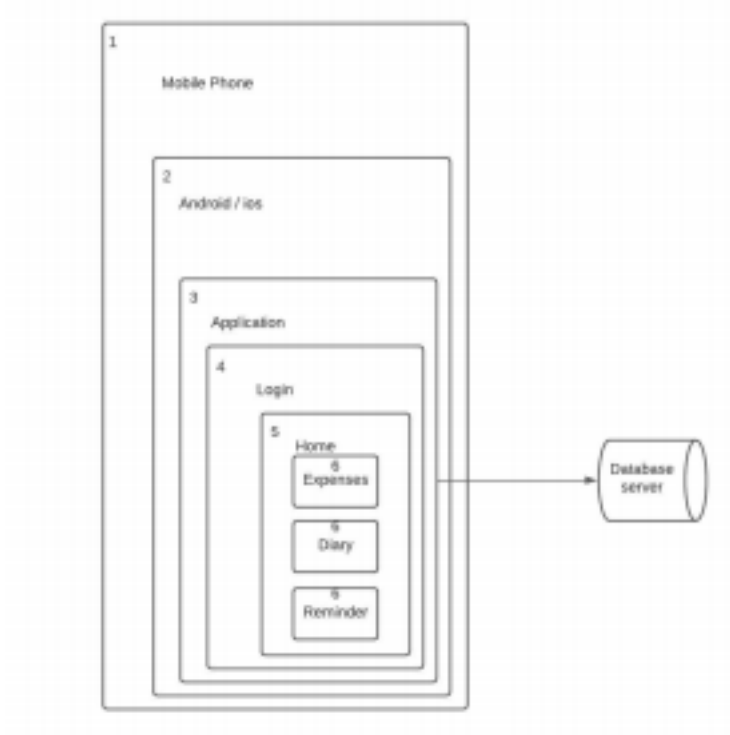
B19CSE045
M K Laksath Adityan
CSL2060 Software Engineering
Software Design and Architecture document with focus on the chosen Quality
Topic : Group Assignment from Software Engineering Course (B19CSE045 & B19CSE039)

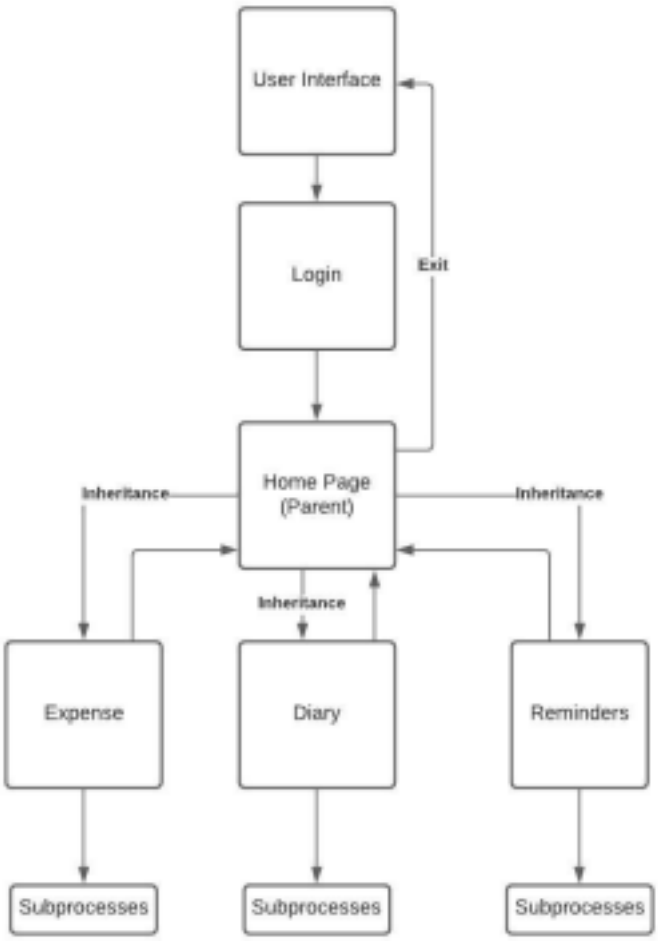
1.Physical / Deployment View :

This view explains about the system's execution environment.



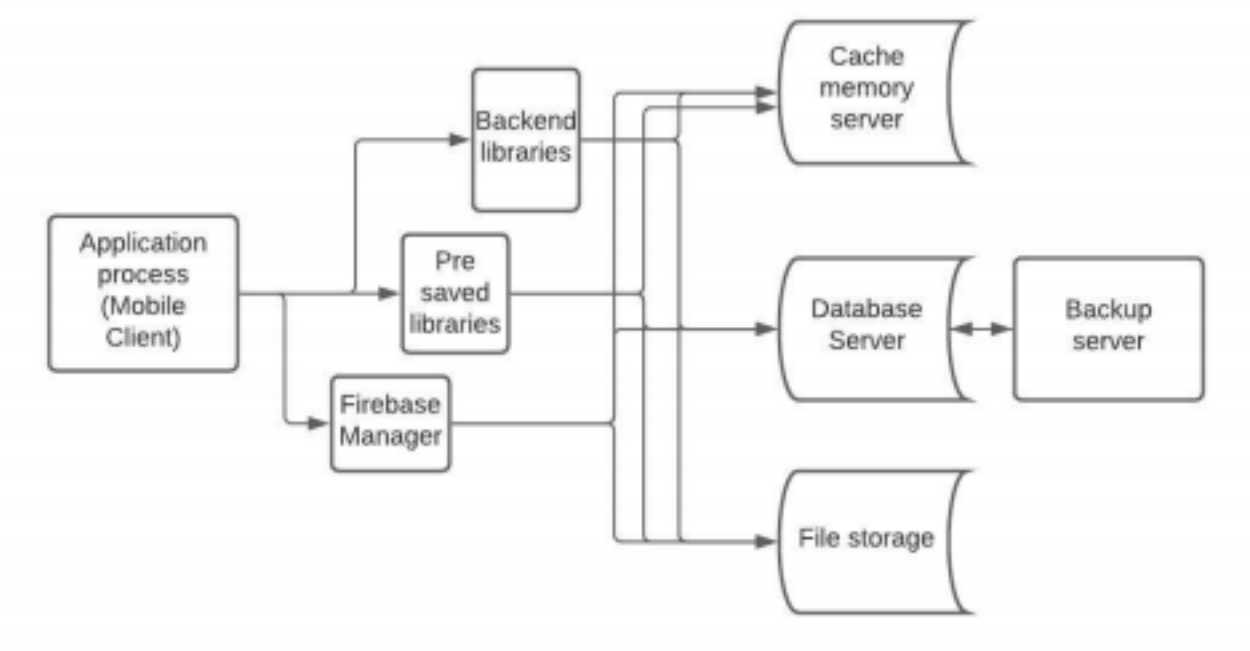
- This shows how the data is physically stored in a database and the process behind it.
- This also explains about the deployment of the application.
- End users and software engineers would be the major stakeholders in this case. ● The UML component diagram here represents the hierarchical order in which the software application is located and also shows the major components involved in the application.
- Changes to the application will be saved in the database server.

2.Logical Vlew :



- This view explains the hierarchical processes of the application through class and object Inheritances.User must Login as it is the most important part to use the application and the home page is the parent of all subclasses.
- Software designers, end users,programmers and software engineers would be the major stakeholders in this phase.
- Here, the subclasses are not related to each other.
- Home page : Parent Class.
- Expense :Child Class.
- Diary : Child Class.
- Reminders : Child Class.
- Home page must remain as a Stateful widget while the expense , diary and reminder classes may not have a state and hence can remain a stateless widget.

3.Process View :



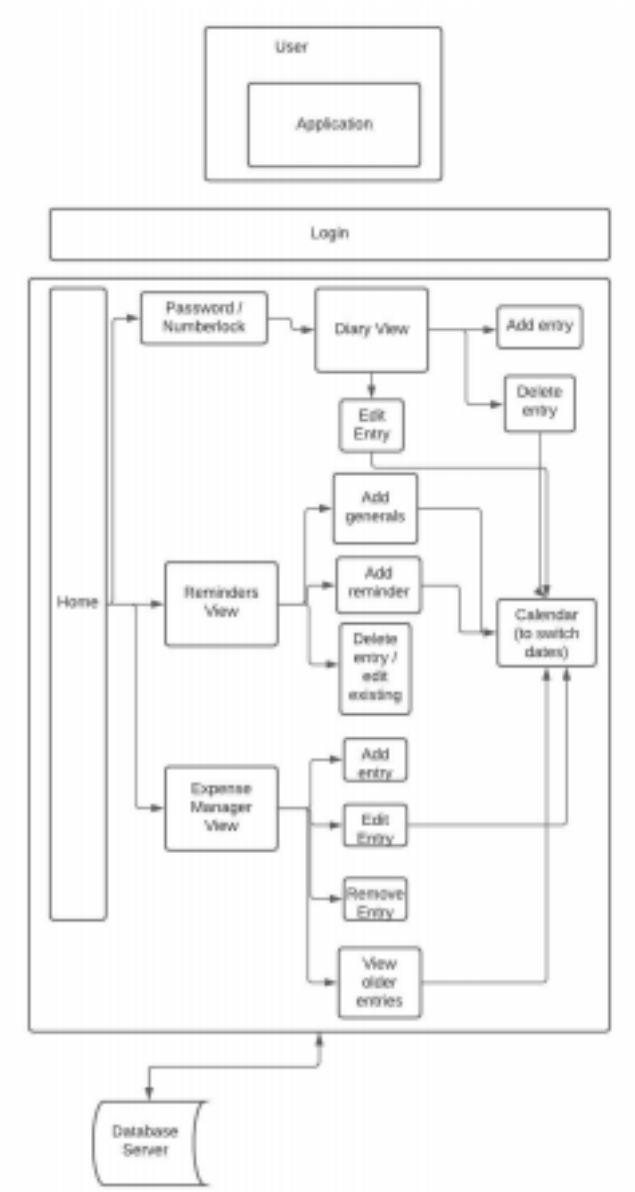
● The process View explains about how the Software communicates, how it runs in the database management servers and how these groups of tasks are executed. ● Backend programmers would be a major stakeholder in this category. ● When the client uses the software and makes several changes, these are stored in the backend libraries and are added to the part of pre-saved libraries.

● Firebase Manager also helps in the processing of the new data.

● These are then added to the cache Memory server and are mainly stored under the database server.

● The backup server is used when the main database server is down and hence is only connected to the database server.

4.Developement View :



● The development View explains the software from the programmer's Viewpoint.
● Hence, the software engineers, programmers and project managers are major stakeholders in this particular category.
● It shows the subsystems, libraries used, and their hierarchical order in which they perform.
● It is important to notice from the diagram that at least one of the subfeatures from every subclass requires the Calendar Feature to edit their data.
● Hence, the calendar is one of the most important features in the software. ● All the changes / editing made in the software by the user is then stored in a database. ● It is also important to note that none of the subfeatures' features are related to each other and Hence no representation of it is displayed.

5. Use-cases/ User Scenarios:

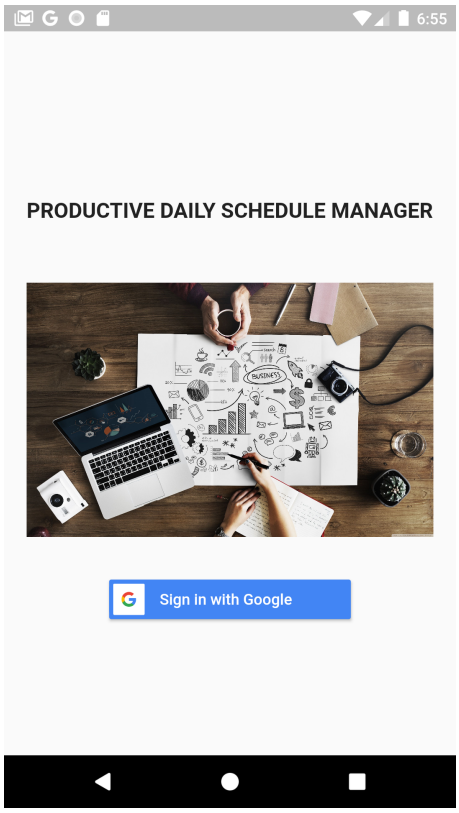
There are three main features in this application.They are : Reminders, Diary and Expense Manager.

- Setting up reminders/deadlines: Users have an option to set up a reminder for their custom made deadlines.
- Authentication: Username and password is required to keep log of all the activities (Gmail Authentication) in order to backup the data.
- Expenses : Expenses can be managed through entering the amount spent with the title entered on what the money was spent on.In this way, the application keeps track about the dates at which certain money was spent and provides a weekly graph that displays money spent via comparison.
- The Diary feature consists of an extra password which is provided with a local authentication.The user may add entries in that and may edit it in the future. ● Stakeholders Involved : Development Team, Designer Team, Testing Team , user end,Maintenance Team and Database Team.

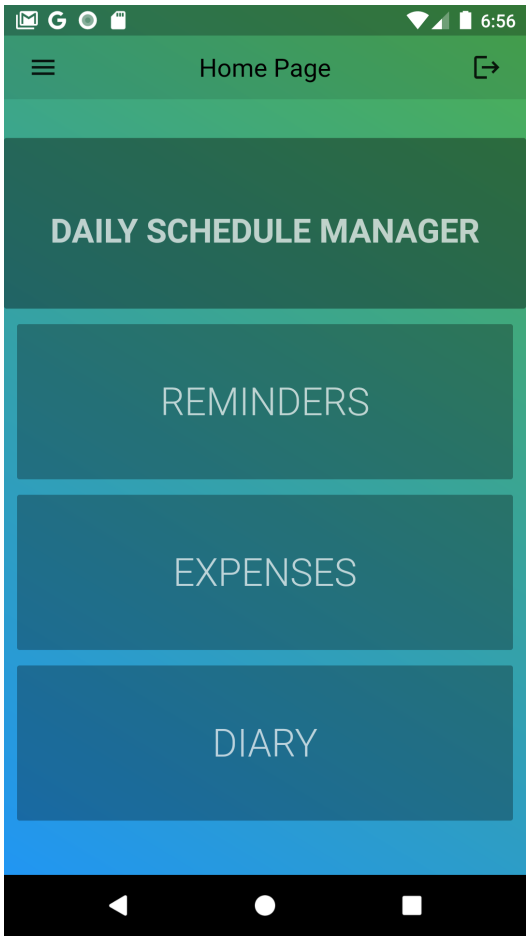
Quality Attribute Chosen :

Maintainability

1 (Login Page) :



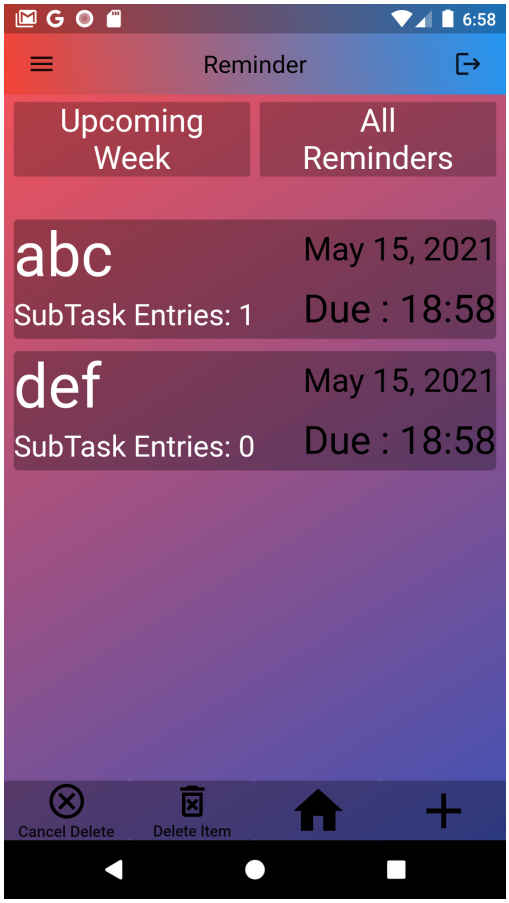
- This is used for signing in with google to keep track of the user's data.
- By the Justification based on User Requirements, This is an important feature since the user could use the same app without losing data in other devices.
- The user could also use a guest login just in case the user doesn't want to sign in with their google account.
- Sign in with the google button is imported from Firebase's facility to add userName via the gmail address.
- Therefore, this is easily maintainable as it consists of a 3rd party library that is easy to be handled.



2 (Home Page) :

- This is the homepage of the software application.This is used for navigating between the main features of the app (Reminders , Expenses and Diary).
- Accounts can be switched by tapping on the top right corner.
- The icons are made big enough and buttons are made comfortable meeting the needs of user requirements.
- This is the homepage that connects the application with All the subComponents such as Reminders, Expenses and Diary.
- It is made as a separate class to separate it from the rest of the code.
- Any changes / improvement on this page alone, can be easily made.
- Hence, this is easily maintainable.

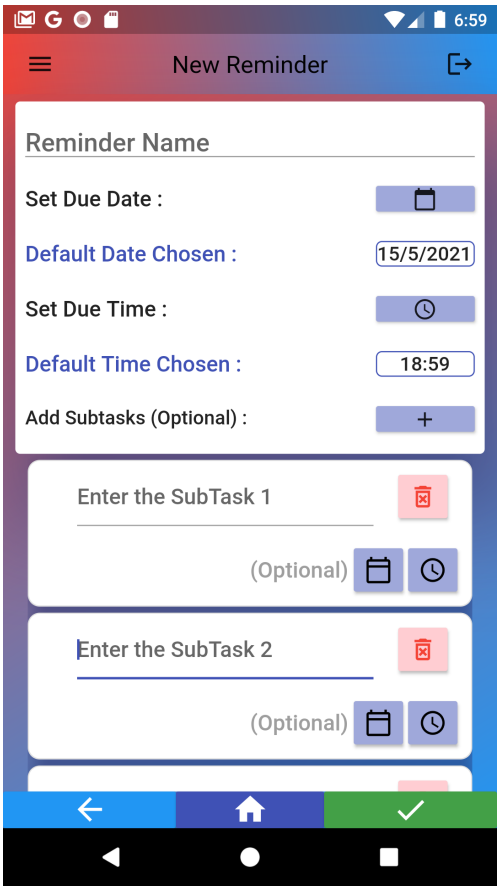
3 (Reminders) :



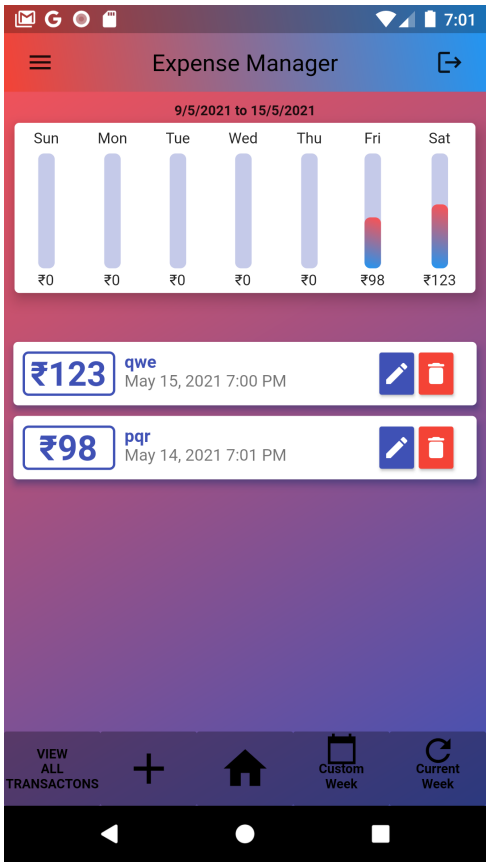
- This is the reminders of the application and
- The first column shows the reminders that are due this week and 2nd column shows that are due All time.
- The following set of rows shows the total number of SubTask entry reminders set which includes those without due dates.
- The reminders section is scrollable (when it exceeds the height limit).
- New reminders other than the general reminders can be added by clicking on the '+' icon (calendar with '+' icon).
- The user is required to select any of those displayed entries to get detailed information about it.Details such as time remaining (deadline) , a brief description about the title (if any) will be displayed along with the subTask entry names and deadlines.

- The delete reminder allows the user to delete the reminders and one can cancel a delete by tapping on the cancel delete button.
- It is also important to note that here, the list of rows that appear are a separate class in the code, the bottom bar is also a seperate class , and the appBar too is a separate class and this makes the application easily maintainable since the changes that we might make in the near future to a certain part wouldn't affect the other as every part is isolated from the other.

4 (Adding a new Reminder):

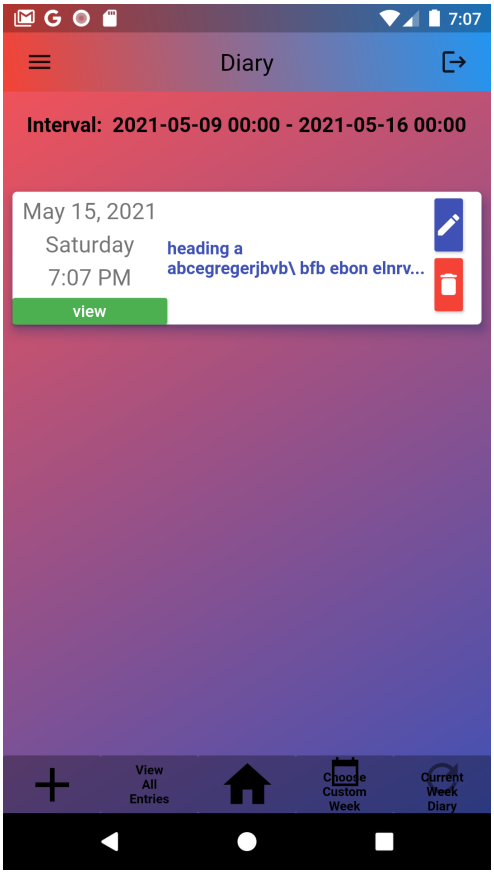


- Upon choosing the ‘add new reminder icon’ , the User is expected to enter a :
    - Heading : The title through which a particular reminder is identified when searched through the search bar.
    - Details : A brief information regarding the reminder can be given although one may choose not to provide any.
    - Date & Time : This is the deadline feature and the user will be notified before the deadline to finish his/her task.This feature is also made optional for the user.
    - Optional SubTasks with optional date and time for it.
- Upon pressing the Add button, a new subTask is added into the all reminders section.
- Upon pressing the delete button it is deleted.
- We can cancel an entry by pressing the left arrow.
- These new reminder entries are easily maintainable since they are isolated from the other and subTask entries are provided with a separate link that is uniquely linked to its appropriate reminder.
- Changing the design for this in the near future should not be a difficult task as they are just basic renderings.



5 (Expense Manager):

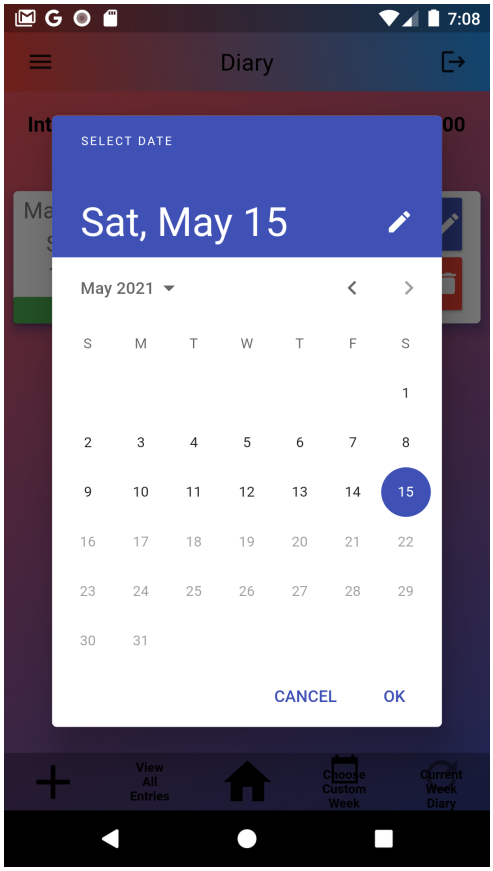
- The expense manager displays a title (on which the money is the money spent) and below that , a Date and time on which the particular entry was entered.
- The bar graph above shows the ratio of money spent on each day to the total money spent on that weekend.
- One may click on the trash icon to delete it if the transaction was done by mistake.
- The Home screen button allows the user to go back to the home screen.
- To add a new expense report, the user may click on the '+' button.
- This would be useful to the user as it avoids complicated features and provides simple functionality.
- The calendar button opens the calendar on which a day can be chosen.
- Upon choosing the date , All the expenses of that particular week will be displayed.
- On choosing the view all transactions button, the expense report is displayed at an all time.
- The bar graph seen is made into a class called chart where each bar graph is made from another class chartBar.Hence, changing the colour of the bar graph or the layout's position change in the near future should not be a difficult task in the future updates.In this way, this page is easily maintainable.



6 (Diary):

- The '+' button can be used to add new entries.
- Every entry displays the Month,day,date and time of entry being made at the left end of the entry.
- The beginning details of the diary is displayed so that the user might remember what was written that day by just looking at the beginning content.
- Pressing on any view button opens the complete content of that particular day.
- The list is scrollable and provides all the entries of that particular month.
- The calendar button is provided at the right end to switch between different months and dates.
- The current week button brings the user back to the current month and year.
- An existing entry can be edited using the edit button and deleted using the delete button.
- THe list of diary entries are scrollable when they exceed the size limit of the mobile application and It is made from a separate class.In that class there are rows and columns for the rendering purpose of the widgets present in it.
- This makes the list unique in nature and hence, this page is easily maintainable

7 (Calendar):



- The calendar wireframe is used in all the three features of the application.
- It allows the user to pick a date.
- Clicking on the Month,Year , the user may choose the year and month from there.
- The user may also slide between months.
- Pressing on the > button leads to next month and pressing on the < button leads to the previous month.
- The cancel button is used to exit the calendar without choosing anything from the calendar.
- The ok button is used for confirming the date , month and year from the calendar.
- This is a feature that is provided by flutter in default and this can be changed by overriding the codes if necessary.
- Hence, the calendar part is easily maintainable.