

SOFTWARE ENGINEERING PROJECT SYNOPSIS/PROJECT PROPOSAL

Project Title: Garble - an encrypted journal

Team profile:

1. Keerthi R - PES1UG20CS205
2. Maani Jacob Manayath - PES1UG20CS241
3. Mahesh KM - PES1UG20CS242
4. Mahika Gupta - PES1UG20CS243

Proposed Project Description:

A journal in which entries made can be encrypted to maintain confidentiality.

Our target audience consists of journalists, writers, regular travelers, youngsters who want to preserve their privacy, professionals who want to protect their intellectual property, and entrepreneurs who seek to store and keep track of their ideas.

Features include:

- The journal entries are encrypted.
- Password protection.
- The program also analyses the entries, based on which the user's moods on any particular day are tracked.
- Database online to back up the data for ease of access across devices.

Plan of Work and Product Ownership:

In the next few weeks, we plan to:

- Create a structured prototype of our web application. This will be done by Keerthi.
- Basic functionality to enter the journal entries. This will be done by Maani and Mahesh.
- Login using a password, for the user to access his entries, done by Mahika.
- Look into the encryption methods to be used to encrypt the journal entries, done by Mahika.



PES UNIVERSITY, BANGALORE

Department of Computer Science and Engineering

Software Requirements Specification

for

Garble: An Encrypted Journal

Version 1.0 approved

Prepared by Team 5

Team:

1. Keerthi R - PES1UG20CS205
2. Maani Jacob Manayath - PES1UG20CS241
3. Mahesh KM - PES1UG20CS242
4. Mahika Gupta - PES1UG20CS243



PES UNIVERSITY, BANGALORE

Department of Computer Science and Engineering

Table of Contents

Table of Contents ii Revision History ii 1. Introduction 1

1.1 Purpose	1
1.2 Intended Audience and Reading Suggestions	1
1.3 Product Scope	1
1.4 References	1
2. Overall Description	2
2.1 Product Perspective	2
2.2 Product Functions	2
2.3 User Classes and Characteristics	2
2.4 Operating Environment	2
2.5 Design and Implementation Constraints	2
2.6 Assumptions and Dependencies	3
3. External Interface Requirements	3
3.1 User Interfaces	3
3.2 Software Interfaces	3
3.3 Communications Interfaces	3
4. Analysis Models	
5. System Features	4
5.1 System Feature 1	4
5.2 System Feature 2 (and so on)	4
6. Other Nonfunctional Requirements	4
6.1 Performance Requirements	4
6.2 Safety Requirements	5
6.3 Security Requirements	5
6.4 Software Quality Attributes	5
6.5 Business Rules	5
7. Other Requirements	5
Appendix A: Glossary	5
Appendix B: Field Layouts	5
Appendix C: Requirement Traceability matrix	6



PES UNIVERSITY, BANGALORE

Department of Computer Science and Engineering

Revision History

Name	Date	Reason For Changes	Version
Garble: An encrypted journal	07/09/20 22		- 1.0



PES UNIVERSITY, BANGALORE

Department of Computer Science and Engineering

Introduction

Purpose

Garble is an application in which journal entries made can be encrypted to maintain confidentiality. The program will also track the user's mood based on the entries made. This document covers the requirements of the first completed model of the application.

Intended Audience

The target audience consists of journalists, writers, regular travelers, youngsters who want to preserve their privacy, professionals who want to protect their intellectual property, and entrepreneurs who seek to store and keep track of their ideas. This document ensues the product scope, description, interface requirements, analysis models, system requirements, functional and non-functional requirements.

Product Scope

This web application will let the user make journal entries. The user can log in using a login id and password. This will ensure that only authorized users can access the journal. The entries made will be encrypted using an encryption algorithm(AES). This will keep it confidential. For every entry made, a trained machine learning model will predict the mood of the user. Using this feature the user can keep track of his/her mood. A real time database will be maintained online to keep the journal entries of the user backed up, for ease of access through multiple devices. This application can be used by anyone who has internet access, and wants to keep their information confidential.

References

Project Synopsis : [ProjectSynopsis_team5](#)

Overall Description

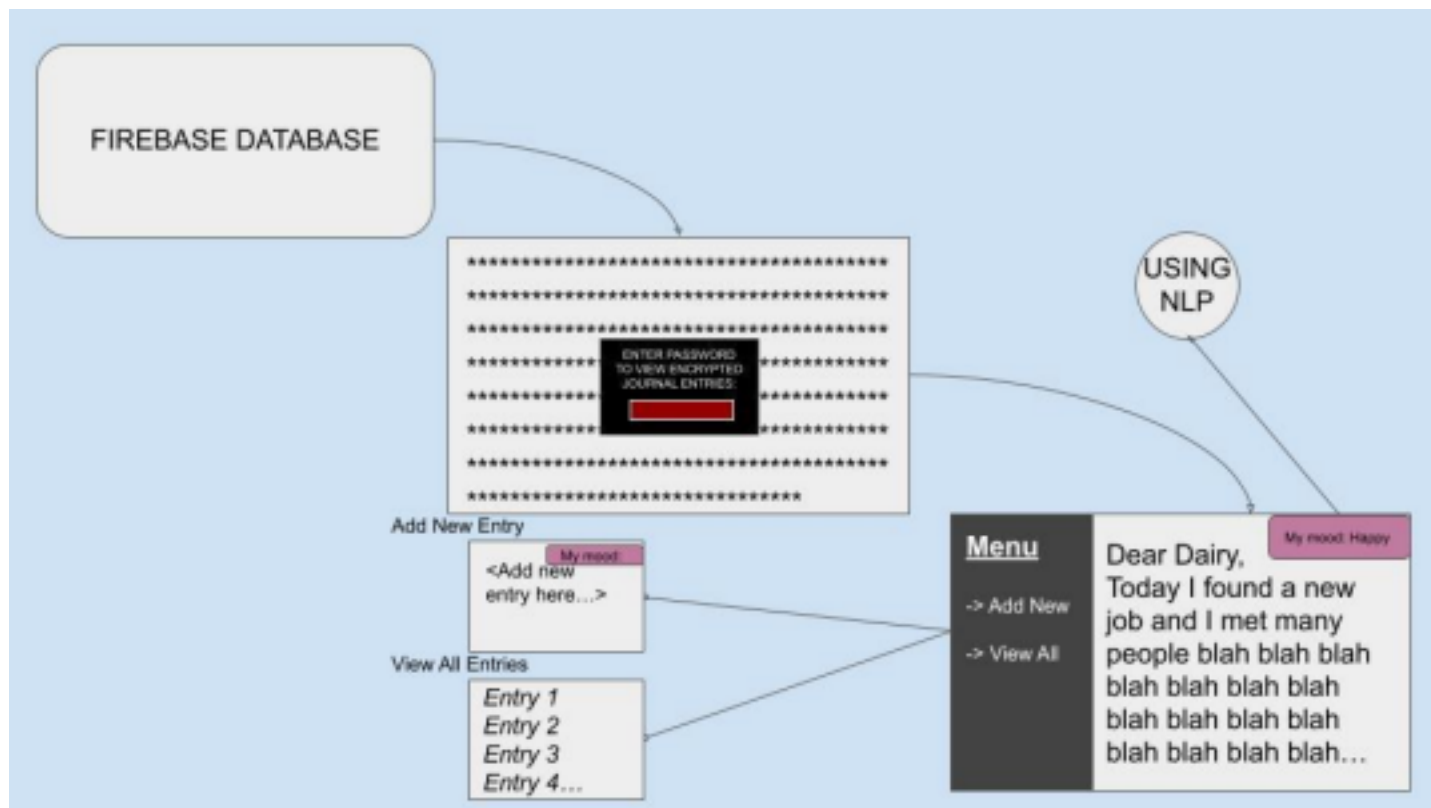
Product Perspective

This web application is a new, self-contained product. It is open-source. A realtime database will be maintained on Firebase application, and the Firebase data can be accessed by using the Firebase JS SDK.



PES UNIVERSITY, BANGALORE

Department of Computer Science and Engineering



Product Functions

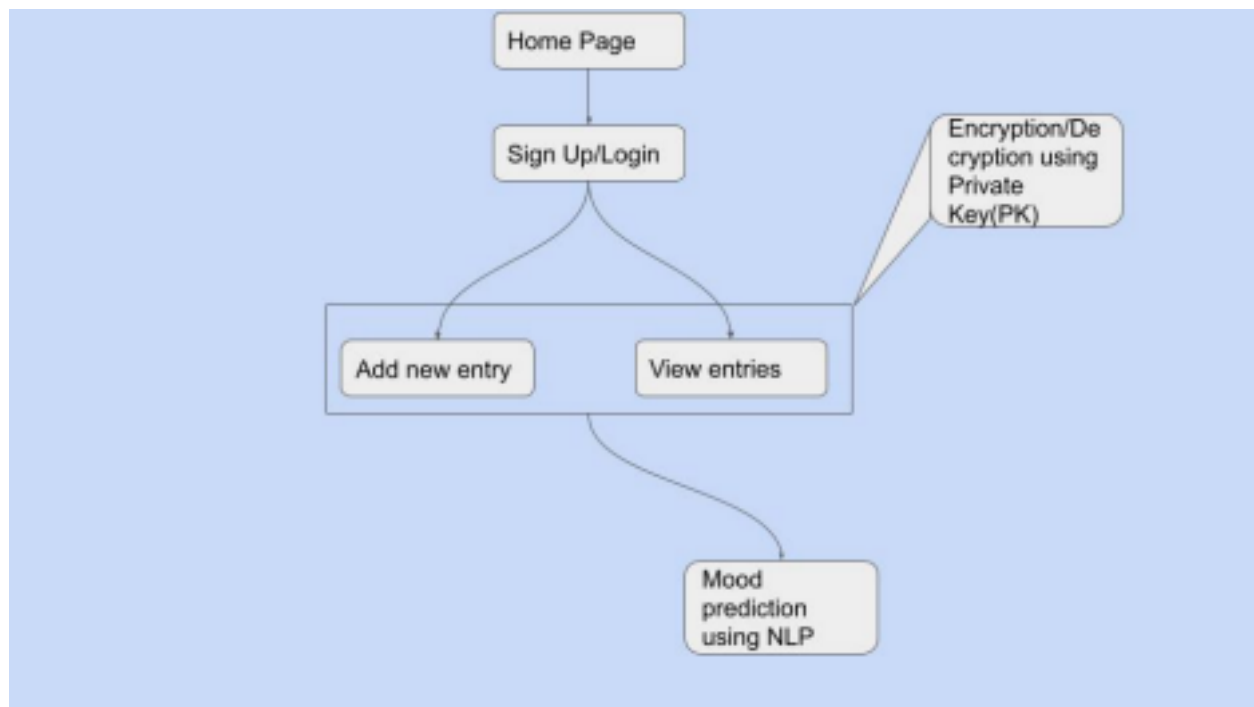
Functions:

- Logging in using a password
- Making a new journal entry
- sorting journal entries into folders
- encrypting and decrypting journal entries using a private key
- mood predicting to track mood everyday based on journal entry



PES UNIVERSITY, BANGALORE

Department of Computer Science and Engineering



User Classes and Characteristics

1. Technical Professionals- intellectuals who want to protect their information: -
 - researchers
 - professors with theses
 - scientists
 - for protection of intellectual property
2. Journalists- with internet access, can make entries anytime, anywhere: -
 - Travelers
 - bloggers
 - writers
3. Students/youth - can keep their entries private and track their daily mood: (most important class to satisfy)
 - Teenagers
 - diarists
 - individuals subjected to mental illnesses



PES UNIVERSITY, BANGALORE

Department of Computer Science and Engineering

Operating Environment

1. System requirements:
 - a. minimum 2GB memory
 - b. processor - 2 x 1,6 GHz CPU (recommended)
 - c. Windows 10 and above/MacOs/Linux 64-bit

- d. Internet access
- 2. Software requirements:
 - a. VS code editor
 - b. Firebase database
 - c. web browser - Google Chrome, Firefox, Safari, Brave, etc
 - d. Programming languages:
 - i. Javascript
 - ii. Nodejs
 - iii. Reactjs
 - iv. html
 - v. css

Design and Implementation Constraints

The goal of the web application is to be platform independent on the client side wherever possible. Therefore, the web applications will be implemented to run on the server side as much as possible.

- Customers cannot use the application without internet access.
- Customers cannot upload images, videos or any other media to the journal entries as input.
- Security - user will have to keep his/her login password and key safe. - Application limited to the web.
- The data stored in the database should be accessible by the website.

2.6 Assumptions and Dependencies

- It is assumed that the Firebase software will work correctly with the application program. - If firebase database fails to handle our traffic well, we will have to explore other database management systems.
- Application requires MERN stack and Javascript to be developed.

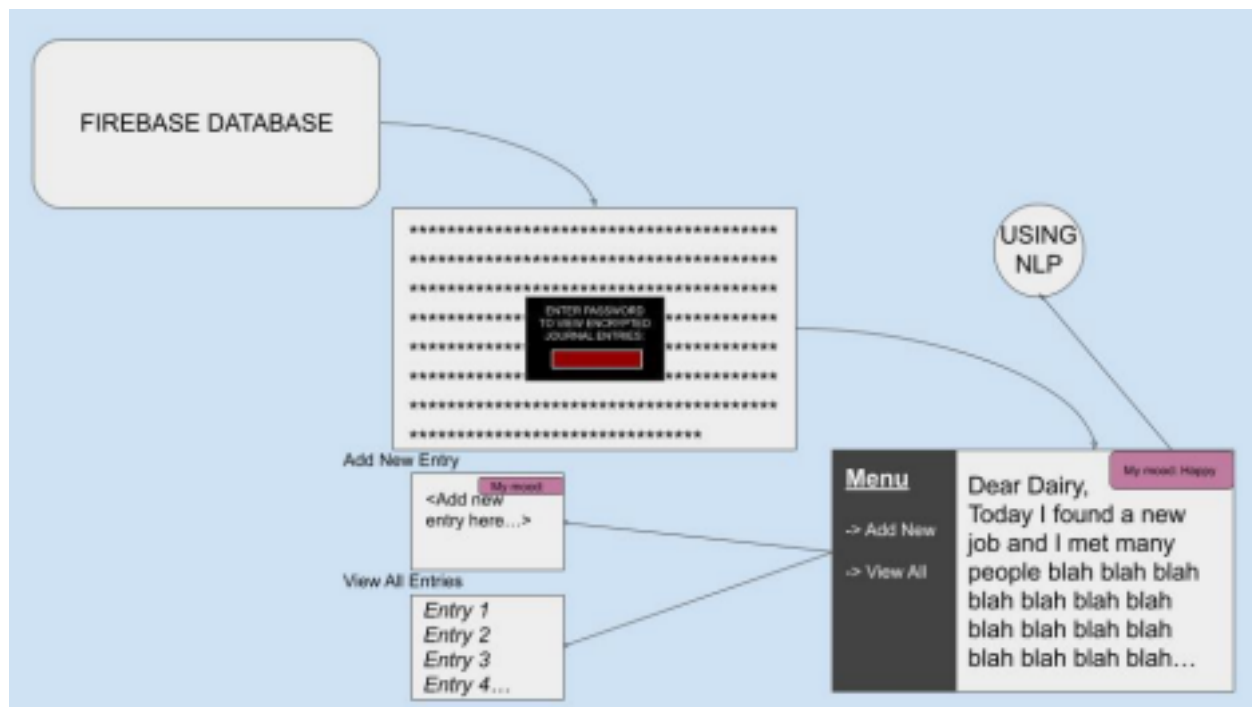


PES UNIVERSITY, BANGALORE

Department of Computer Science and Engineering

External Interface Requirements

User Interfaces



The font used in our app would be “Open Sans”. The colour palate that is used for the user interface will be mild colours like grey and pink. However, error messages and dialogue boxes will be in darker shades. We will be using css stylesheets for the UI and similar style guides will be followed. As you can see in the above image, we will have a side bar with “Menu” which will redirect us to either a “Add new entry” or “View all entry”. The “Mood” of the writer which will use NLP for the prediction will be displayed on the top right corner of every entry.

Software Interfaces

This software will be based on the firebase database(v9.0.2) and MERN Stack (Javascript(ES2015) and ReactJS(18.2.0)) for the entire development of the software. The entire software will be highly portable and will be run on all operating systems (Windows/MacOS/Linux). The data that is stored in the software would be local to the user login and protected by encryption and decryption of data using a private key.



PES UNIVERSITY, BANGALORE

Department of Computer Science and Engineering

Communications Interfaces

The software does not support any sort of communication within the application.

Analysis Models

Database - Schema Diagram

User

U_I Name Email Ph. No.

Mood(Weak Entity)

U_I E_I

EntriesD

E_ID Date Name Description

System Features

Functional requirements:

- Login authorization is done using firebase.
- Encryption of the text using AES algorithm.
- Database online to back up the data entered in Journal.

Major services :

- Maintains privacy
- Enables the users for an effortless storage of ideas
- Secure storage of data
- Information gets encrypted before being saved.

Data encryption:

5.1.1 Description and Priority

.Data encryption: *All the data entered in the journal shall be encrypted before saving it*

Priority: High

Benefit: 8

Penalty: 4

Cost: 6

Risk: 2



PES UNIVERSITY, BANGALORE

Department of Computer Science and Engineering

5.1.2 Stimulus/Response Sequences

- User enters Data / Information into a journal.
- When the user wants to save the data entered, we encrypt the data and store it in a cloud hosted database.
- Data is encrypted using the AES algorithm.

5.1.3 Functional Requirements

REQ-1: Up to date version of Python

REQ-2: Any common browser software

REQ-3: AES algorithm: An algorithm which encrypts data

Password Protection:

5.1.1 Description and Priority

Password protection: *All the data entered in the journal shall be encrypted before saving it*

Priority: High

Benefit: 9

Penalty: 3

Cost: 7

Risk: 2

5.1.2 Stimulus/Response Sequences

- Separate account is created for each user which is password protected - This enables the user to keep his data secure.

5.1.3 Functional Requirements

REQ-1: ReactJS latest version - to configure login page (seq no 1)

REQ-2: TBD

Online Database:

5.1.1 Description and Priority

Online Database: *This feature makes a backup store of all the user's data and stores it in an online real time Cloud-hosted database.*



PES UNIVERSITY, BANGALORE

Department of Computer Science and Engineering

Priority: High

Benefit: 8

Penalty: 4

Cost: 9

Risk: 4

5.1.2 Stimulus/Response Sequences

- After the data is entered in the journal, the user will save the data
- We save the data in a cloud-based database.

- It will be easy to modify the data whenever the user wants to.

5.1.3 Functional Requirements

REQ-1: Firebase: Database

REQ-2: MERN Stack: Web Development

Other Nonfunctional Requirements

Performance Requirements

- *CLOCK SPEED : 2.8 GHz*
- *MAIN MEMORY : 1GB*
- *SECONDARY MEMORY : 40GB*
- *CACHE MEMORY : 1MB*
- *TIME TO TITLE* - The amount of time between the instant a visitor requests your website and the moment your site's title shows up in their browser tab - *Connection time*: The time between a request and when a connection is established between the user's browser and your origin server is called the connection time.
- *Error rate*: The average number of problem requests compared to total requests is your error rate.

Safety Requirements

- Information and diary entries may be leaked to an unauthorized individual/organization if the key is not safely guarded.
- personal information such as email id and phone number may be leaked to unauthorized individuals/organizations if https or secure web access is not used.
- Loss of data due to failure in technology possible.



PES UNIVERSITY, BANGALORE

Department of Computer Science and Engineering

Security Requirements

- Information and diary entries may be leaked to an unauthorized individual/organization if the key is not safely guarded.
- personal information such as email id and phone number may be leaked to unauthorized individuals/organizations if https or secure web access is not used.
- User identity authentication done using password and email id or phone number - Browser must have SSH or HTTPS

Software Quality Attributes

- *This journal can be used by everyone for many purposes*
- *All the features provided are secure and easily adaptable.*
- *The Journal is flexible to be used by people from different professions. - It's an online web application, which makes it much simpler for the user to edit it whenever he wants.*
- *This is a reusable and user-friendly web-application.*

Business Rules

A person who likes to digitally secure their journal entries and at the same time monitor their mood over a period of time and gain a better understanding of themselves would use this software. There are no other roles that are supported in this software.

Other Requirements

None that are not already mentioned.

Appendix A: Glossary

- AES - Advanced Encryption Standard
- HTML - Hypertext Markup Language
- HTTPS - HyperText Transfer Protocol Secure
- MERN - MongoDB, ExpressJS, ReactJS, NodeJS application stack
- SDK - Software Development Kit
- SSH - Secure Shell

PES UNIVERSITY, BANGALORE

Department of Computer Science and Engineering

Appendix B: Field Layouts

An Excel sheet containing field layouts and properties/attributes and report requirements.

PES UNIVERSITY, BANGALORE

Department of Computer Science and Engineering

Appendix C: Requirement Traceability Matrix

Sl. No	Requirement ID	Brief Description of Requirement	Architecture Reference	Design Reference	Code File Reference	Test Case ID	System Test Case ID
1	1	Login page - to give access to user to his/her journal		https://ass - ets.justin mind.com /wp-content/uploads/2018/10/diprella_login.gif		11	21
2	2	Password protection - to authorize user		https://ass - ets.justin mind.com /wp-content/uploads/2018/10/diprella_login.gif		12	22

4 4 Online database handling -
using
Firebase SDK

<https://cdn.wikilabs.com/6pi71cGa4Nz1yhgU%2Biwp5cDD7Mv2JLuall>

HmW448Xiw %3D
14 24

PES UNIVERSITY, BANGALORE

Department of Computer Science and Engineering



PES UNIVERSITY, BANGALORE

Department of Computer Science and Engineering

Project Plan Document

Instructions:

- 1: Prepare a detailed plan for your project which comprises the below mentioned details.
- 2: Upload pdf document through the given link.
- 3: The name of the document should be your Project ID.

Things to be included as part of the project plan.

- 1: Identify the lifecycle to be followed for the execution of your project and justify why you have chosen the model.

Incremental cycle.

We want to make each deliverable one at a time, and increment the project with new functionalities.

For each increment we will be using the 'Waterfall model'.

Incremental models are easier to test and debug smaller modules.

Incremental module reduces over functionality and gives priority to essential features over additional.

We want to release a working version of the software with every module.

- 2: Identify the tools which u want to use throughout the lifecycle like planning tool, design tool, version control, development tool, bug tracking, testing tool.


























1. Planning Tool: Atlassian Jira
2. Design Tool: Canva
3. Version control tool: GitHub
4. Development tool: Visual Studio Code, Jupyter Notebook
5. Bug tracking: Sentry
6. Testing Tool: Selenium

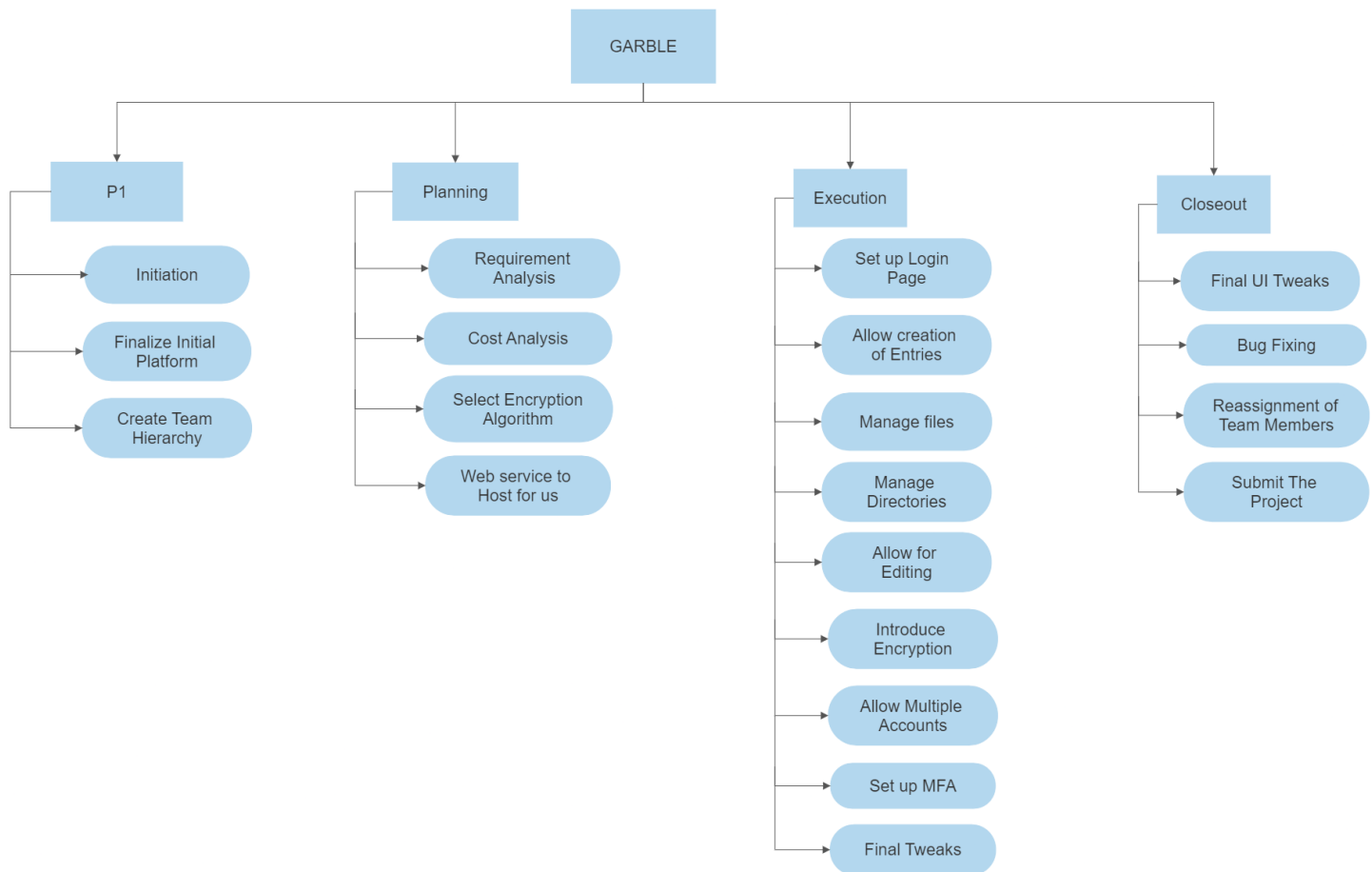
- 3: Determine all the deliverables and categorize them as reuse/build components and justify the same.

1. Login page : username and password : Build Component
2. Creating new entry : Build Component
3. Storing: Build Component
4. Viewing/opening : Build Component
5. Editing : Reuse Component : Can reuse the functionality of the "Create New Entry" feature

6. Encrypting : Build Component
7. Mood prediction: Build Component
8. Multiple account login : Reuse Component - Can reuse the functionality of the “Login” feature
9. Multi Factor Authentication: Reuse Component - Can reuse the functionality of the “Login” feature, by asking for phone number/email entry.

4: Create a WBS for the entire functionalities in detail.

▼  P1								0%
 Initiation		100%	== Me...				No	0%
 Finalize initial platform		100%	== Me...				No	0%
 Create team hierarchy		100%	== Me...				No	0%
▼  Planning								0%
 Requirement Analysis		100%	== Me...				No	0%
 Cost Analysis		100%	== Me...				No	0%
 Select Encryption Algorithm		100%	== Me...				No	0%
 Web Service to host for us		100%	== Me...				No	0%
▼  execution								0%
 Set up Login page : userna...		100%	== Me...				No	0%
 Allow creation of entries		100%	== Me...				No	0%
 Manage storing the files, fi...		100%	== Me...				No	0%
 Manage opening files dire...		100%	== Me...				No	0%
 Allow for editing of entries		100%	== Me...				No	0%
 Introduce encryption of e...		100%	== Me...				No	0%
 allow multiple accounts to...		100%	== Me...				No	0%
 Set up MFA		100%	== Me...				No	0%
 Final tweaks		100%	== Me...				No	0%
▼  Closeout								0%
 Final UI tweaks		100%	== Me...				No	0%
 Bug fixing		100%	== Me...				No	0%
 Reassignment of team me...		100%	== Me...				No	0%
 Submit the project.		100%	== Me...				No	0%



5: Do a rough estimate of effort required to accomplish each task in terms of person months.

In the next few weeks, we plan to:

- 1 week: Planning - 4 people
- 1 week : Create a design of our web application - 2 teammates.
- 1 weeks : Login using a password, for the user to access his account - 2 teammates.
- 1 week: Basic functionality to enter the journal entries . - 2 teammates
- 1 week: Editing feature - 2 teammates
- 1 week : Look into the encryption methods to be used to encrypt the journal entries - 2 teammates.
- 2 weeks: Create the encryption algorithm program to encrypt entries. - 3 teammates
- 1 week: Configuring the storage tool with the program application. - 2 teammates
- 2 weeks: Creating, training and testing the mood prediction machine learning model. - 4
- 2 weeks: Additional features (multiple login, multi factor authentication) - 4 teammates.

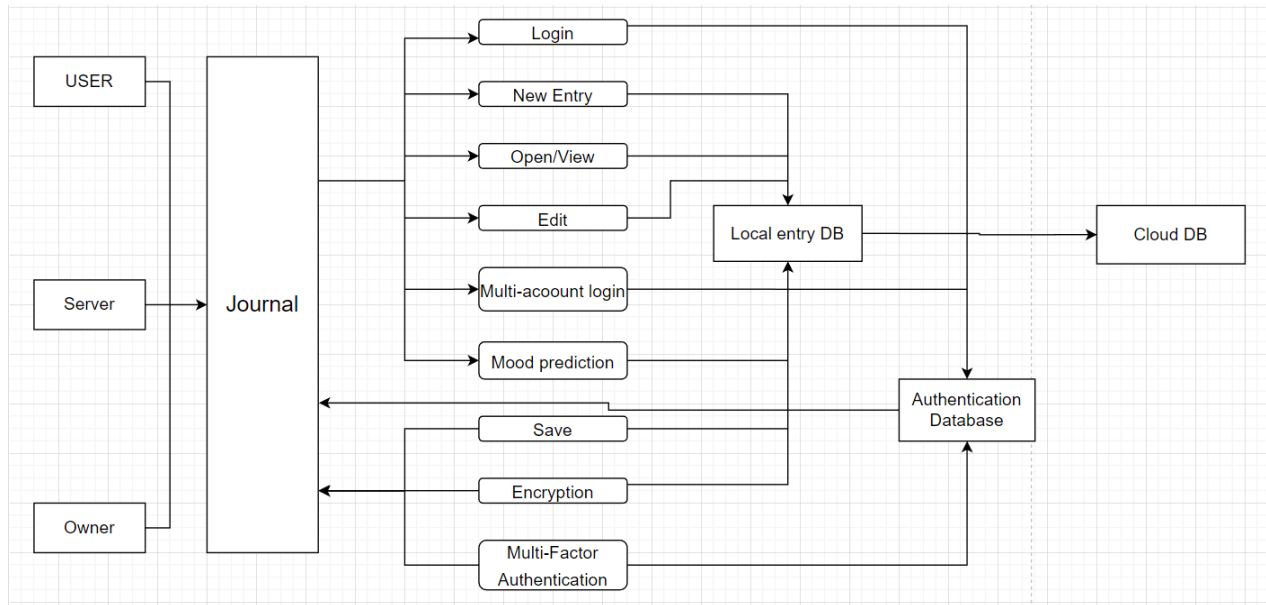
6: Create the Gantt Chart for scheduling using any tool.

[GARBLE - GANTT CHART](#)

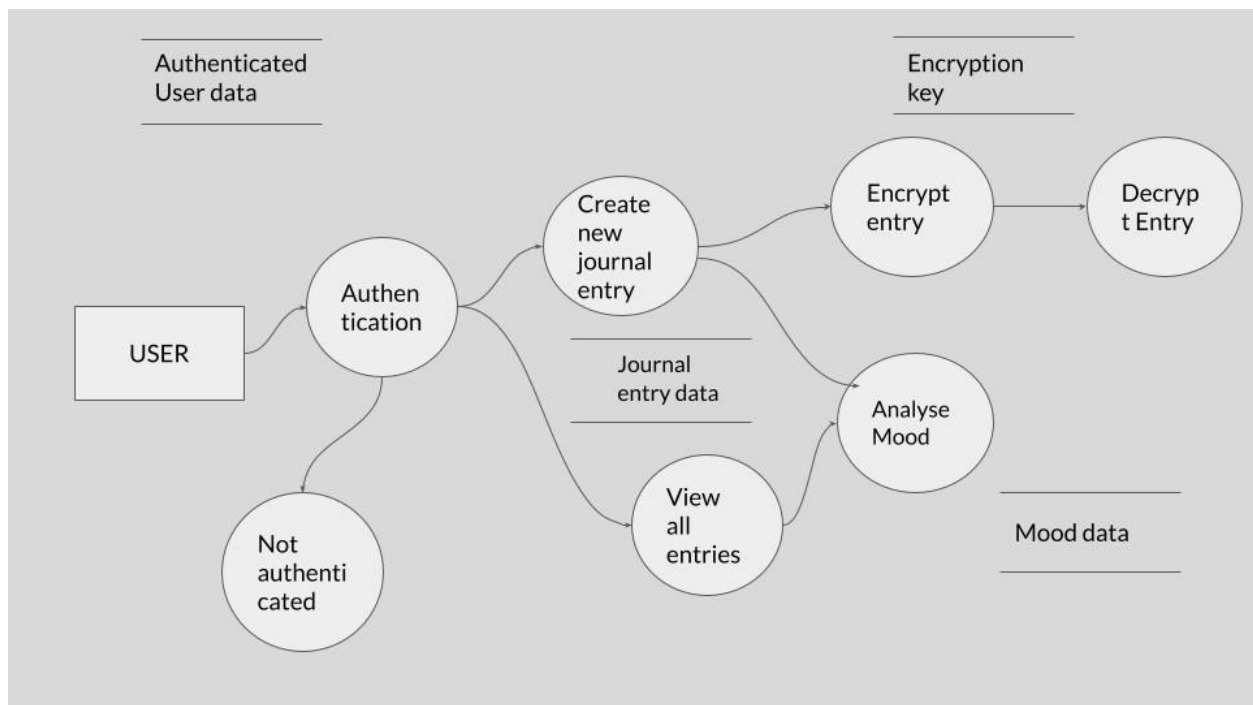
LAB-04

Propose Architectural style /design for the project selected. Design pattern for their problem statement(Architectural) identifying Quality Attributes

Architectural design:



Data Flow Diagram:



AUGUST – DECEMBER 2022 SEMESTER 5

SOFTWARE ENGINEERING LAB

TASKS

CASE STUDY 4

GARBLE: Journal Encryption

Team Members:

Keerthi R - PES1UG20CS205

Mahesh KM - PES1UG20CS242

Mahika Gupta - PES1UG20CS243

Maani JM - PES1UG20CS241

Problem Statement – 1: Unit Testing

A unit is the smallest block of code that functions individually. The first level of testing is Unit testing and this problem statement is geared towards the same.

- Discuss with your teammates and demarcate units in your code base
 - Note: discuss why the code snippet you have chosen can be classified as a unit
 - The following part of the code base is chosen as a unit for unit testing, as it is a functionality which is independent of other units of the code, and can be tested easily. The following

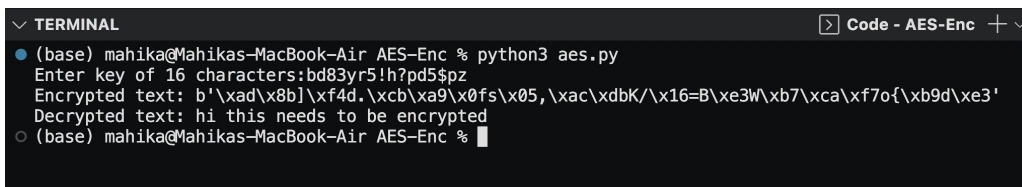
```

1  from Crypto.Cipher import AES
2
3
4  def acceptKey():
5      key_string = input("Enter key of 16 characters:")
6      if len(key_string)!=16:
7          raise Exception("Key length not 16")
8      key = bytes(key_string, 'utf-8')
9      #print(key)
10     return key
11
12 #mahika
13 # key= b'C&F)H@McQfTjWnZr'
14 def encryptText(key):
15     #key = acceptKey()
16     cipher = AES.new(key,AES.MODE_EAX)
17     nonce = cipher.nonce
18     data = "hi this needs to be encrypted".encode()
19     ciphertext = cipher.encrypt(data)
20     #print("Ciphertext:",ciphertext)
21     return ciphertext,nonce
22
23 def decrypt(key,ciphertext,nonce):
24     # key = acceptKey()
25     # ciphertext,nonce = encryptText()
26     cipher = AES.new(key,AES.MODE_EAX,nonce=nonce)
27     plaintext = cipher.decrypt(ciphertext).decode()
28     #print("Plaintext:",plaintext)
29     return plaintext
30
31
32 key = acceptKey()
33 ct,nonce = encryptText(key)
34 pt = decrypt(key,ct,nonce)
35 print("Encrypted text:", ct)
36 print("Decrypted text:", pt)

```

- Develop test cases for both valid and invalid data

- Test case 1: for valid data:



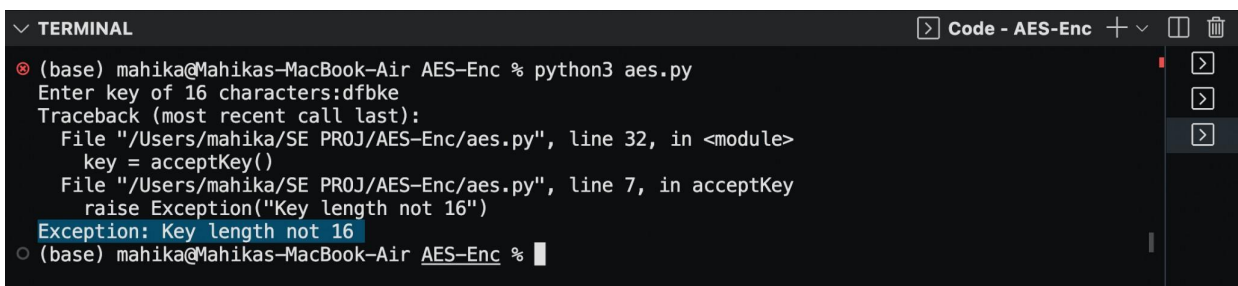
```

▼ TERMINAL
Code - AES-Enc + v
• (base) mahika@Mahikas-MacBook-Air AES-Enc % python3 aes.py
Enter key of 16 characters:bd83yr5!h?pd5$pz
Encrypted text: b'\xad\x8b]\xf4d.\xcb\xa9\x0f\x05,\xac\xdbK/\x16=B\xe3W\xb7\xca\xf7o{\xb9d\xe3'
Decrypted text: hi this needs to be encrypted
○ (base) mahika@Mahikas-MacBook-Air AES-Enc %

```

Here since the constraint on the key input is to enter a key with 16 characters, the test case passes as a 16 byte key is entered. Test case is passed.

- Test case 2: for invalid data:



```

▼ TERMINAL
Code - AES-Enc + v
ⓧ (base) mahika@Mahikas-MacBook-Air AES-Enc % python3 aes.py
Enter key of 16 characters:dfbke
Traceback (most recent call last):
  File "/Users/mahika/SE PROJ/AES-Enc/aes.py", line 32, in <module>
    key = acceptKey()
  File "/Users/mahika/SE PROJ/AES-Enc/aes.py", line 7, in acceptKey
    raise Exception("Key length not 16")
Exception: Key length not 16
○ (base) mahika@Mahikas-MacBook-Air AES-Enc %

```

Here we see that the condition for the key size being 16 characters is not satisfied. The value entered is invalid and an exception is raised. Test case is passed.

- Ideate how you could further modularize larger blocks of code into compact units with your teammates
- This unit can be further isolated for rigorous testing by dividing it into 2 units of one function each:
 - One unit for the function `encryptText(key)`:
 - The other unit for the function `decrypt(key,ciphertext,nonce)`

Problem Statement – 2: Dynamic Testing

Dynamic testing involves execution of your code to analyse errors found during execution. Some common techniques are Boundary Value Analysis and Mutation Testing.

Problem Statement – 2.a: Boundary Value Analysis

When it comes to finding errors in your code base, they are often found at locations where a condition is being tested. Due to this, developers often use Boundary Value tests to reduce defect density.

- How would you define a boundary test?
 - Note: Simple relational conditions are a basic example

Boundary testing is a black-box testing technique used to identify the errors at the boundary or the extreme ends of the model.

- Build your boundary test cases and execute them

In the unit code that we are testing, the input must have exactly 16 characters. Hence, in this case the boundary value considered will be 16.

- Test 1: entering boundary value as input:

On passing an input of 16 characters:


```
▼ TERMINAL Code - AES-Enc +
● (base) mahika@Mahikas-MacBook-Air AES-Enc % python3 aes.py
Enter key of 16 characters:bd83yr5!h?pd5$pz
Encrypted text: b'\xad\x8b]\xf4d.\xcb\xa9\x0fs\x05,\xac\xdbK/\x16=B\xe3W\xb7\xca\x7f0{\xb9d\xe3'
Decrypted text: hi this needs to be encrypted
○ (base) mahika@Mahikas-MacBook-Air AES-Enc %
```

- Test 2: entering value less than boundary value:

On passing a value with 15 characters:

```
⊗ (base) mahika@Mahikas-MacBook-Air AES-Enc % python3 aes.py
Enter key of 16 characters:hsgey378j!heorp
Traceback (most recent call last):
  File "/Users/mahika/SE PROJ/AES-Enc/aes.py", line 32, in <module>
    key = acceptKey()
  File "/Users/mahika/SE PROJ/AES-Enc/aes.py", line 7, in acceptKey
    raise Exception("Key length not 16")
Exception: Key length not 16
○ (base) mahika@Mahikas-MacBook-Air AES-Enc %
```

- Test 3: entering value greater than boundary value:

On passing a value of 17 characters:

```
⊗ (base) mahika@Mahikas-MacBook-Air AES-Enc % python3 aes.py
Enter key of 16 characters:hu8uie0?j2#ywh0ks
Traceback (most recent call last):
  File "/Users/mahika/SE PROJ/AES-Enc/aes.py", line 32, in <module>
    key = acceptKey()
  File "/Users/mahika/SE PROJ/AES-Enc/aes.py", line 7, in acceptKey
    raise Exception("Key length not 16")
Exception: Key length not 16
```

Problem Statement – 2.b: Mutation Testing

- Using your isolated units from the first problem statement, ideate with your teammates on how to mutate the code

The code can be mutated such that key of length greater than 15 is accepted (still throws an error due to python packages)

Original code:

```
def acceptKey():
    key_string = input("Enter key of 16 characters:")
    if len(key_string)!=16:
        raise Exception("Key length not 16")
    key = bytes(key_string, 'utf-8')
    #print(key)
    return key
```

Mutated code:

```
def acceptKey():
    key_string = input("Enter key of 16 characters:")
    if len(key_string)>15:
        raise Exception("Key length not 16")
    key = bytes(key_string, 'utf-8')
    #print(key)
    return key
```

Output with error:

```
Exception: No data entered
⊗ (base) mahika@Mahikas-MacBook-Air AES-Enc % python3 caseStudy.py
Enter key of 16 characters:yueripshdjfkith76
Traceback (most recent call last):
  File "/Users/mahika/SE PROJ/AES-Enc/caseStudy.py", line 35, in <module>
    key = acceptKey()
  File "/Users/mahika/SE PROJ/AES-Enc/caseStudy.py", line 7, in acceptKey
    raise Exception("Key length not 16")
Exception: Key length not 16
○ (base) mahika@Mahikas-MacBook-Air AES-Enc %
```

- Develop at least 3 mutants of the functioning code and test all 4 code bases using the test case from the first problem statement

Mutants:

(i) Giving 17 characteristics as password instead of 16:

```
PS D:\5th sem\SE\Case study-4> python -u "d:\5th sem\SE\Case study-4\original.py"
Enter key of 16 characters:asdfghjklmnbvcxzq
Traceback (most recent call last):
  File "d:\5th sem\SE\Case study-4\original.py", line 27, in <module>
    key = acceptKey()
  File "d:\5th sem\SE\Case study-4\original.py", line 6, in acceptKey
    raise Exception("Key length not 16")
Exception: Key length not 16
```

(ii) Input text is null:

```
PS D:\5th sem\SE\Case study-4> python -u "d:\5th sem\SE\Case study-4\original.py"
Enter key of 16 characters:asdfghjklmnbvcxz
Encrypted text: b''
Decrypted text:
PS D:\5th sem\SE\Case study-4> █
```

(iii) Starting argument without an encoding

```
PS D:\5th sem\SE\Case study-4> python -u "d:\5th sem\SE\Case study-4\original.py"
Enter key of 16 characters:asdfghjklmnbvcxz
Traceback (most recent call last):
  File "d:\5th sem\SE\Case study-4\original.py", line 27, in <module>
    key = acceptKey()
  File "d:\5th sem\SE\Case study-4\original.py", line 7, in acceptKey
    key = bytes(key_string)
TypeError: string argument without an encoding
PS D:\5th sem\SE\Case study-4> █
```

Problem Statement – 3: Static Testing

Static testing involves validating your code without any execution. Under this problem statement, you will be expected to analyze and calculate the cyclomatic complexity of your code.

- Using the unit you selected in the first problem statement as an example, develop the control flow graph of your problem statement.

```

1) def acceptKey():
2)     key_string = input("Enter key of 16 characters:")
3)     if len(key_string) != 16:
4)         raise Exception("Key length not 16")
5)     key = bytes(key_string, 'utf-8')

7) def encrypt(key):

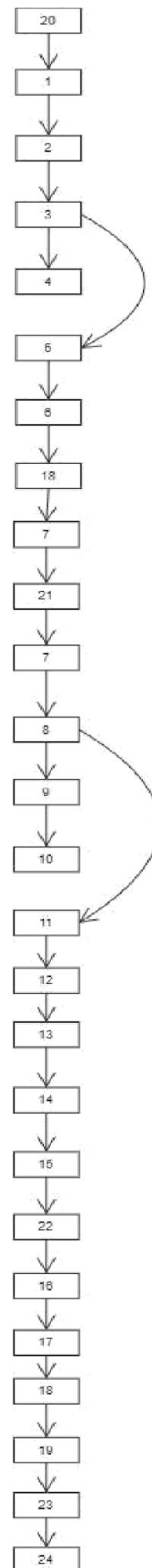
11) cipher = AES.new(key, AES.MODE_EAX)
12) nonce = cipher.nonce
13) data = "hi this needs to be encrypted".encode()
14) ciphertext, nonce = cipher.encrypt(data)
15) return ciphertext, nonce

17) cipher = AES.new(key, AES.MODE_EAX, nonce=nonce)
18) plaintext = cipher.decrypt(ciphertext).decode()
19) return plaintext

data = None

21) ct, nonce = encrypt(key, data)
22) pt = decrypt(key, ct, nonce)

```



- Using the Control flow graph, calculate the cyclomatic complexity of your code.

Cyclomatic complexity = $N - M + 2P$

{where, N=no of Nodes, M=No of edges, P=No of exit points}

Cyclomatic complexity = $6(N = 24, M = 24, P = 3)$

- Using the cyclomatic complexity as an indicator, Ideate and code your unit again to reduce complexity

Cyclomatic complexity = $21 - 21 + 2(2) = 4$

{After removing if statement in 'encrypttext' function}

Problem Statement – 4: Acceptance Testing

Assume your neighboring team is the client for your code. Give them an idea of what your product is and the software requirements for the product.

- Exchange your code base and test each others projects to see if it meets user requirements
- If you identify a bug in the project you are testing, inform the opposing team of the bug
- As a team, based in clients experience, ideate modifications to the existing project that could improve client experience

Answer:

Details of the neighboring team -

Team name: Clap for krishna

Members: Krishna Bajaj,Karthik Nair,Kumar Abhimanyu,Kingsuk Karmakar

Project Name: Garbage disposal system

Project Description:

1. Each garbage can will have a working ultrasonic sensor whose data is sent to an acting server(node mcu)The ultrasonic sensor will be used to check garbage levels(between empty to full)
2. The node mcu which acts as the server will send this data to a firebase database(note : data is real time)

3.A website is present which allows the admin to view this information for all the connected garbage cans.

4.He/she can make a decision when a certain garbage can gets filled and request a worker to take out the garbage at that time.

Our review of their code:

User requirements met are:

1. Connecting to the backend with react : Connection justified with the required keys and IDs
2. Connecting to the firebase backend via node MCU(real time database): Any updating value which happens in that instance gets updated onto the firebase database.
3. The buttons used to input data works well : Data that is input can be seen on the backend as well as the frontend (when the backend data is displayed as a table.)
4. The ultrasonic data is constricted to be within range.

Modification to improve user experience:

The web app needs to be modified with the addition of css. The interface(html) must be re-engineered.

Problem Statement – 5: Maintenance Activities

Once a product is completed, it is handed off to a service based company to ensure all maintenance activities are performed without the added expenditure of skilled developers. However, a few tasks are performed by the maintenance team to gauge the product better. In this problem statement, you will be asked to experiment with your code.

- Exchange code bases with your neighboring teams and reverse engineer a block of code in order to understand it's functionality

```

while (WiFi.status() != WL_CONNECTED) {
    Serial.print(".");
    delay(500);
}

Serial.println();
Serial.print("Connected to ");
Serial.println(WIFI_SSID);
Serial.print("IP Address is : ");
Serial.println(WiFi.localIP()); //print local IP address
Firebase.begin(FIREBASE_HOST, FIREBASE_AUTH); // connect to firebase

Firebase.reconnectWiFi(true);
delay(1000);

void loop() {

    /* Firebase Error Handling And Writing Data At Specified Path***** */

    if (Firebase.setInt(firebaseData, "/Workers/1/Percentage", val)) { // On successful Write operation, function returns 1
        Serial.println("Value Uploaded Successfully");
        Serial.print("Val = ");
        Serial.println(val);
        Serial.println("\n");

        val++;
        delay(1000);
    }

    .se {
        Serial.println(firebaseData.errorReason());
    }
}

```

Reverse-engineering the code :

The Connection to firebase needs to be delayed by a small time interval.

This is done so that there is enough time for the program to read the value at a certain time and display it onto the front end without a new value coming and taking its place.

The main reason is just to avoid constantly updating values which may be quite tedious for the admin who's using the website to watch

- After understanding the code block, Re-Engineer the code
 - Ideate how to refactor the code and the portion of the code base you would have to change
 - The web app needs to be modified with the addition of css.
 - The interface(html) should be re-engineered.
 - Discuss how the new changes would impact the time and space complexity of the project during execution
 - As its a real time system time and space complexity cannot be changed but we can change the user interface to make it easier and more convenient for clients.

- After Reverse Engineering and Re-Engineering the code, perform acceptance testing between the

Enter Block A6

Enter Percentage 98

Enter Worker 6

Add DataUpdate DataDelete Data

#	Block	Percentage	Worker
1	A1	22	1
2	A6	98	6

teams

The values from the boxes got added to the database real time, and hence we can say that the test case has passed.