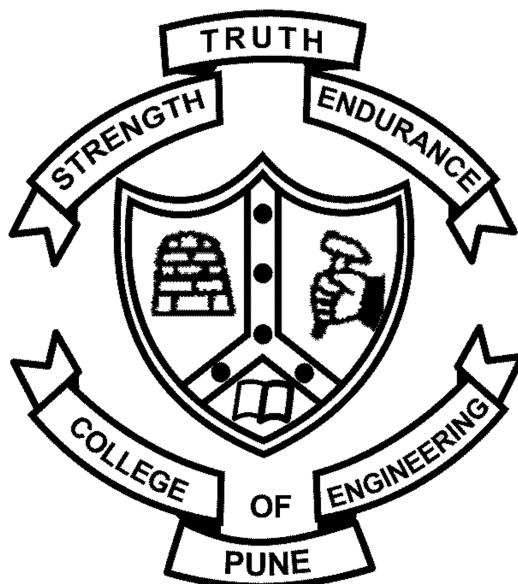


Mini-Project on
D A Y O N E
Analytical Journal

Submitted by

Pranav Sarda (111408053)

Sakshi Nimbarate (111408038)



Department of Information Technology

College of Engineering, Pune

Mrs.Rohini Sarode

Prof.Vandana Inamdar

External

Project-Guide

Head of Department

Project In-charge

Contents

1 Abstract	3
2 Introduction	4
3 Literature Review	5
4 Requirement Specification	6
4.1 Functional Requirements	6
4.2 Non functional requirements	6
4.2.1 Safety requirement :	6
4.2.2 Hardware requirement for developers :	6
4.2.3 Hardware requirement for users:	6
4.2.4 Software requirements for developers:	6
4.2.5 Software requirements for users :	6
4.3 Constraints :	7
5 Proposed System	8
5.1 Process Model:	8
5.2 The incremental model was chosen for the following reasons:	8
5.3 Problem Statement	8
5.4 Problem Definition	8
5.5 Design	9
5.5.1 DFD Level 0	9
5.5.2 DFD Level 1	10
5.5.3 DFD Level 2	11
5.5.4 Usecase Diagram	12
5.5.5 Activity Diagram	13
5.5.6 State Diagram	14
5.5.7 ERD	15
5.5.8 Component Diagram	16
5.5.9 Class Diagram	17
5.6 Module Description	18
5.6.1 System Architecture :	18
5.6.2 Login	19
5.6.3 Sign-up	20
5.6.4 Write new entry	21
5.6.5 Modify old entry	22
5.6.6 View all logs	23
5.6.7 Analyse	24
5.6.8 Push to cloud	25
5.6.9 Logout & Dashboard	26
6 Project Schedule and Estimation	27
6.1 Lines of code	27
6.2 COCOMO model	28
6.3 Gantt chart	29
7 Testing and Result	30
8 Conclusion and future Scope	33
8.1 Conclusion	33
8.2 Future Scope	33
9 References	34

1 Abstract

Abstract

When it comes to keeping a journal, stereotypes quickly come to mind; 'Dear diary' is reserved for the awkward recluse. Others see writing merely as a tool, a pragmatic means to an end, certainly without value in and of itself. But science continues to dissolve skepticism. For those sitting on the fence, keeping a journal has tangible benefits: Evoking mindfulness, sharpening emotional intelligence, improving communication skills and healing.

DAYONE offers a beautiful and interactive user interface to keep all your secrets inside it. There is password protection to avoid unauthorized access to your secrets. If in any case your system needs to be formatted, DAYONE offers safety feature of cloud backup so that your secret can be safe with you on your drive. It can analyze your happy, sad days as per your need. We believe, in the race of this life, everyone is tensed,happy,sad and one needs to record and analyze it for their better future. The intention is to make incorporating journaling into your life smoother, for somewhere, something incredible is waiting to be written.

2 Introduction

The idea of the project is to develop an offline personal diary, with some analytical features. This project will be done using Java (as the main language) and JavaFx will be used to develop the UI. In addition to this, SQL database connectivity will be provided. Using this application, the user can protect his/her entries using password authentication. The basic functionalities that will be provided are: making an entry, viewing and updating it, deleting the entries. The entries can either be text or image along with the option of uploading audio as well as video. The distinctive feature of this project will be a weekly and monthly analysis of past records. The analysis will be shown diagrammatically in the form of pie-charts and line-charts (graphs).

DOMAIN – Desktop Application Development.

3 Literature Review

Table 1: Review Table on currently available Similar Projects

Feature	Journey	Diario	D A Y O N E(mini project)
Platform	Android/Mac/Windows	Android/Web/iOS	OS supporting JDK8+
Cost	In app purchase	In app purchase	Free
Sync	Yes	Yes	Yes
Media Support	Yes	Yes	Yes
Analytical	No	No	Yes
Source	Closed Source	Closed Source	Open Source
Developer	Tow App studio	Pixel Crater, ltd.,	Pranav and Sakshi

4 Requirement Specification

4.1 Functional Requirements

- Sign up and login functionality.
- Keeping profile of user.
- The system should be able to correctly authenticate user.
- The user should be able to make an entry with title.
- The system should accept text, image, audio and video as entries.
- The user should be able to view and edit past entries.
- The system should provide a scale to the user to rate his day.
- On the basis of ratings, the system should be able to provide selective daywise analysis.

4.2 Non functional requirements

4.2.1 Safety requirement :

- Every account must have password.
- Username should be unique.
- User must select backup directory for safety cloud backup.
- User must accept google acceptance form if prompted.

4.2.2 Hardware requirement for developers :

- RAM: 128 MB
- Machine supporting OS with Oracle JDK 8+
- Disk space: 124 MB for JRE;2 MB for Java Update
- Minimum Pentium 2 266 MHz processor.

4.2.3 Hardware requirement for users:

- RAM: 128 MB
- Disk space: 124 MB for JRE;2 MB for Java Update
- Minimum Pentium 2 266 MHz processor.

4.2.4 Software requirements for developers:

- Oracle JDK
- JetBrains IntelliJ IDEA/Eclipse IDE
- JavaFX gluon Scene Builder
- SQLite.jar

4.2.5 Software requirements for users :

- Oracle JRE
- OS: MacOSX, Windows, Unix/Linux

4.3 Constraints :

- Hardware and Software requirements should be fulfilled.
- Authentication is mandatorily required to use application.
- User need to have internet connection in order to use cloud functionality.
- User has to enter the correct answer in order to continue playing.
- User can not enter many entries on same date , however it can be modified.

5 Proposed System

5.1 Process Model:

The incremental model will be used in the development of this project as some specifications may evolve with time. The intermediate versions will be reviewed.

5.2 The incremental model was chosen for the following reasons:

- 1) It is flexible as it gives freedom to modify the project based on review at any point of time.

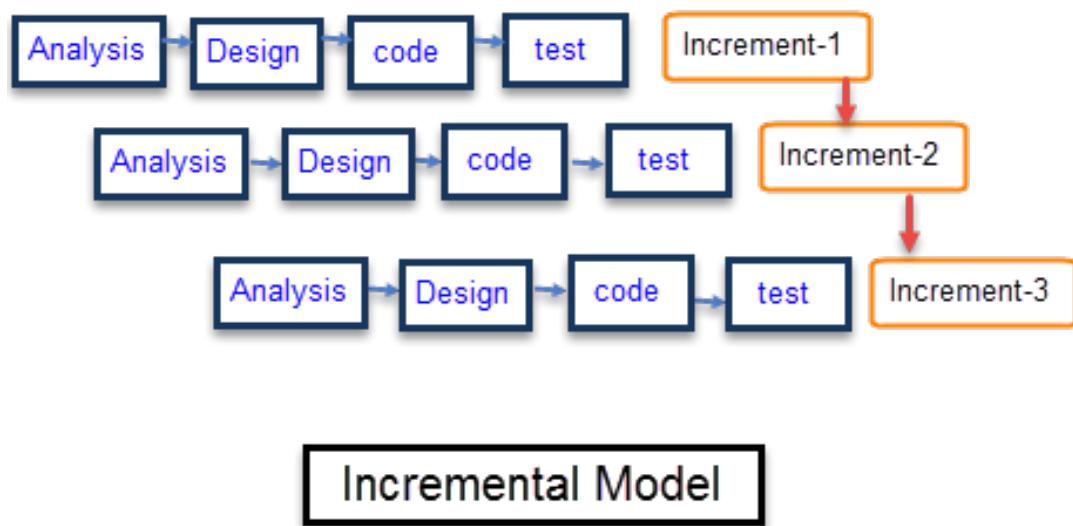


Figure 1: Incremental Model

- 2) After modifications, intermediate versions will be created.

5.3 Problem Statement

To create a desktop journal application for the purpose of providing an analytical, cloud-supported, user-friendly journaling experience.

5.4 Problem Definition

Create a desktop application for journaling that accepts text, image and video as input entries. The entry logs are to be saved in a database and can be viewed and modified later. For each entry, user also provides rating (of the quality of his day) on a scale of one to ten. The user can do a qualitative analysis of his days in a specified time using these ratings. The text entries are to be pushed to the cloud.

5.5 Design

5.5.1 DFD Level 0

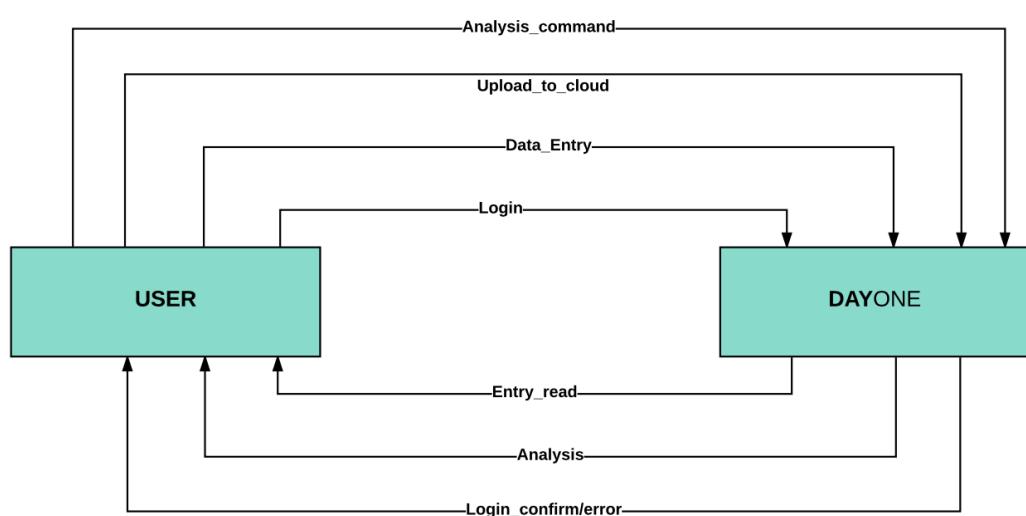


Figure 2: DFD Level 0

5.5.2 DFD Level 1

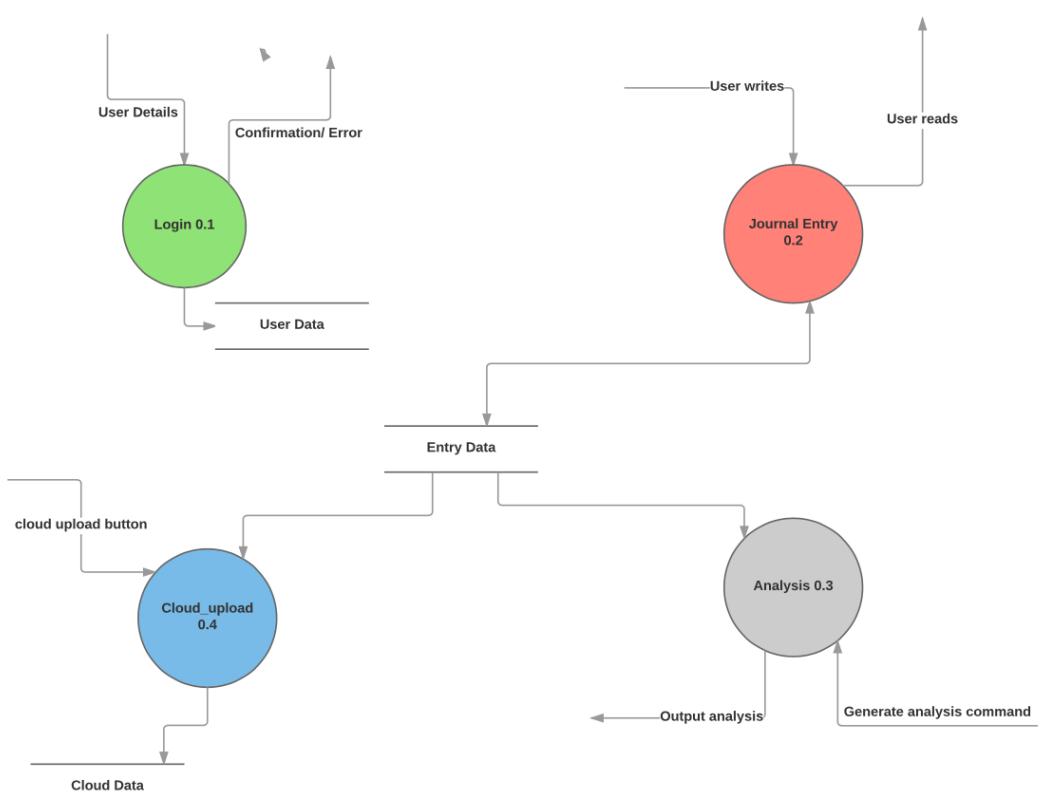


Figure 3: DFD Level 1

5.5.3 DFD Level 2

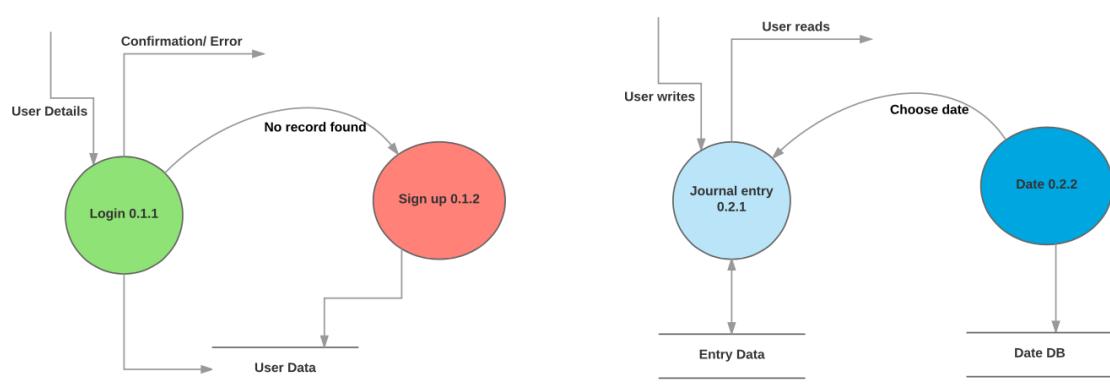


Figure 4: DFD Level 2

UML Diagrams

5.5.4 Usecase Diagram

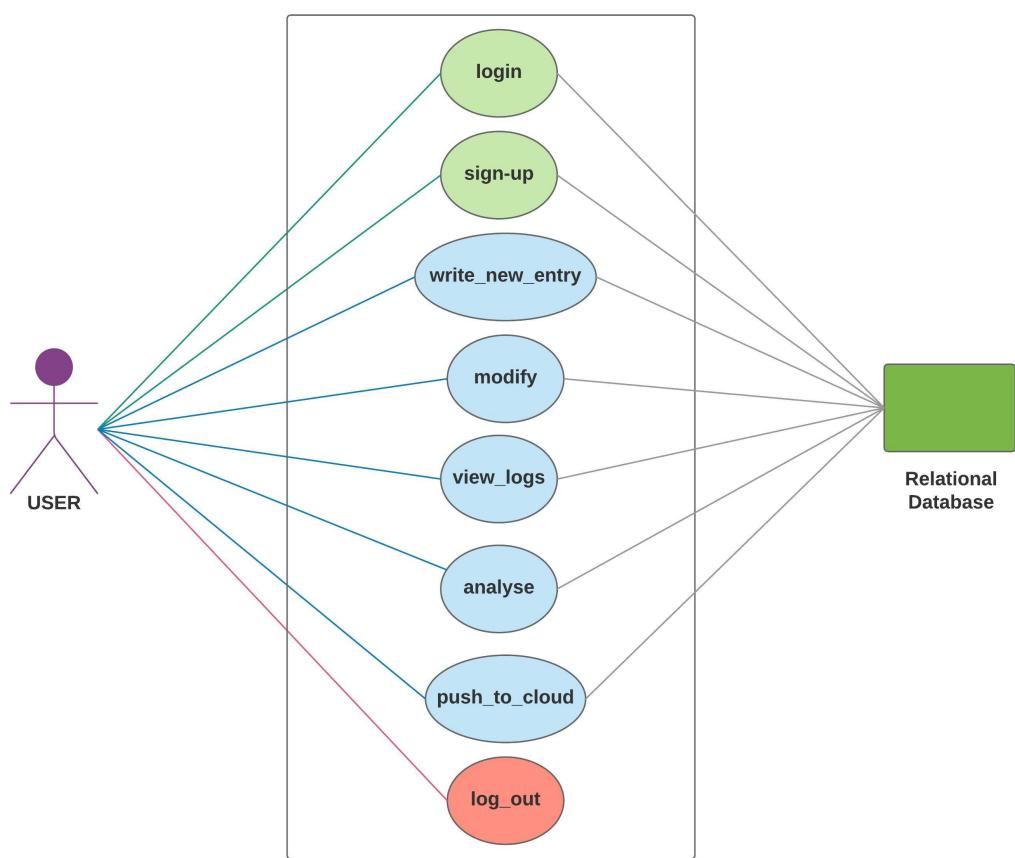


Figure 5: Use-Case Diagram

5.5.5 Activity Diagram

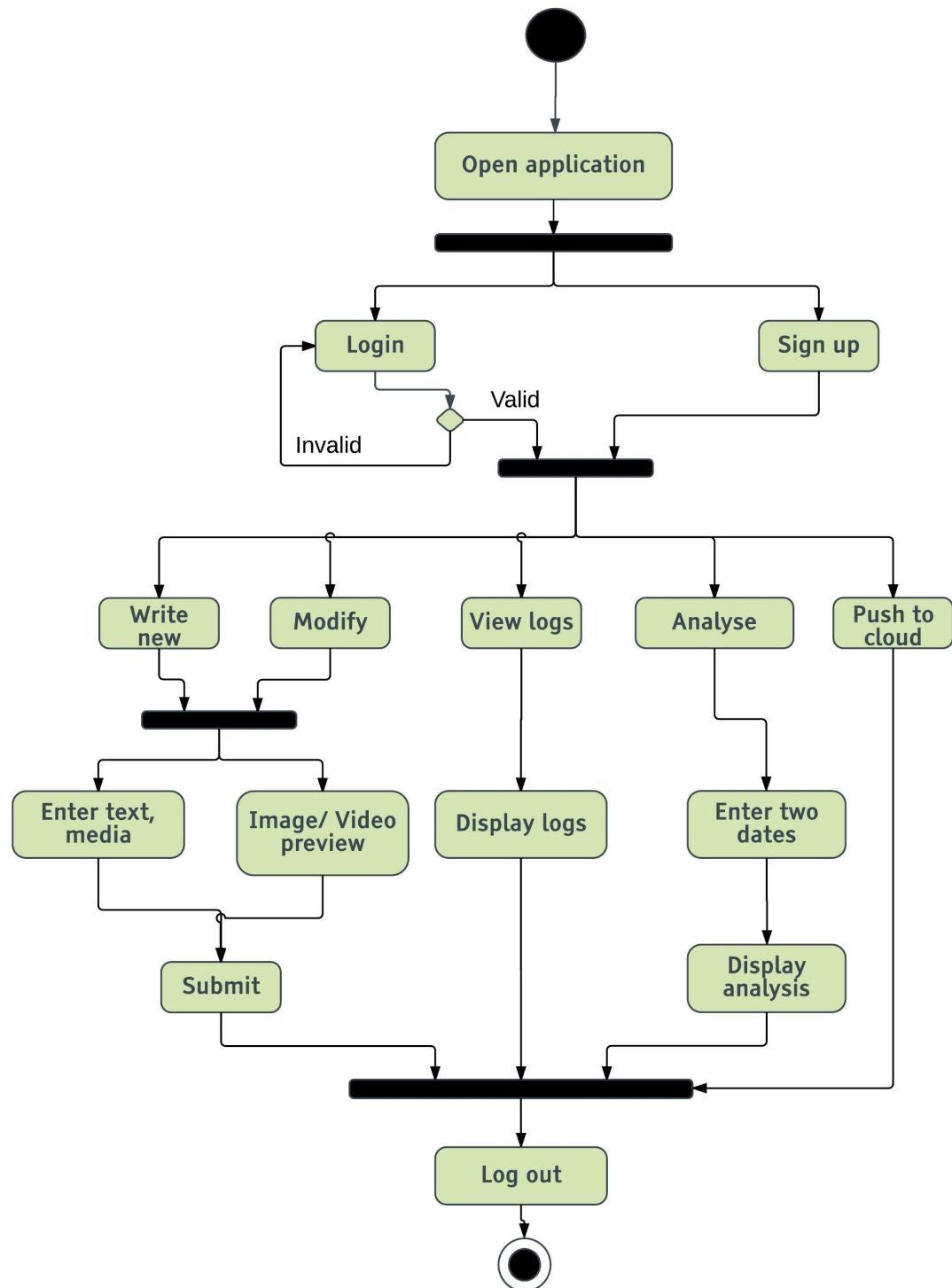


Figure 6: Activity Diagram

5.5.6 State Diagram

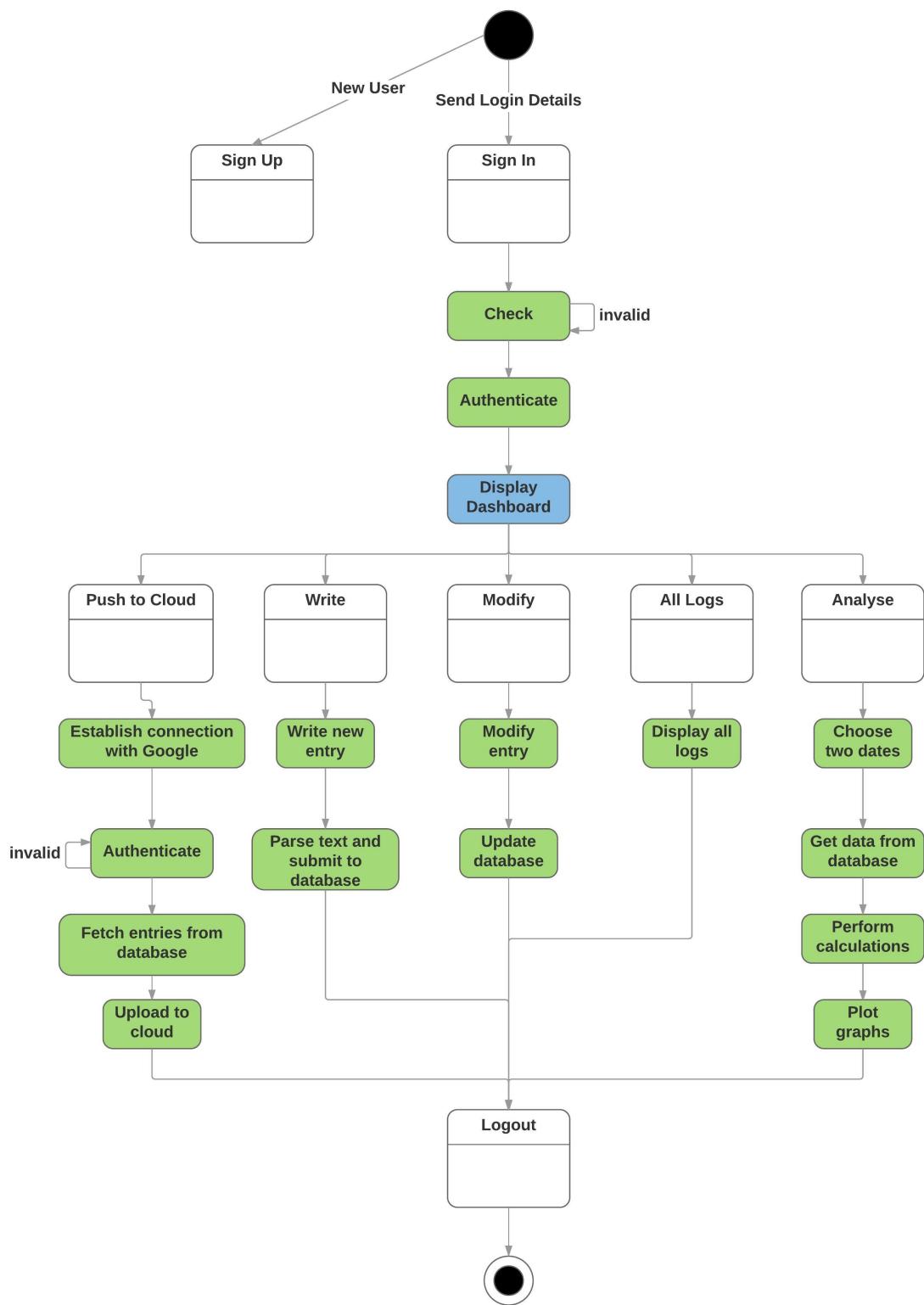


Figure 7: State Diagram

5.5.7 ERD

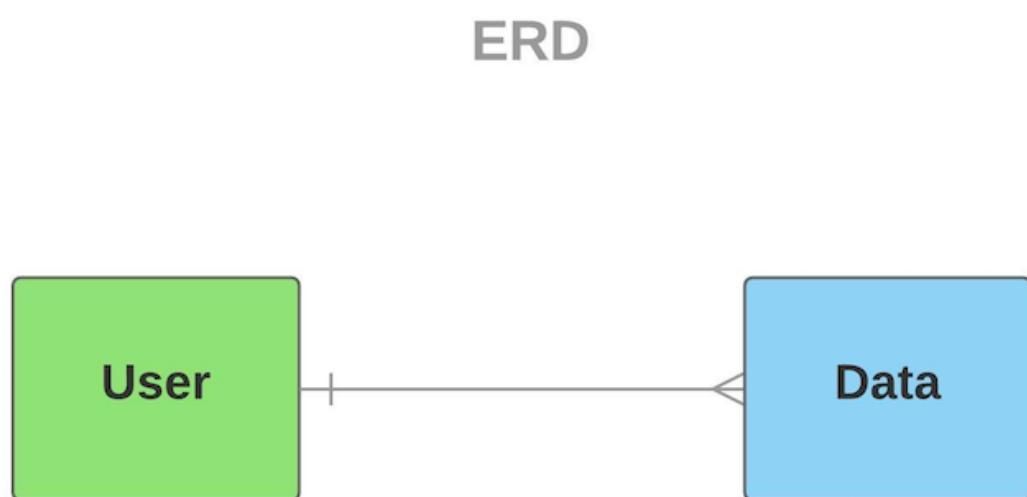


Figure 8: Entity Relationship Diagram

5.5.8 Component Diagram

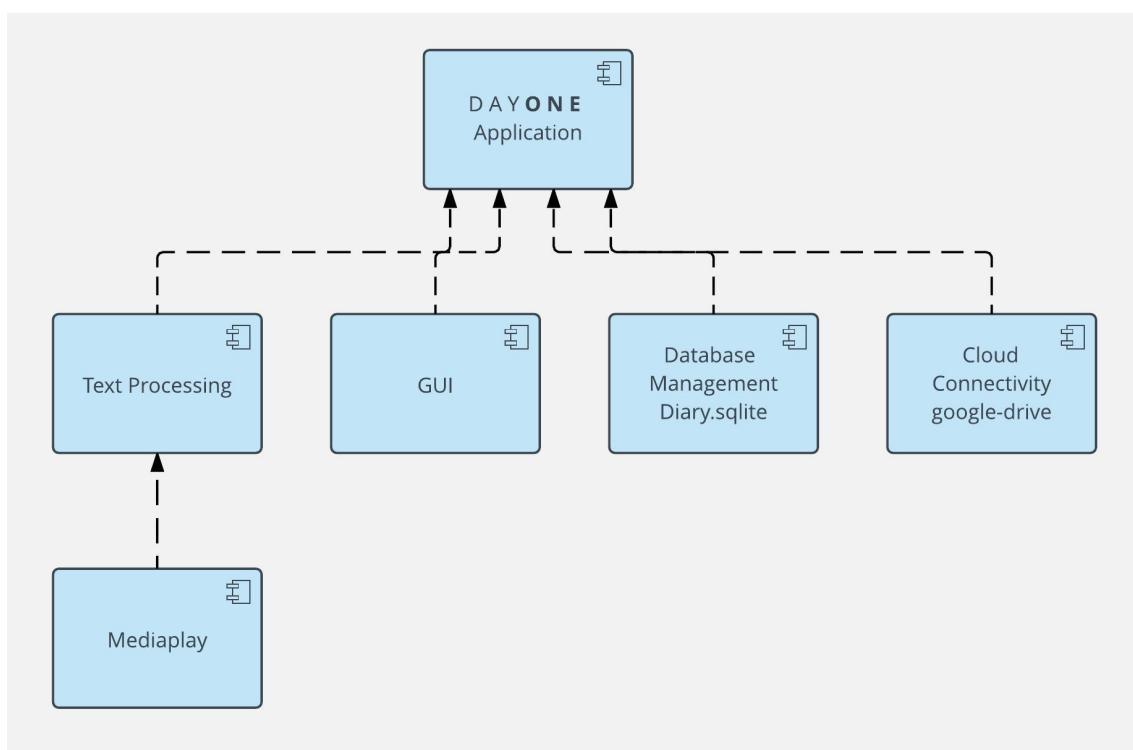


Figure 9: Component Diagram

5.5.9 Class Diagram

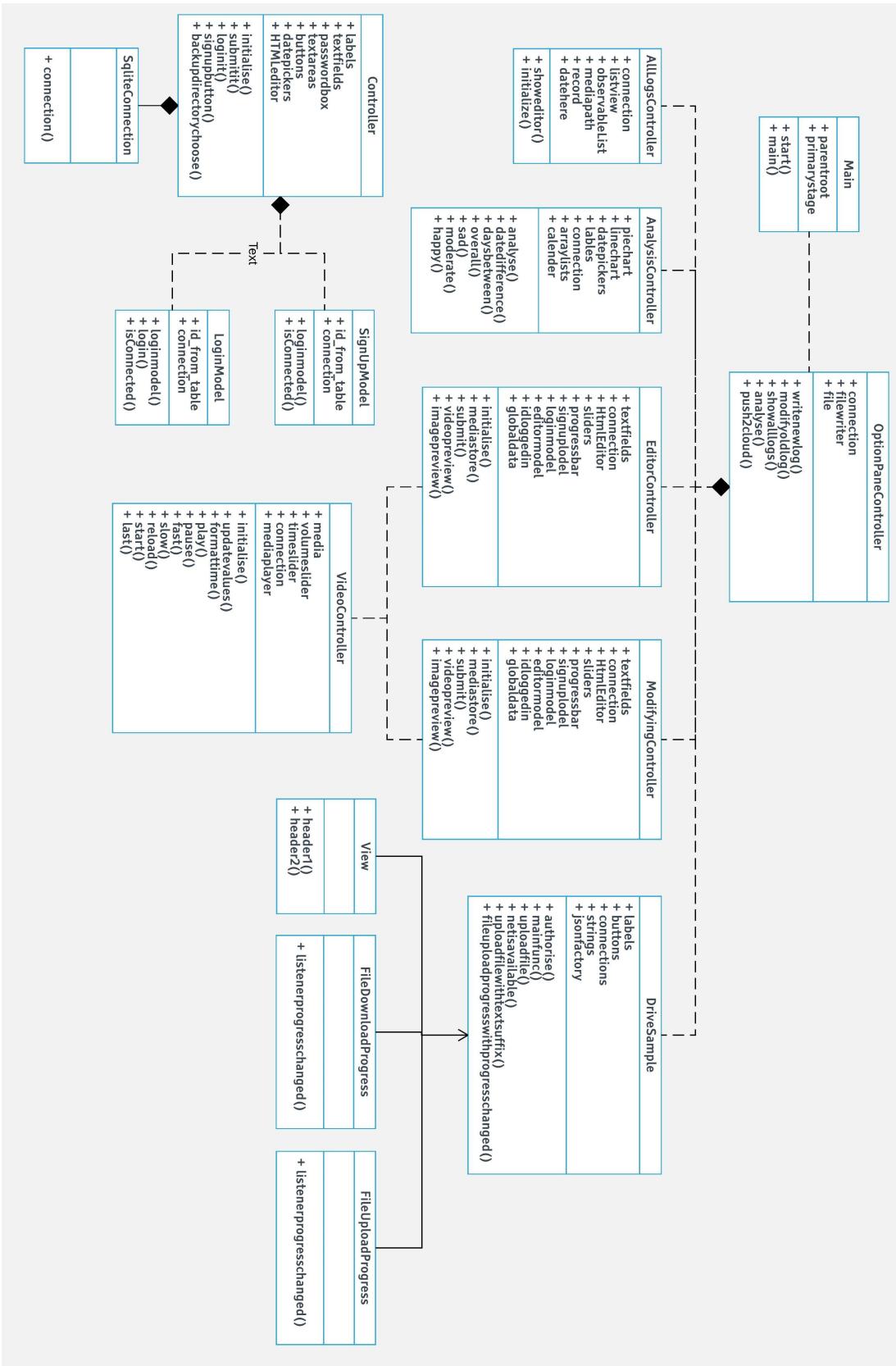


Figure 10: Class Diagram

5.6 Module Description

5.6.1 System Architecture :

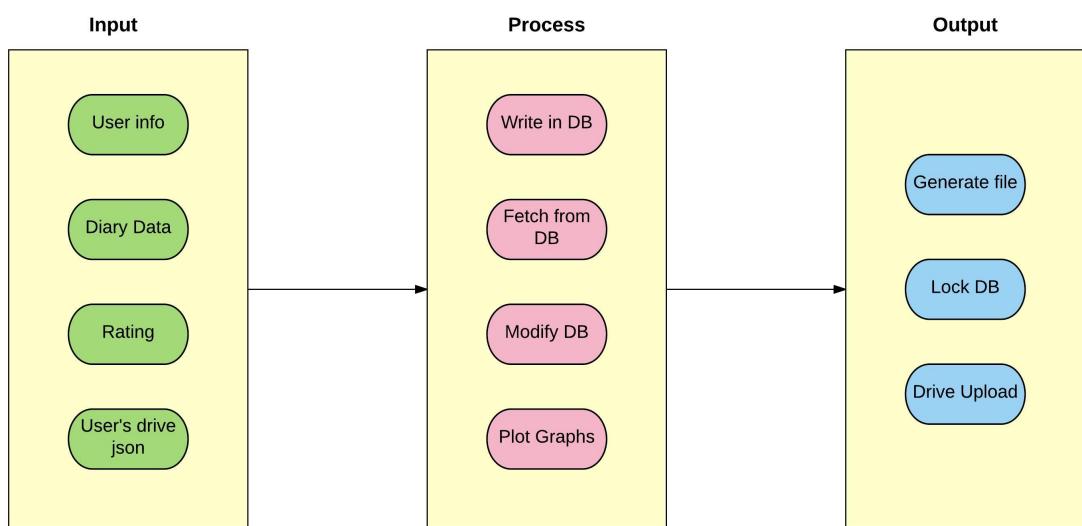


Figure 11: System Architecture Diagram

Modules :

5.6.2 Login

The module allows existing user to access her account.

To login successfully, the user has to provide the correct username and password. If entered password is incorrect, an error message is shown.

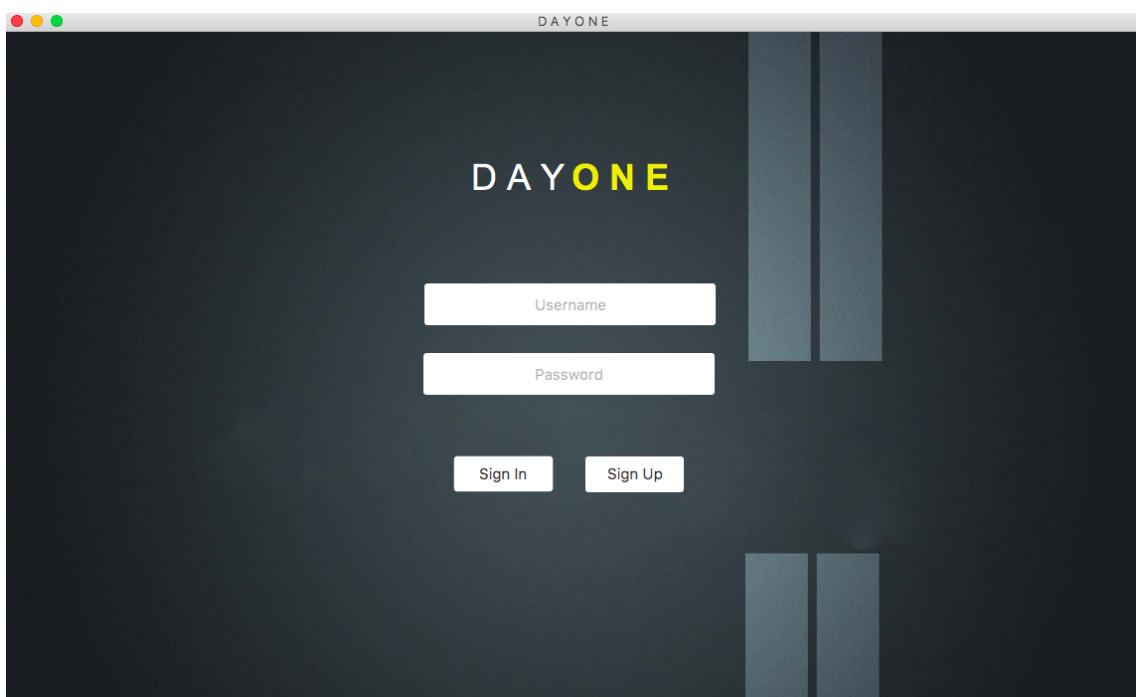


Figure 12: Login Stage

5.6.3 Sign-up

A new user has to sign-up and create her account before she can start using the application. While signing up the following details need to be provided:

1. First name
2. Last name
3. Phone number
4. e-mail id
5. Gender
6. Date of birth
7. Other details (optional)
8. Username
9. Password
10. Back-up directory

Numbers and special characters are not allowed while entering name. Also, e-mail id should be of correct format. e.g. sample@gmail.com

The screenshot shows a sign-up interface with two main sections: 'Personal Details' and 'Account Details'.
Personal Details Section:
- **Name:** Two input fields for 'Name' and 'Last Name'.
- **Email:** Input field containing 'example@example.com'.
- **Ph. No.:** Input field containing 'eg. 9988776655'.
- **Gender:** Radio buttons for 'Male', 'Female', and 'Other'.
- **Date of Birth:** Input field with placeholder 'Select Date' and a calendar icon.
- **Self Description:** A large text area with placeholder text 'Tell me about yourself'.
Account Details Section:
- **Username:** Input field.
- **Password:** Input field.
- **Confirm Password:** Input field.
- **Backup Directory:** Input field.
- **Instructions:** Text 'Please choose backup directory' in red.
- **Submit Button:** A white button labeled 'Submit'.

Figure 13: Sign-up stage

5.6.4 Write new entry

After the user has successfully logged in, she can write a new entry. While doing so, the date is automatically chosen as the current date. The user need to give a title (the default title being 'Title here'.) The content is to be written in a HTML editor.

The user can upload either an image or a video in her entry. The video or image preview can be viewed. Uploading and previewing can be done in a user-friendly manner through buttons.

The user needs to rate her day on a scale of zero to ten with the help of a slider.

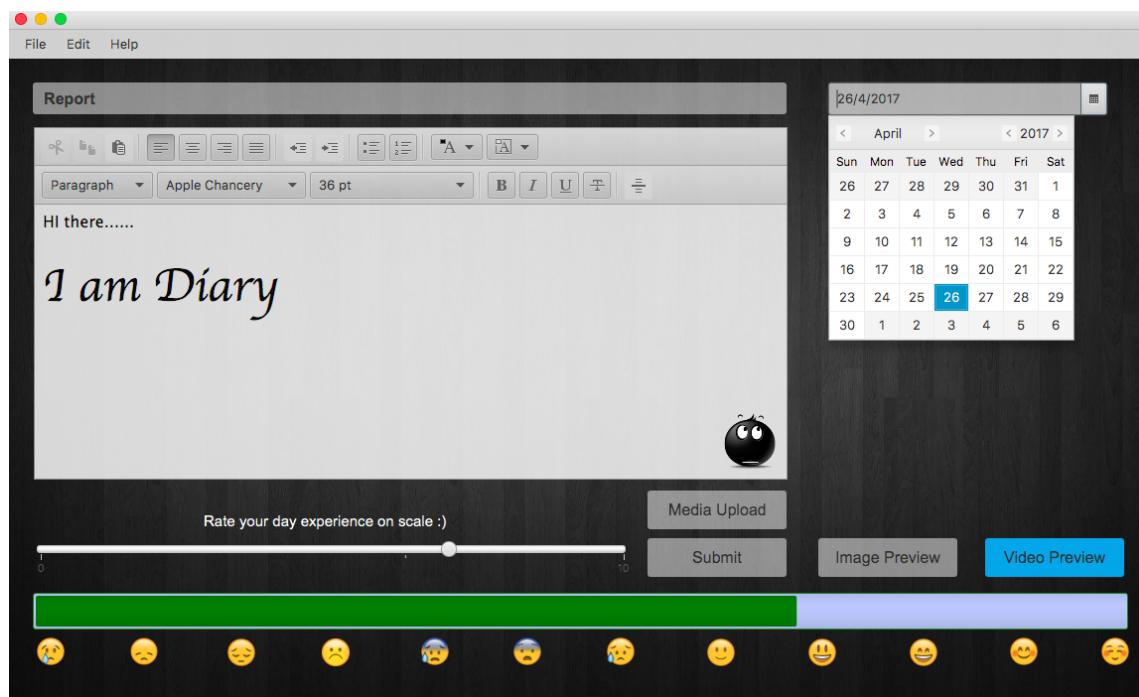


Figure 14: New log

5.6.5 Modify old entry

The user can modify old entries. After opening the modify window, the user has to first choose the date for which the entry she wants modified. After that the user can change the title, content, media uploaded and the rating of that day. The user interface is similar to that of the previous module.

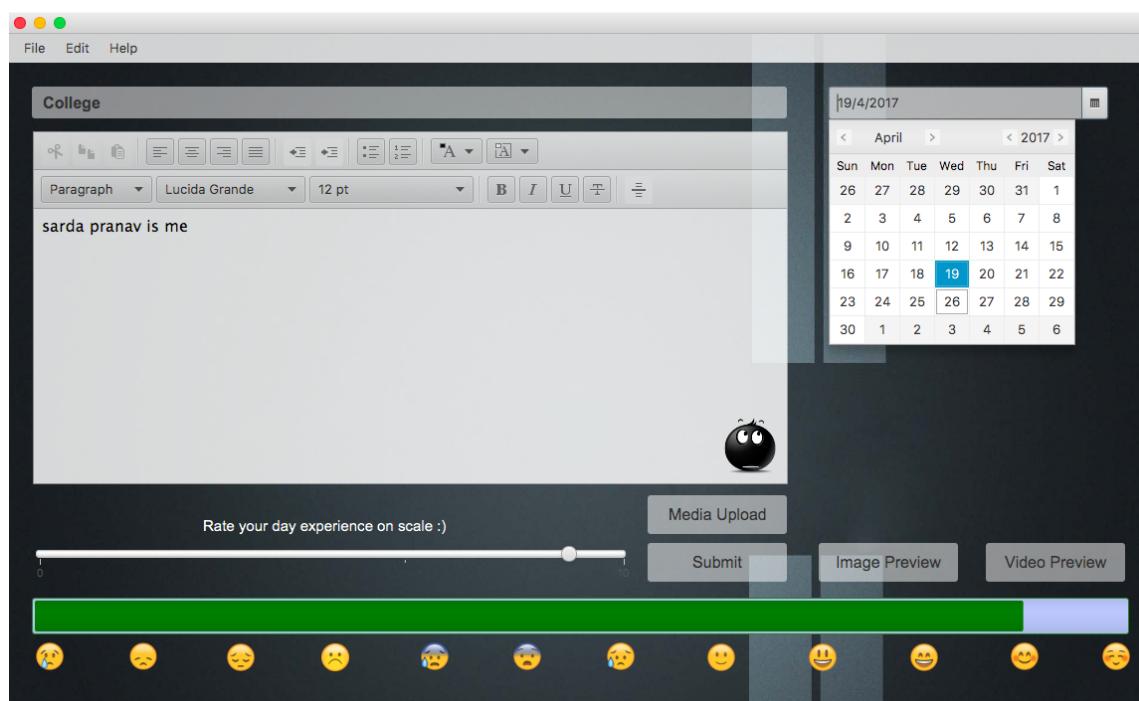


Figure 15: Modify old log

5.6.6 View all logs

The user can view a log of her entries.
The logs are of the format: Sr. no. | Date | Title
The logs are arranged date-wise.

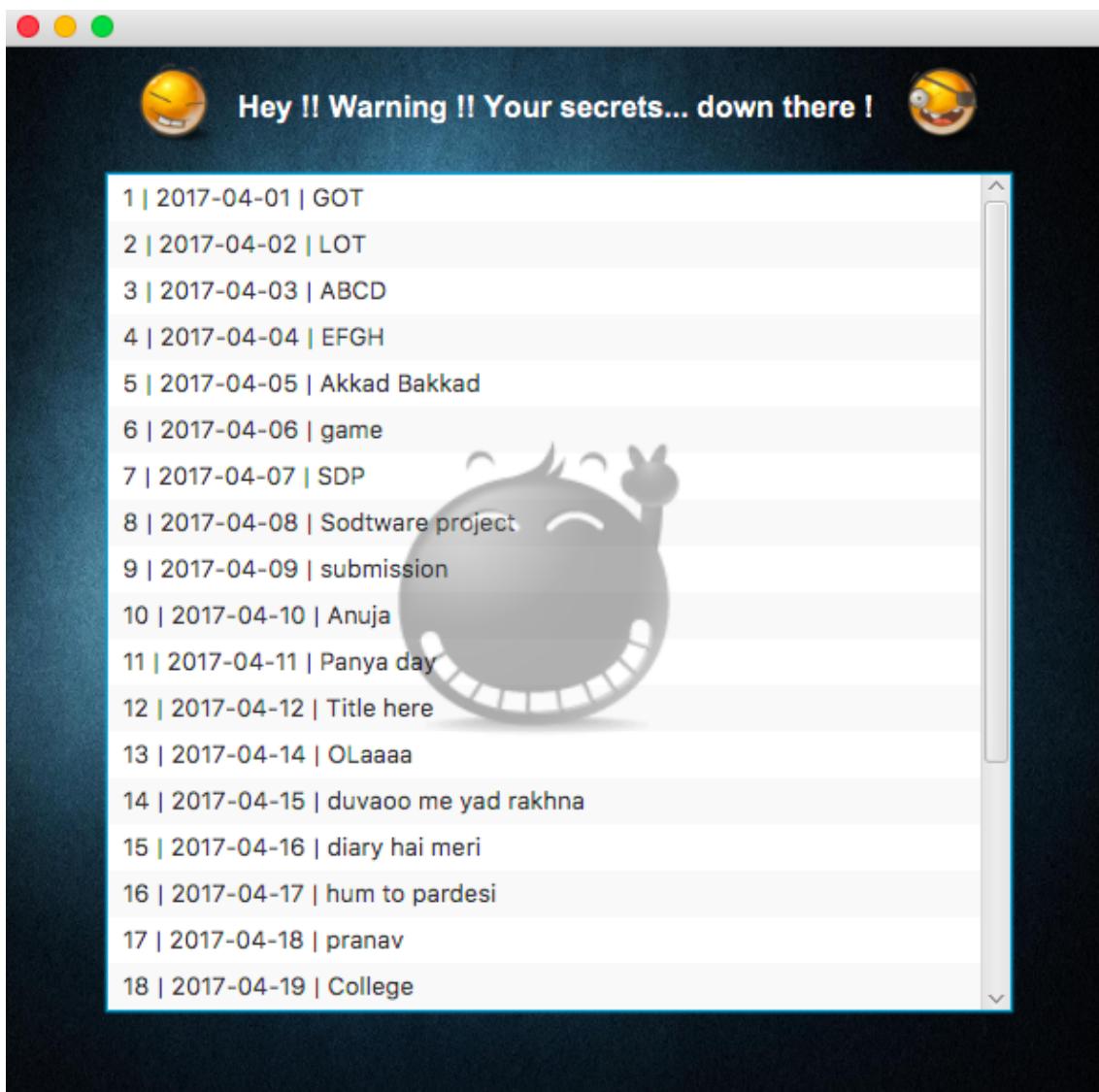


Figure 16: Display all logs

5.6.7 Analyse

This is the feature module of this project. In this module, the user can do analysis of her days based on past entries.

The user needs to choose two dates; the duration between which will be analysed.

Analysis can be done in four flavours: overall, happy, moderate and sad. A line-graph and a pie-chart depict the analysis.

The basis for analysis is the ratings which the user has given for each day. If for some day an entry is missing, then default rating is considered which is five.

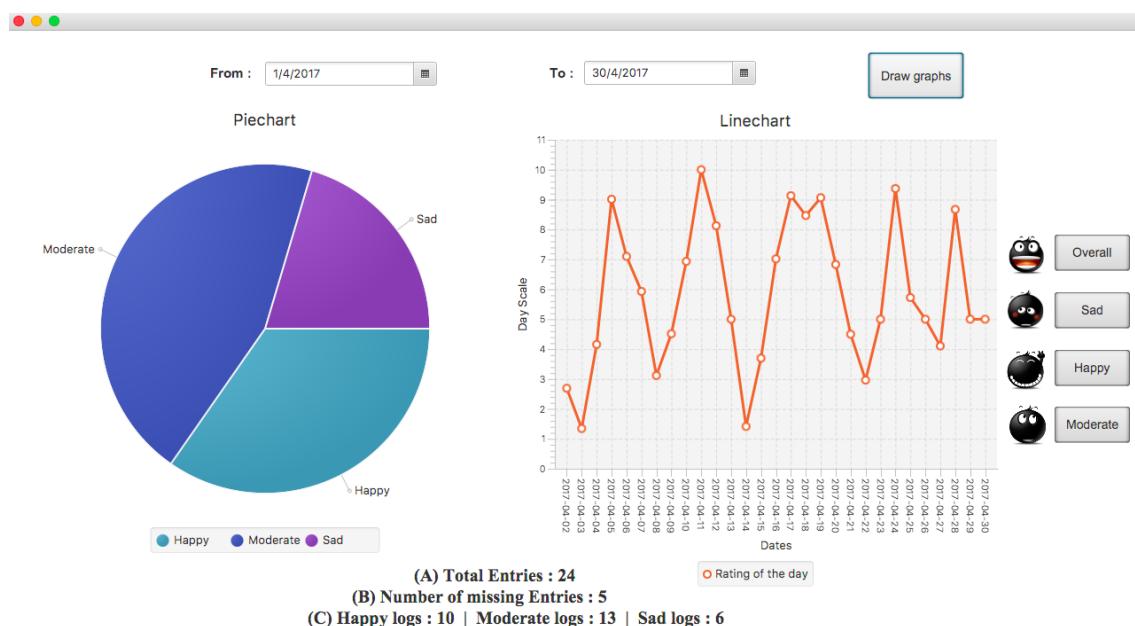


Figure 17: Analyse stage

5.6.8 Push to cloud

This module pushes the content of user's entries to the user's google drive account. The text content of the entries are stored in a .txt format in the backup directory (chosen by the user initially while signing up). These entries are pushed to the cloud cumulatively. This is achieved using the Drive API of the Google App Engine.

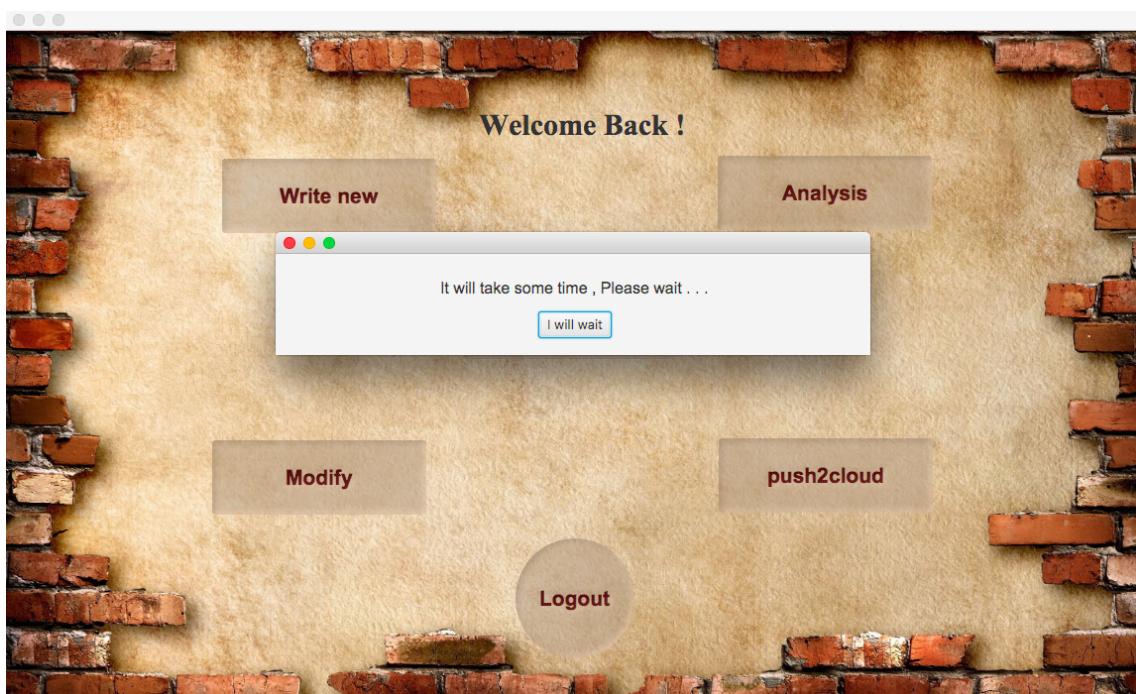


Figure 18: push2cloud confirmation dialogue

5.6.9 Logout & Dashboard

The log-out module enables the user to sign-out of his/her account.

The log-out button is present on the dashboard and takes the user to the sign-in/sign-up page.

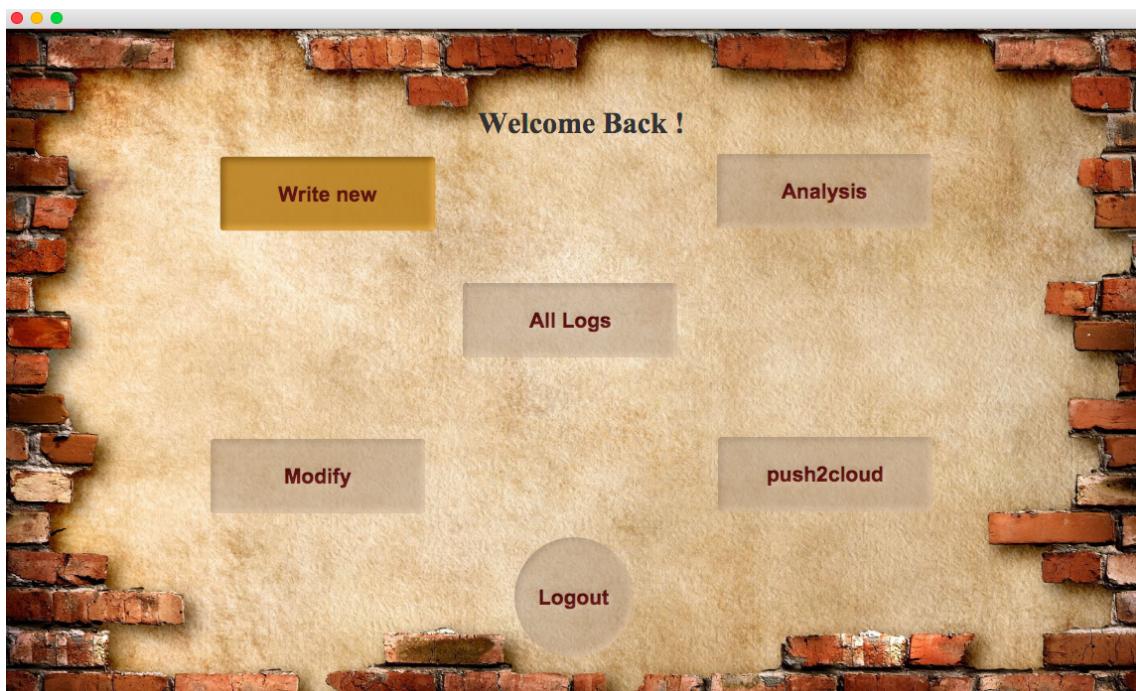


Figure 19: Dashboard

6 Project Schedule and Estimation

6.1 Lines of code

Table 2: LOC count

Class(.java)	LOC
AllLogsController.java	85
AnalysisController.java	210
Controller.java	156
DriveSample.java	303
EditorController.java	193
EditorDatabase.java	75
FileDialogDownloadProgressListener.java	41
FileDialogUploadProgressListener.java	54
LoginModel.java	63
Main.java	21
ModifyingEditorController.java	317
OptionPaneController.java	168
README.md	12
SignUpModel.java	62
SqliteConnection.java	23
ThankYouController.java	14
VideoController.java1	97
View.java	35
TOTAL LOC	
= 2029	

6.2 COCOMO model

Formulas :

Effort = $A \times KLOC^B$, in person/months, with KLOC lines of code, (in the thousands)

Duration = $C \times Effort^D$, in months

Staffing = Effort/Duration

Calculations : Using model - Organic

Table 3: Calculation of Basic COCOMO Model

Mode	'A'var	'B'var	'C'var	'D'var	KLOC	EFFORT(pm)	DURATION(m)	STAFFING
Organic	2.4	1.05	2.5	0.38	2.029	5.04495	4.62408	1.091

6.3 Gantt chart

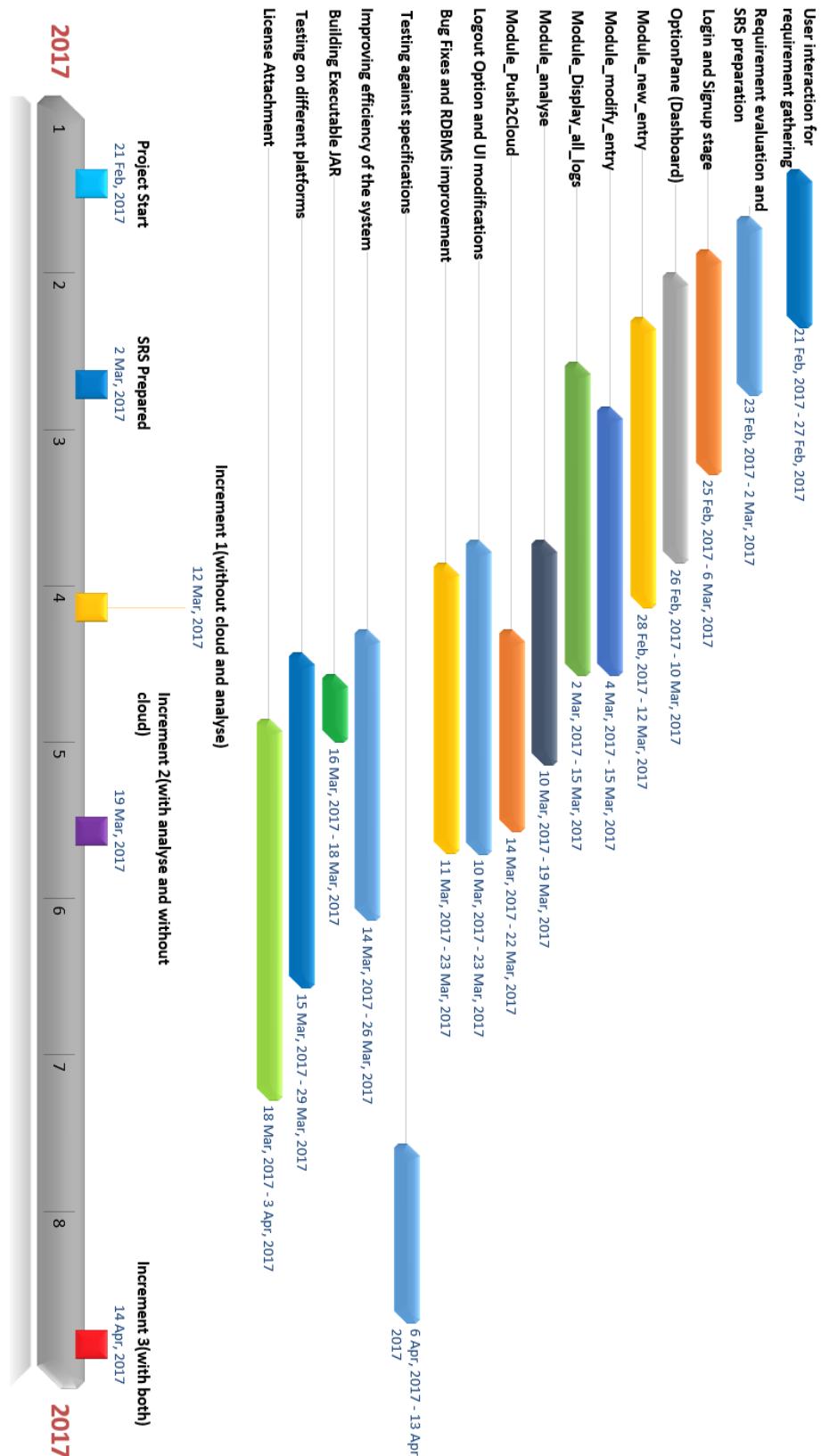


Figure 20: Gantt Chart (TimeLine)

7 Testing and Result

Test cases :

- ID 1. Login
- ID 2. Password matching while signing up
- ID 3. Entry added to database
- ID 4. Entry is updated in database
- ID 5. All logs relating to the account to be visible
- ID 6. Internet availability while pushing to cloud
- ID 7. Logout

(A)

Test Case ID:1

Purpose:

User logs into the application

Assumption:

Username and password are available in database.

Pre-condition:

Log-in page is displayed.

Steps:

1. User enters username and password.
2. System does authentication.

Expected outcome:

User gets logged into the system if valid credentials are provided, otherwise error message is thrown.

Actual outcome:

User gets logged into the system if valid credentials are provided, otherwise error message is thrown.

Post-condition:

Dashboard window opens.

Result:

Success

(B)

Test Case ID: 2

Purpose:

Password matches while signing up

Assumption:

New user is signing up and has entered password once.

Pre-condition:

Sign-up page is displayed.

Steps:

1. User enters password and then re-enters it for confirmation.
2. System does authentication

Expected outcome:

User account gets created successfully if both passwords entered match, otherwise error message is thrown.

Actual outcome:

User account gets created successfully if both passwords entered match, otherwise error message is thrown.

Post-condition:

Message showing successful registration is shown.

Result:

Success

(C)

Test Case ID: 3

Purpose:

Entry is added to database

Assumption:

User has made a new entry.

Pre-condition:

Write-new widow is open.

Steps:

1. User enters content into the diary for a date
2. System submits the entry into the database

Expected outcome:

The entry gets added to the database successfully.

Actual outcome:

The entry gets added to the database successfully.

Post-condition:

Message showing successful addition to database is shown.

Result:

Success

(D)

Test Case ID: 4

Purpose:

Entry is updated to database

Assumption:

User has modified an old entry.

Pre-condition:

Modify widow is open.

Steps:

1. User modifies the content of the diary for a date
2. System updates that entry in the database

Expected outcome:

The entry gets updated in the database successfully.

Actual outcome:

The entry gets updates in the database successfully.

Post-condition:

Message showing successful updation to database is shown.

Result:

Success

(E)

Test Case ID: 5

Purpose:

All logs relating to an account are shown

Assumption:

User has selected view logs button.

Pre-condition:

Dashboard is open.

Steps:

1. User selects the view logs button on the dashboard
2. System opens a window showing all previous logs of the user

Expected outcome:

A window showing all previous logs of the user is opened.

Actual outcome:

A window showing all previous logs of the user is opened.

Post-condition:

A window showing all previous logs of the user is opened. On closing it, user is redirected to dashboard.

Result:

Success

(F)

Test Case ID: 6

Purpose:

Internet connection must be available if entries are to be pushed to the cloud

Assumption:

User has selected push2cloud button.

Pre-condition:

Dashboard is open.

Steps:

1. User selects the push2cloud button on the dashboard
2. System checks whether an internet connection is available.

Expected outcome:

If internet connection is available, entries are pushed to the user's drive account.

If not, an appropriate error message is shown.

Actual outcome:

If internet connection is available, entries are pushed to the user's drive account.

If not, an appropriate error message is shown.

Post-condition:

After either successful or unsuccessful operation, user is redirected to dashboard.

Result:

Success

(G)

Test Case ID: 7

Purpose:

User logs out of the application

Assumption:

User has selected logout button on dashboard.

Pre-condition:

Dashboard is displayed.

Steps:

1. User selects logout button
2. System logs her out and redirects to the login page

Expected outcome:

User gets logged out of the system and redirected to login page.

Actual outcome:

User gets logged out of the system and redirected to login page.

Post-condition:

Login window opens.

Result:

Success

8 Conclusion and future Scope

8.1 Conclusion

The project (DAYONE - Analytical Journal desktop application) was designed and built using the incremental model of software development.

Three increments of the project were made. Each increment built and improved on the previous increments. The first increment had basic functionalities of writing, modifying and viewing logs.

In the second increment, the analysis module was perfected and integrated into the first increment. Lastly, in the third increment, the cloud feature was assimilated into the application.

The project helped in getting a hands-on experience of using platforms like IntelliJ IDEA and Gluon Scene Builder and strengthened concepts of JAVA, JAVAFX, SQLITE.

It assisted in learning about cloud services as the Drive API of Google App Engine was used.

8.2 Future Scope

Certain in-app changes can be implemented to make the application robust. Some of them are: stronger password security measures, designing and implementing a more comprehensive UI, etc.

Currently, only the text entries are being cumulatively pushed to the cloud. This can be improved by making the pushes real-time and non-cumulative which has significant impact on saving time and data.

Apart from text entries, media uploads can also be pushed to the cloud. Google Drive limitations can be taken care of by using other cloud service providers like Amazon Web Services or Windows Azure.

The entire application can be deployed to the cloud to be accessed as SaaS.

9 References

Community

www.stackoverflow.com

Existing Journal App

www.diarioapp.com

www.2appstudio.com/journey/

Research paper on Sentimental analysis using product review data

<https://journalofbigdata.springeropen.com/articles/10.1186/s40537-015-0015-2>