COEN 285-01 SE

Group Name: DiaryCraft

Final_Deliverables

Haoyu Yuan, Jinhao Wang, Pengwei Lin, Yukun Zhang

Introduction

Our project goal is to build a fully-functional Android application with a neat interface to record daily lives via texts, images, and tags.

People used to write articles, diaries, and notes with pens and notebooks. But the thing is that they need to bring all the stuff to one place if they are desired to write down any texts. Notebooks are too big and soft for some wild or crowded scenes. Also, to some extent, the speed of writing is slow. Therefore, the disadvantages of traditional writing way are time-consuming and inconvenient. However, things are much different if we try a new approach: create the content in a smartphone. After decades of technology development, our mobile phones are getting thinner and smarter, most of the tasks that may be handled in the PC before can be done by our mobile devices now. Based on the advantages of mobile devices, it is more convenient, efficient and relaxed to write any notes on a phone with ease.

Our lives are full of all kinds of experiences, some are amusing, others are significant to us in a special way. Other things like sudden thoughts also greatly enrich our lives. On the other hand, however, our brain is too weak to remember all of the things. These experiences and sudden thoughts are all important and worth recalling. Meanwhile, for the future, we need to build specific schedules in order to achieve our goals and make everything in order.

The application we designed and implemented encourages people to record fascinating and meaningful experiences or setup future schedules. It makes writing diaries no longer a shallow work but can record users' real passions.

Requirements Analysis

By the time of the final demo, our team set up 16 user stories with our customer. Among them, 11 of them were implemented during the first 2 sprints, 4 of them (No. 6, 12, 13, 14 and 15)are being implemented during the 3rd sprint (which is scheduled from Nov 25th to Dec 8th).

1. As a diary user, I want to create a new diary so that I can start to record a new experience;
2. As a diary user, I want to check my previous diaries so that I can recall my sweet memory;
3. As a diary user, I want to update a previous diary so that I can add the memory I forgot before;
4. As a diary user, I want to attach photos to diaries so that I can recall memory through vivid pictures;
5. As a diary user, I want to delete a previous diary so that I can throw away the memory that now hurts me;
6. As a diary user, I want to share my diaries so that my friends can enjoy my experience with me;
7. As a diary user, I want to search for a specific diary so that I can find some useful information I need;
8. As a diary user, I want to sort diaries in a specific order so that I can quickly find the diary I am looking for;
9. As a diary user, I want to add the location where I wrote the diary so that I can remember the place that day;
10. As a diary user, I want to classify my diary and be able to see all categories as tags so that I can be reminded that what photos I inserted and got the sweet memory from them;
11. As a diary user, I want to have to page to see all the photos I have ever added into diaries so that I can easily search the diary and see the category a diary belongs to;
12. As a diary user, I want to watch the app in a more comfortable way so that I can protect my eyes while watching display;
13. As a diary user, I want to log in as a unique user so that I can have a cloud backup of my diaries and share my diaries on social platforms;
14. As a diary user, I want to log out the current account so that I can switch between different accounts or continue as un-login status;
15. As a diary user, I want to change the application's theme color under light mode so that I can use the application more comfortable;
16. As a diary user, I want to add the date to every diary I write so that I can keep my diaries in order and easily find the one I look for.

Below is the 4 use cases raised by each team member in Assign1.

1. Use case developed by team member Haoyu Yuan:

Use-case: Delete one of the existing records
Actors: User
Type: Primary, Essential

Description: User wants to delete one of the existing records in the app.
Preconditions: There are existing data currently.
Scenario Details (Typical course of events)

Actor Action	System Response
1. User swipe left on one of the records in the list.	2. “Delete” button is showed.
3. User presses the “delete” button.	4. A dialog saying ”are you sure?” is shown.
5. User press yes.	6. System deletes the record.
	7. System updates the showing list.

Extensions:
2.5 If the user presses other areas, the delete button will disappear.
4.5 If the user presses no, the dialog will disappear.

This use case is completely implemented.

2. Use case developed by team member Jinhao Wang:

Use-Case: Create New Note
Actors: Application User
Purpose and Description: User wants to create a new note in the application
Type: Primary, Essential
Precondition: None
Scenario Details:

Actor Action	System Response
1. User presses the “Add” button shown in the application	2. Prompts a blank page (Add Activity) for user to manipulate.
3a.User finishes the manipulation and presses the “Save” button.	4a1.Save the new note to local storage. Check whether the user has an account logged in and whether the user turns on cloud saving. If yes, upload the new note to cloud storage.
	5a1.Go back to the main activity and show the note’s abstraction on the list, sorted by the user’s preference.
	5a2.Fail to upload the new note to cloud storage, save to local storage and go back to the main activity, display an alert message and show the note’s abstraction on the list, sorted by the user’s preference.
	4a2.The number of diaries reaches the maximum amount.

	5a3.Go back to the main activity and display an error message.
3b.User changes his/her mind and does not want to save the note. He presses on the "Discard" button.	4b.Go back to the previous activity.

This use case is not completely implemented. Our team has not fully implemented the login function. Therefore, system response 4a1, 5a2 are impossible to be executed.

3. Use case developed by team member Pengwei Lin:

Use-Case:(Goal): Edit the diary content
Actors: Customer
Type: Primary, Essential
Description: Customer wants to edit one of the diary content.
Preconditions: Selected Diary exists in the system.

Actor Action:
1. Customer taps the diary card view on the main activity page.
3. Tap the EDIT button on the top right corner.
5. Tap content textfield and type new content (delete or add new text)
7. Tap the SAVE button on the top right corner.

System Response:
2. Present the diary detail page.
4. EDIT button changes to SAVE button (Hide EDIT, show EAVE), diary content is able to edit with tapping on textfield.
6. Waiting for the customer to save content. (Nothing responses.)
8. SAVE button changes to the EDIT button (Hide SAVE, show EDIT), new content has been saved. Database update new content data.

This use case is completely implemented.

4. Use case developed by team member Yukun Zhang:

Use-Case: Checklist
Actors: Young man
Type: Primary, Essential
Description: Young man wants to check his previous diaries
Preconditions: none
Scenario Details

Actor Action	System Response
1. User opens the app	2. Shows the list of every diary

This use case is completely developed.

Estimation

Part1. Agile Estimation Method

Product Backlog:

Backlog Item	Estimate
Create a new page	1
Checklist	1
Attach photos	5
Share diaries	34
Sort list	3
Add tags for diaries	2
Dark mode	8
Log out	21
Add date	5
Update diary	2
Delete diaries	1
Search keywords	5
Location	8
Waterfall photo wall	13
Log in	21
Change theme color	2

Backlog for Sprint 1:

1. Create a new page
2. Checklist
3. Update diary
4. Add tags for diaries
5. Delete diaries

Sprint 1 started from Nov 4th to Nov 17th.

Team Velocity for Sprint 1 is 7.

Backlog for Sprint 2:

1. Attach photos
2. Search keywords
3. Location
4. Waterfall photo wall
5. Add date
6. Sort List

Sprint 2 started from Nov 18th to Dec 1st.

Team Velocity for Sprint 2 is 39.

The average team velocity is 37 (Plus the planned Sprint 3 team velocity).

Effort in Person-Months to complete all user stories: 1.667.

Part2. Function Points Method

Step1: Count the total Unadjusted FP(UFP)

Domain Characteristic Table

MEASUREMENT PARAMETER	COUNT (value >= 0)	WEIGHTING FACTOR Simple	Average	Complex
Number of User Input	6	○	◉	○
Number of User Outputs	6	○	◉	○
Number of User Inquiries	6	○	◉	○
Number of Files	1	○	◉	○
Number of External Interfaces	0	◉	○	○

Complexity Adjustment Table | FP Calculation

Step2: Rate the 14 general system characteristics(GSC)

Complexity Adjustment Table

ITEM	COMPLEXITY ADJUSTMENT QUESTIONS	SCALE No Influence 0	1	2	3	4	Essential 5
1	Does the system require reliable backup and recovery?	◉	○	○	○	○	○
2	Are data communications required?	○	○	○	○	○	◉
3	Are there distributed processing functions?	◉	○	○	○	○	○
4	Is performance critical?	○	○	○	○	○	◉
5	Will the system run in an existing, heavily utilized operational environment?	○	○	○	○	○	◉
6	Does the system require on-line data entry?	◉	○	○	○	○	○
7	Does the on-line data entry require the input transaction to be built over multiple screens or operations?	◉	○	○	○	○	○
8	Are the master files updated on-line?	○	○	○	○	○	◉
9	Are the inputs, outputs, files or inquiries complex?	○	○	○	○	○	◉
10	Is the internal processing complex?	○	○	○	○	○	◉
11	Is the code to be designed reusable?	○	○	○	○	○	◉
12	Are conversion and installation included in the design?	○	○	○	○	○	◉
13	Is the system designed for multiple installations in different organizations?	○	○	○	○	○	◉
14	Is the application designed to facilitate change and ease of use by the user?	○	○	○	○	○	◉

Domain Characteristic Table | FP Calculation

Step 3: Compute the Function Point count

RESULT	
PROJECT FUNCTION POINTS	101.19999999999999

Top of Page | Domain Characteristic Table | Complexity Adjustment Table

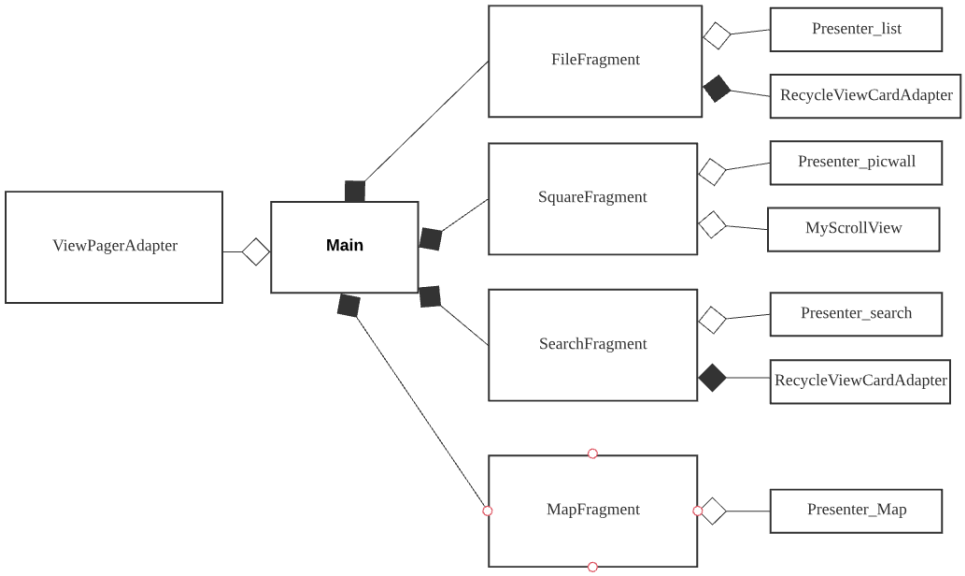
Step 4: Determine the SLOC estimate
Our programming language is Java so there are 46 lines per function point, that is:
SLOC = 46 lines * 101 function points = 4646 lines

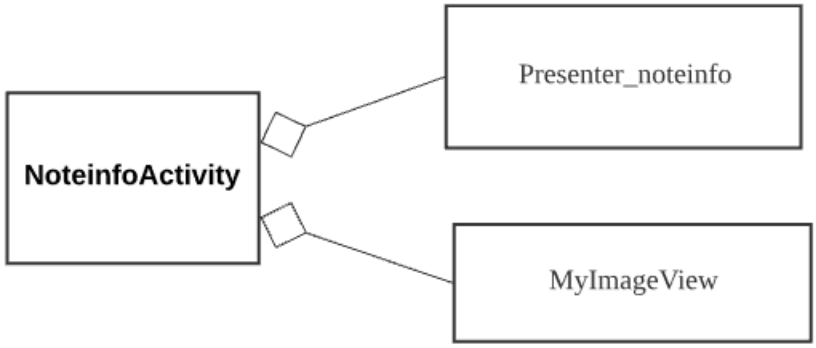
Part3.

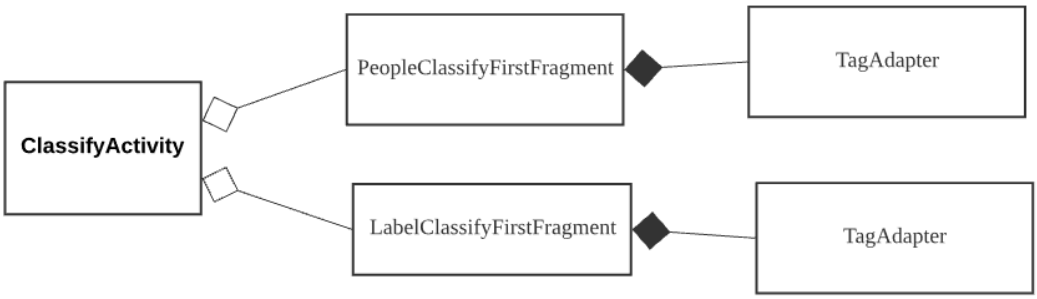
We use Agile as our process model. It is effectively to adapt to a new development plan when our team decided to implement new features into our system after interviewing with our potential users. Also, we get to respond to change feature detail after our first sprint. Therefore, Agile keeps us to be able to make continuous improvements and changes plan quickly. However, we got less predictability and cannot quantify the full extent of the required efforts. since we cannot get well-known customer requirements. Although the incremental delivery style might help our system to be completed faster, the complete output could become fragmented but we want one cohesive unit.

Design

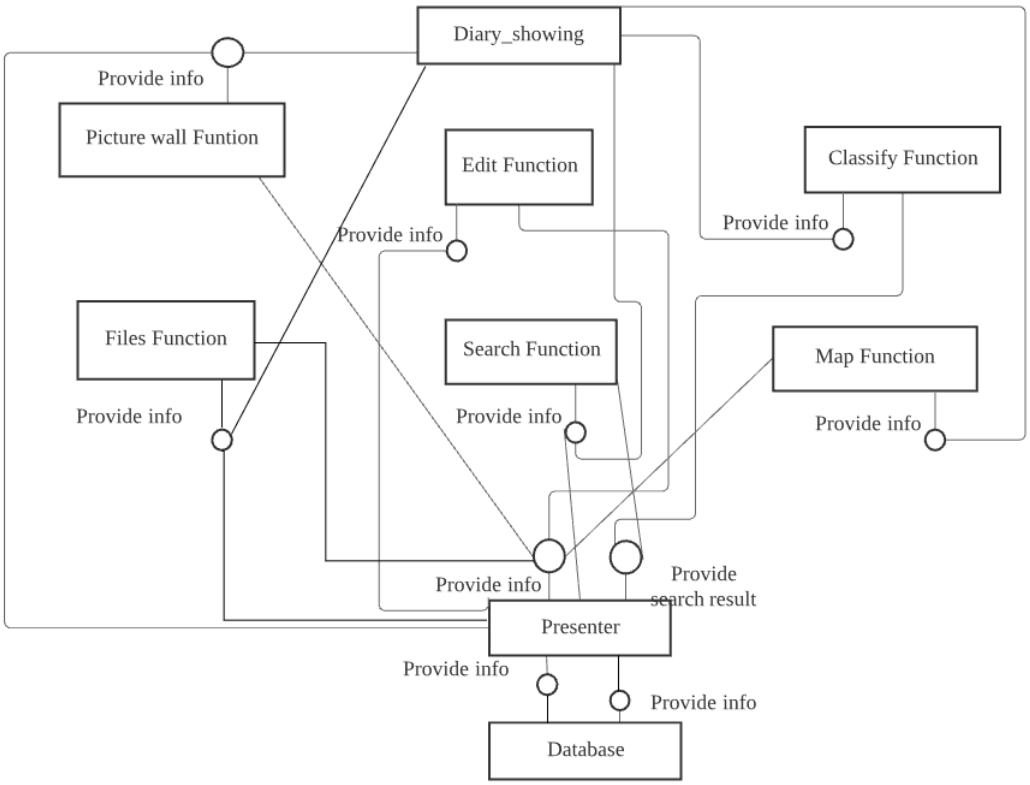
UML Class Diagram:



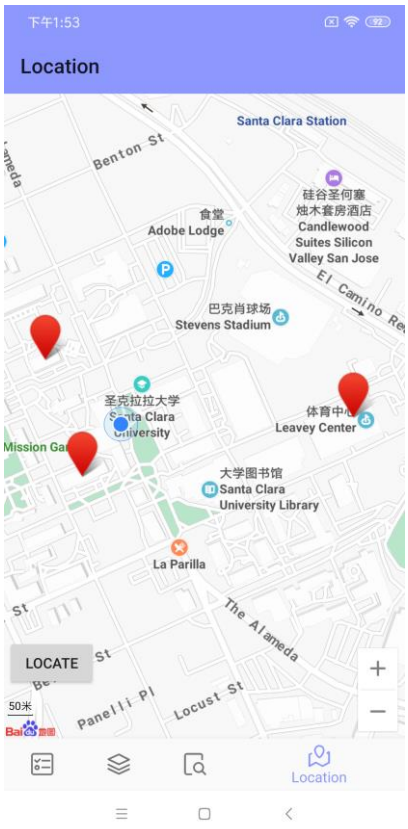


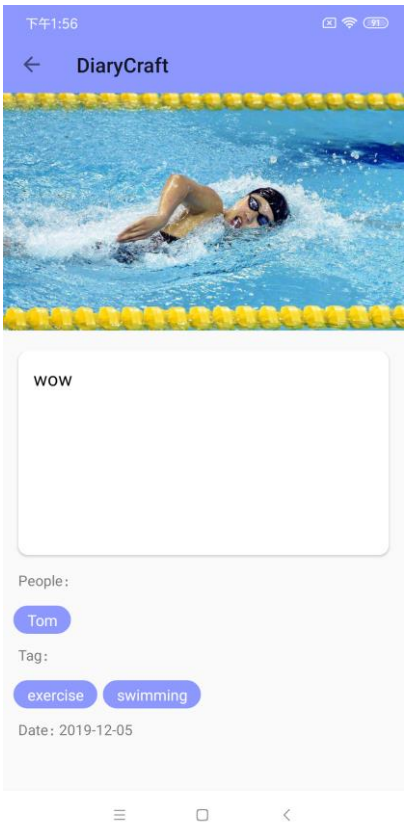


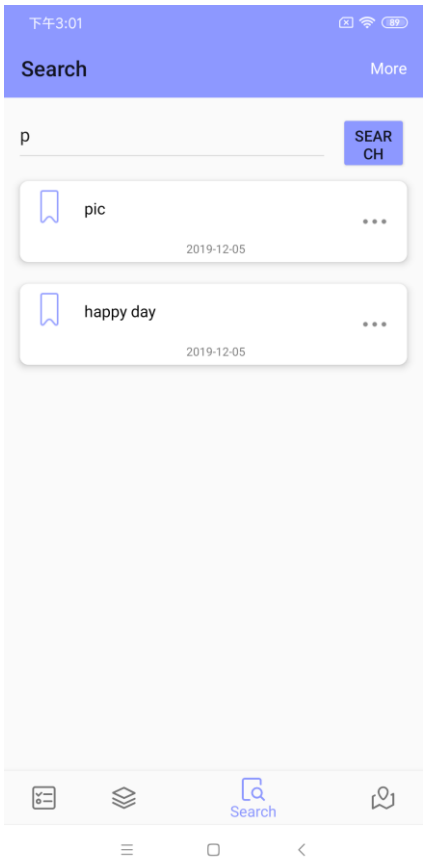
Component Diagram:

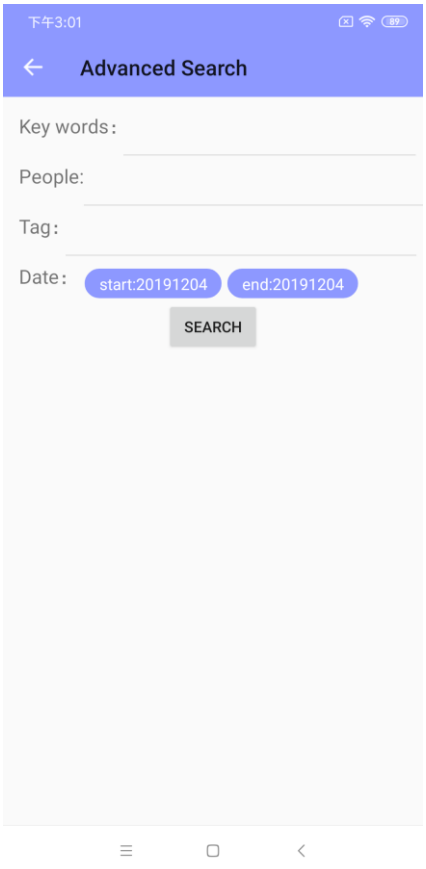


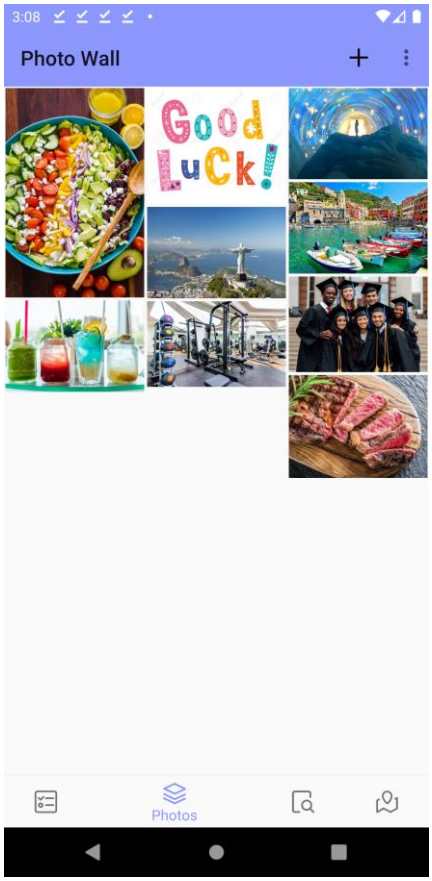
Screenshot of User Interface:






下午3:01
Search
More
p
SEAR CH
pic
2019-12-05
happy day
2019-12-05
Search


下午3:01
Advanced Search
Key words:
People:
Tag:
Date:
start:20191204
end:20191204
SEARCH


3:08
Photo Wall
Good Luck!
Photos

References

Waterfall photo wall: https://blog.csdn.net/yangwen123/article/details/11311273