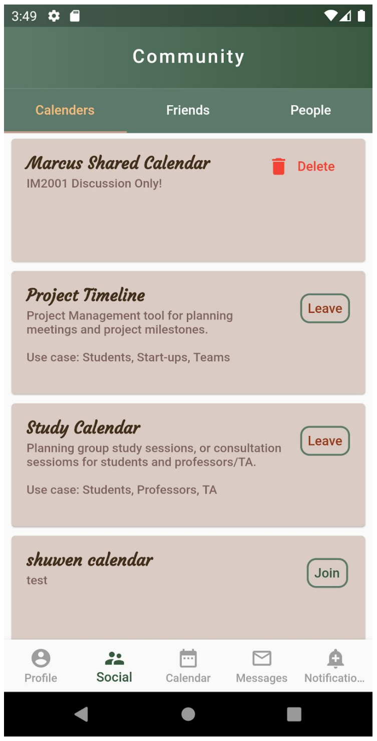


Figure 4.7: *Create Shared Calendar*



Marcus Shared Calendar
IM2001 Discussion Only!

Project Timeline
Project Management tool for planning meetings and project milestones.
Use case: Students, Start-ups, Teams

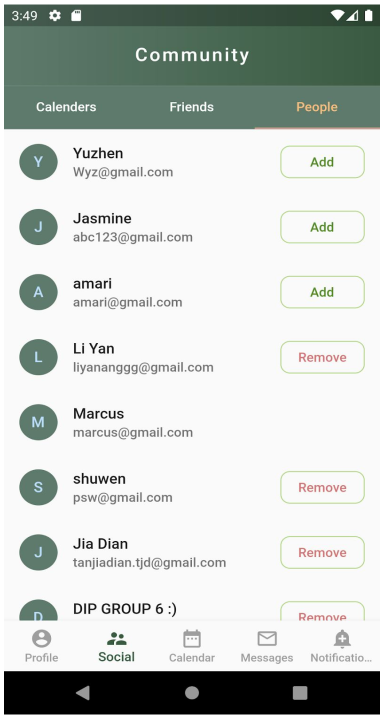
Study Calendar
Planning group study sessions, or consultation sessioms for students and professors/TA.
Use case: Students, Professors, TA

shuwen calendar
test

Figure 4.8: *List of Shared Calendars*

4.3.3 Add Friends and Messaging feature

Users can find the list of friends at the social bottom navigation bar under the “People” tab (see Figure 4.9). Adding and removal of friends will be found here. The friends that the user added will then be added to the “Friends” tab, with the option of starting a conversation with them (see Figure 5.0). Users are able to see the complete list of conversations with their friends over at the Messages bottom navigation bar, including the timestamp of all messages (see Figure 5.1).

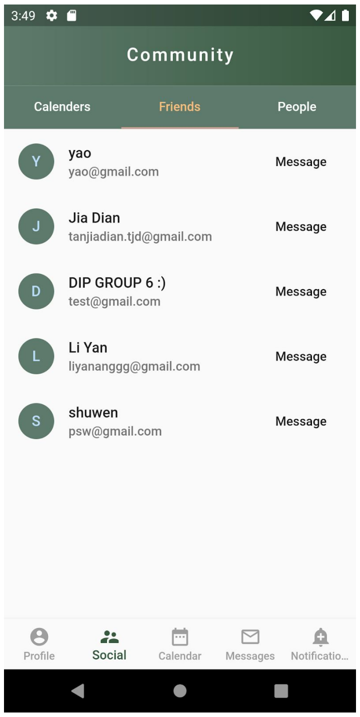

3:49

Community

Calenders | Friends | People

Y	Yuzhen Wyz@gmail.com	Add
J	Jasmine abc123@gmail.com	Add
A	amari amari@gmail.com	Add
L	Li Yan liyananggg@gmail.com	Remove
M	Marcus marcus@gmail.com	
S	shuwen psw@gmail.com	Remove
J	Jia Dian tanjiadian.tjd@gmail.com	Remove
D	DIP GROUP 6 :)	Remove

Figure 4.9: *List of Users*

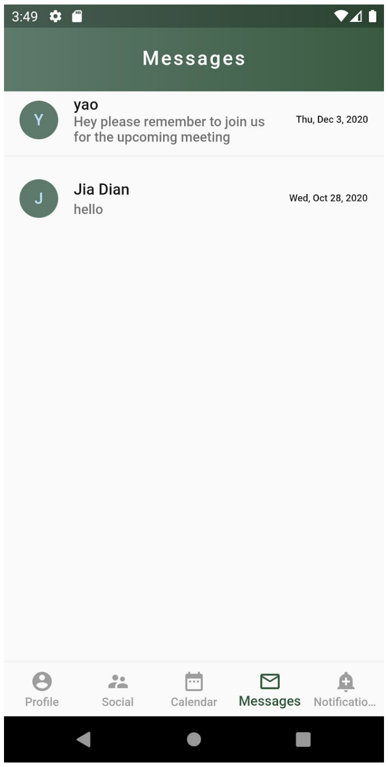

3:49

Community

Calenders | Friends | People

Y	yao yao@gmail.com	Message
J	Jia Dian tanjiadian.tjd@gmail.com	Message
D	DIP GROUP 6 :) test@gmail.com	Message
L	Li Yan liyananggg@gmail.com	Message
S	shuwen psw@gmail.com	Message

Figure 5.0: *List of Friends added*


3:49

Messages

Y	yao Hey please remember to join us for the upcoming meeting	Thu, Dec 3, 2020
J	Jia Dian hello	Wed, Oct 28, 2020

Figure 5.1: *Conversations*

4.4 Discussion

4.4.1 Use Cases and Evaluation

There are several use cases that The Friendly Planner can cater to. In general, the more prominent use cases that our application can be used for are for project management, community collaboration for users to find interest groups and schedules, as well as for clubs/societies/organisations to publicise events on a public calendar that can be viewed by everybody.

Main Use Case 1: Individual and team project management

When working on team projects, users can create a common shared calendar detailing all the common schedules and tasks that the project requires. From there, users working on the same project will then be able to coordinate their time and manage the tasks that need to be completed better. By making use of the social functions within the chat, users can also coordinate their project timeline and schedules better. Other similar use cases will include project management for students, teams, start-ups etc.

Main Use Case 2: Community collaboration

A more powerful use case will be to allow for greater community collaboration where users are able to find calendars that cater to their interest. For example, users who are nature enthusiasts can search for public calendars related to outdoor adventures/hiking that shows the dates and events of nature trips happening in a calendar format. Also, users can view the list of participants that joined the nature adventure calendar which allows users to chat and make friends with other users who share the same interest. Other similar use cases will include students looking for study group related public calendars to find other students to study together with etc.

Main Use Case 3: Event Publicity and Broadcast

Organisations and teams can also make use of the public shared calendar feature to publicise their event dates and schedules. This will allow organisations to collate their event schedules in a concise manner and broadcast these events for other interested users to follow. For example, social media channels such as GoodLobang SG can publicise all promotional events happening in Singapore through a shared calendar that can be viewed by public users. In another example, NTU EEE Club can publicise and update their upcoming events using the shared calendar feature which can be followed and viewed by students using the application.

4.4.2 Process and procedure

Our original idea was education-based, focused on increasing inclusiveness and productivity for NTU students. Creating a unique scheduling application for users to track their timetable and organise study dates will help to encourage study discussions where they can receive help from one another with increased accessibility.

However, throughout the weeks of thought process and implementation, we decided to make our application more “open” to the public and not be limited to just students. Revamping it to a Project Management application will generally make the app more useful and productive for each and every user out there.

With a separate personal and work calendar, we can also help promote a better work-life balance for our users. Having an organised calendar for different uses or projects will definitely be soothing for the eyes and promote the usage of calendars.

4.4.3 Limitations and difficulties faced

As stated earlier, during the first few weeks of our project, our application was restricted to a limited audience, namely NTU students. Although focusing on just one type of audience makes the app easier to design and implement, we realised during the recess week that it was not a good option to do so, and revamping it to all kinds of users out there will definitely make our app more useful.

The biggest difficulty that our entire group faced would be the usage of Dart language, Flutter. This is the first time we are using such language compared to the usual C or Java, and information regarding Dart language is quite limited online. We also could not find the solution from stackoverflow for some problems we faced while coding, which led to us finding alternatives.

There were also issues with the integration with Firebase using Dart as we needed to watch multiple video tutorials to understand how Firebase works so that we can code accordingly.

5. Conclusion and Recommendation

5.1 Conclusion

It is definitely the first time we are doing such a project with a huge group of 10people. We realized that there are many factors that could affect our workflow, such as some members not able to meet up due to other commitments, different ideation, etc., and it all comes down to how well we can manage our time and people.

Different people have different skills, and it is a positive thing for the group, whether they are good at coding or UI, we do polling so that all members choose the role they excel in. It was tough for us at the start as we had to learn a new language (Dart) and improvise the app accordingly, checking online resources such as stack overflow often, finding alternatives as we could not resolve the problem, but all in all, we are glad we manage to finish this project in a good note.

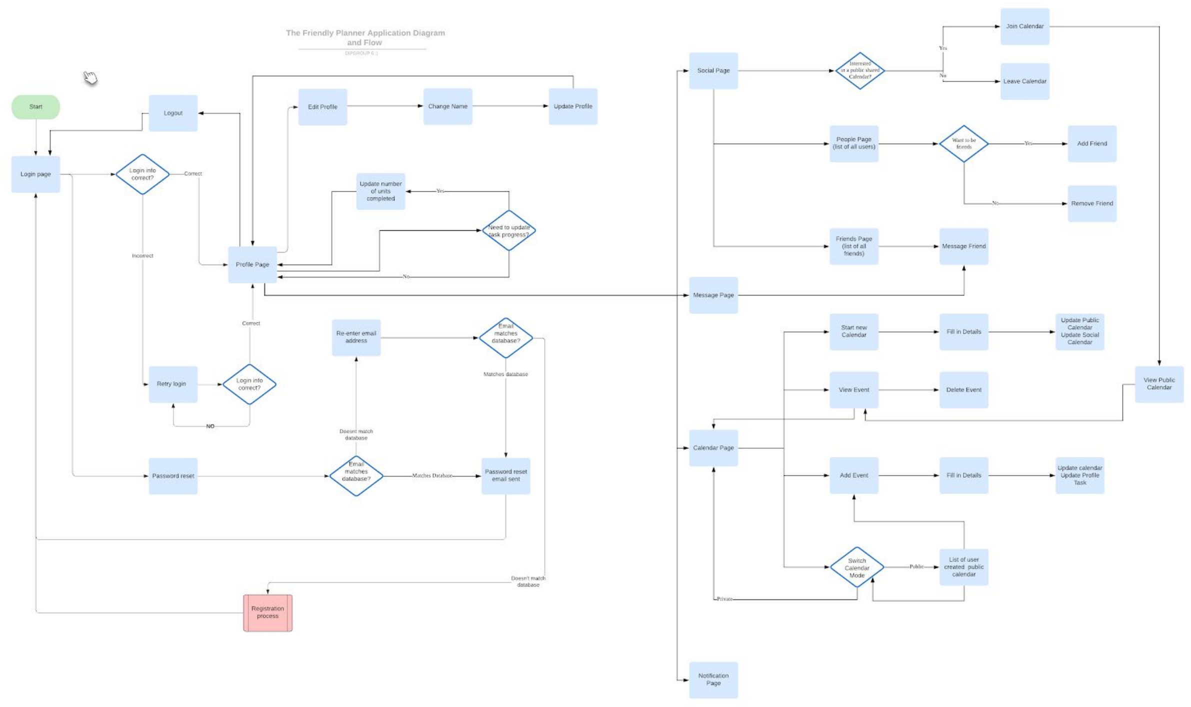
5.2 Recommendation for future works

Taking into consideration the limitations of our application as well as to tailor the application to suit the goal of improving collaboration, our team has identified several areas that we can work on for the future development of our application.

- Make notification page real time - Users can get updates in real time. Examples of updates includes when shared calendar has been updated, when a friend has completed their task etc
- Streamlining the public/private/personal calendars (eg: public is what we currently have, private to be only selected people that can join the calendar, personal is calendars that can be viewed only by the viewer themself) so as to ensure greater control and privacy
- Function to share calendars in group chat
- Add feature to arrange for study date, add stickers for chat to make the application more user friendly
- User authentication and passwords set for privately shared calendars - accessible to selected friends to view and/or edit
- Expand beyond project management - can delve into allowing users to find like minded people with similar interest e.g. users who are interested to go climbing/hiking can search for public calendars within the application and add to their own calendars
- Have a comment function and option to rsvp to event/see participant list
- Collaborate with group 7 (Arrange zoom meetings as part of study date with friends)

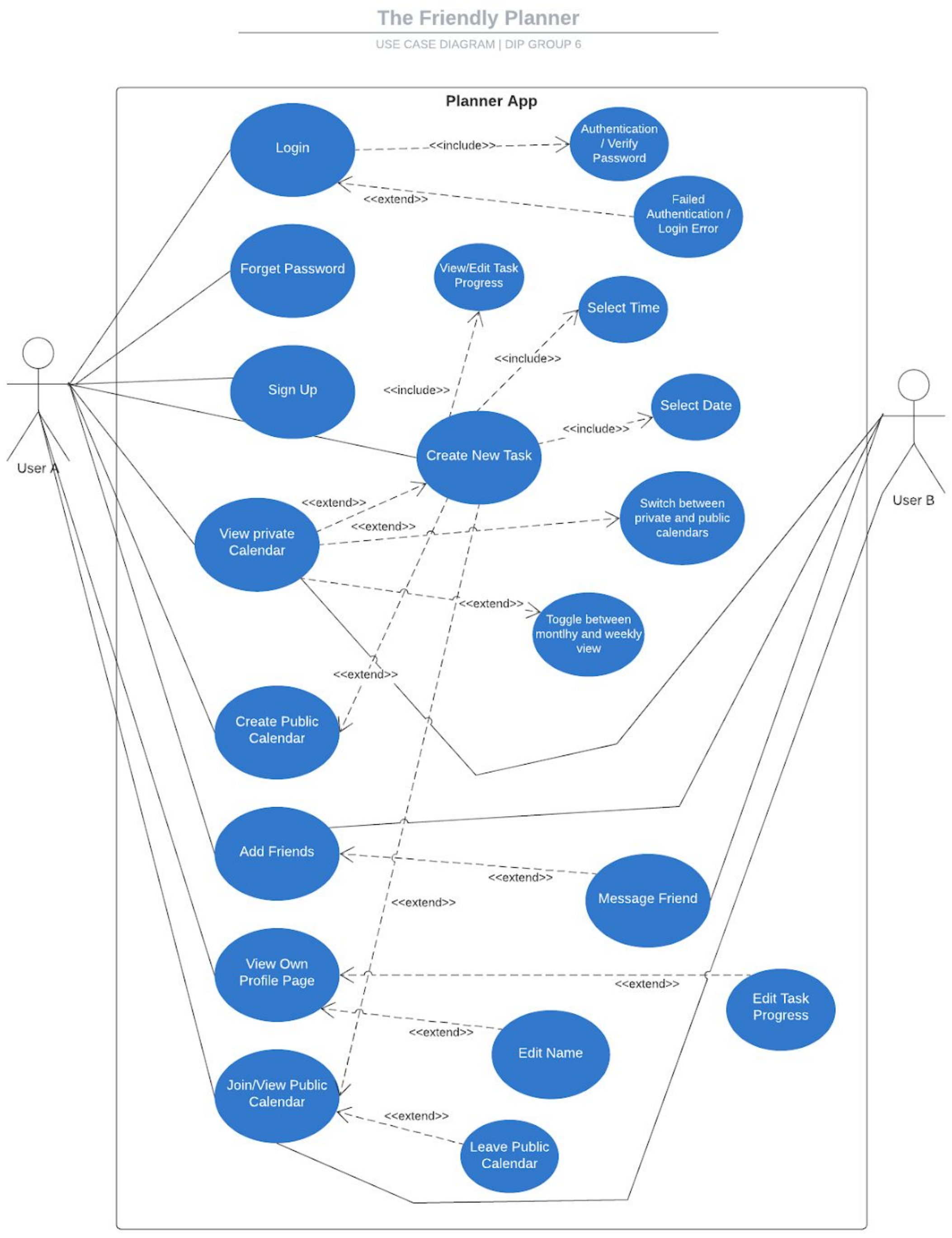
Appendices

Appendix A: Design Diagrams



A1: Flowchart/Block Diagram for Application

https://github.com/ChenZengYao/LifeBalance-flutter/blob/master/documentations/TFP%20Flow%20diagram.pdf

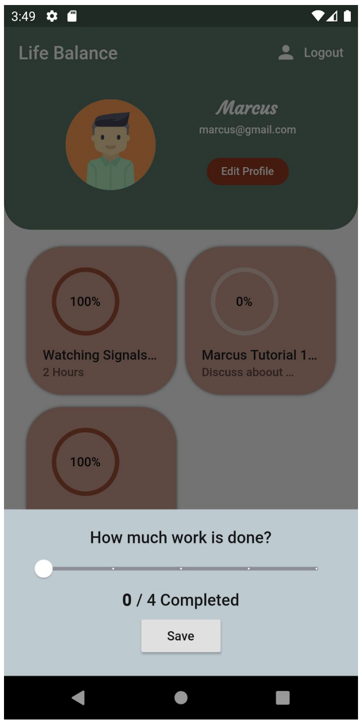


A2: Use Case Diagram

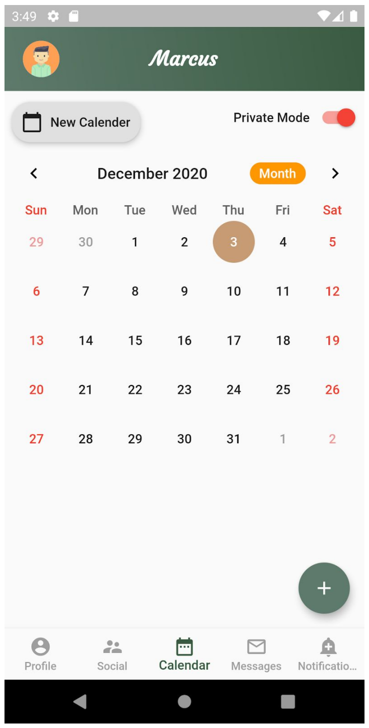
https://github.com/ChenZengYao/LifeBalance-flutter/blob/master/documentations/Use%20Case%20Diagram%20For%20The%20Friendly%20Planner.png

Appendix B: User Guide

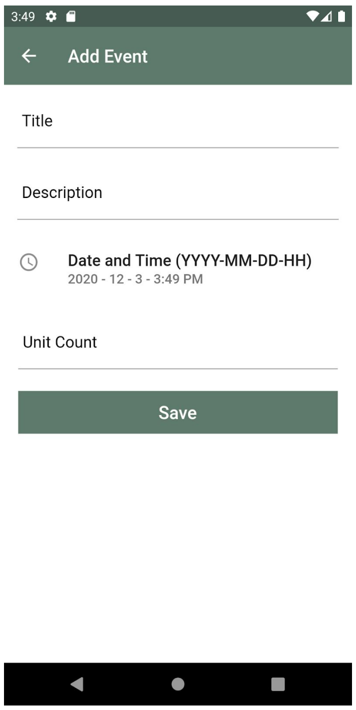
a. Screenshot and Functions of each screen



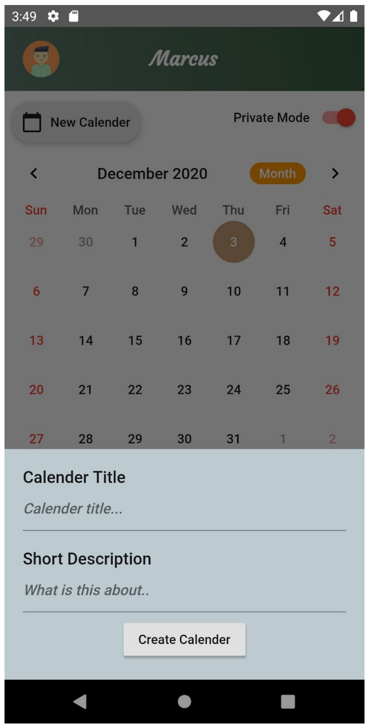
Slider Bar to show work done



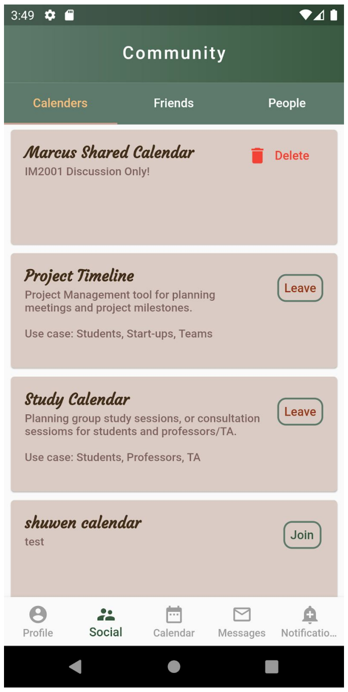
Personal calendar to add events



Add events here



Create a shared calendar



3:49

Community

Calenders | Friends | People

Marcus Shared Calendar Delete
IM2001 Discussion Only!

Project Timeline Leave
Project Management tool for planning meetings and project milestones.

Use case: Students, Start-ups, Teams

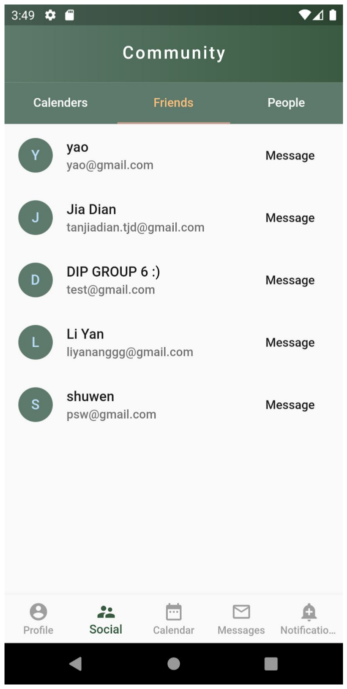
Study Calendar Leave
Planning group study sessions, or consultation sessioms for students and professors/TA.

Use case: Students, Professors, TA

shuwen calendar Join
test

Profile | Social | Calendar | Messages | Notificatio...

List of all shared calendars



3:49

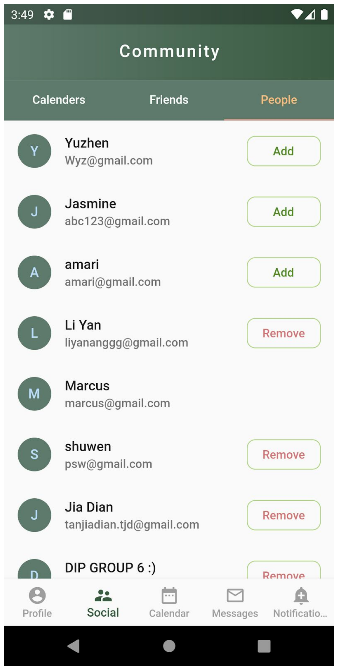
Community

Calenders | Friends | People

Y	yao yao@gmail.com	Message
J	Jia Dian tanjiadian.tjd@gmail.com	Message
D	DIP GROUP 6 :) test@gmail.com	Message
L	Li Yan liyananggg@gmail.com	Message
S	shuwen psw@gmail.com	Message

Profile | Social | Calendar | Messages | Notificatio...

List of friends added



3:49

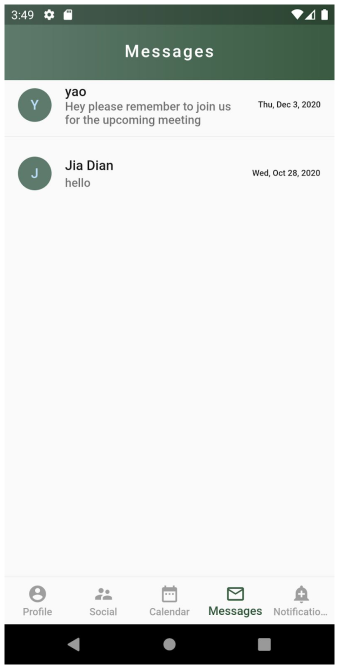
Community

Calenders | Friends | People

Y	Yuzhen Wyz@gmail.com	Add
J	Jasmine abc123@gmail.com	Add
A	amari amari@gmail.com	Add
L	Li Yan liyananggg@gmail.com	Remove
M	Marcus marcus@gmail.com	
S	shuwen psw@gmail.com	Remove
J	Jia Dian tanjiadian.tjd@gmail.com	Remove
D	DIP GROUP 6 :)	Remove

Profile | Social | Calendar | Messages | Notificatio...

List of available people



3:49

Messages

Y	yao Hey please remember to join us for the upcoming meeting	Thu, Dec 3, 2020
J	Jia Dian hello	Wed, Oct 28, 2020

Profile | Social | Calendar | Messages | Notificatio...

Conversations

Appendix C: Maintenance Guide

a. Security updates and code maintenance
    i. Patches (dart version)
    ii. Firebase compatibility versions (need backend team help)
    iii. Offer Scheduled System Maintenance
    Scheduling regular system maintenance to avoid missing out on issues or bugs. It also ensures your app is always up-to-date and works smoothly over time. Also, informing users beforehand about the scheduled maintenance sets expectations between you both. This is important because you do not want to unexpectedly hit your users with a “System down for maintenance” message, which can be quite frustrating.

    iv. Fix Bugs Timely
    Users will be driven away, if issues in your app are not fixed regularly. Accumulated bugs can also lead to crashing of apps and failing for good. They can also multiply your expenses if left alone for a long time. Always keep an eye out for bugs, technical issues and fix them as soon as possible. Keep track of user feedback and reviews to efficiently address any issues or complaints.

    v. Android version
    It is crucial to make your application compatible to all android versions(old and new). Hence, users will be able to download and use it even if their phones have the older version of android.

    vi. Add New Feature Updates
    Adding new feature updates to your app based on user feedback and usage patterns helps retain them and keep them interested. You should also keep a check on your app reviews. Most users tend to leave suggestions for improvements and new features. Use those ideas to add better updates to your app. Adding regular, minor feature updates is also a more cost-effective solution compared to building an entirely new one or updating after a long time.

Appendix D: How-To Guides

a. https://docs.google.com/presentation/d/1HTTLDrV-SMnvX7Q5kmDNsidMg6T76tu4l_2OGSeaauw/edit#slide=id.p1

Appendix E: Source Code

b. Link to github repository : https://github.com/ChenZengYao/LifeBalance-flutter

Appendix F: Others

a. Our website: https://isorandom95.wixsite.com/dipgrp6

b. Our Poster

The Friendly Planner
New planner for your every needs
All in one solution that is engineered specifically to suit the needs of all individuals that like to have a clean and neat powerful calendar!
Features
Interactive Slider Bar
Adjust accordingly to track your calendar event progress!
Notifications
Get notified to keep up with what your friends are doing to their calendar!
Social Friends
Feel free to text your friends who are in the friend list!
Social People
Users that can be added to your friend list!
Personal & Shared Calendar
Create events in personal or shared calendar!
GET IT ON Google Play
Download on the App Store
Download & Plan Now!
Check out more at our website!
https://isorandom95.wixsite.com/dipgrp6
The Friendly Planner
Logout
Grp 6
Edit Profile
32%
DIP Final ...
70%
Event Proposal
Proposal for CAC ...
Profile
Social
Calendar
Messages
Notifications
Community
Marcus
December 2020

c. Figma Planning:
https://www.figma.com/file/OZeh1JpiZXfskH2jvwC9sY/Planner?node-id=22%3A35