

VILLAGE DEVELOPMENT

SOFTWARE ENGINEERING PROJECT



B.Sc. (Hons.) Computer Science (II Year)

Semester IV

PGDDAV COLLEGE (M)

UNIVERSITY OF DELHI

SUBMITTED BY:

VINEET 3953

KOMAL 3933

ACKNOWLEDGEMENT

On the successful completion of our project, we would like to express our sincere gratitude to everyone who helped us in the completion of the project.

Apart from the efforts of the team, the success of any project depends largely on the encouragement and guidelines of many others. We take this opportunity to express our gratitude to the people who have been instrumental in the successful completion of this project.

The completion of any inter-disciplinary project depends upon cooperation, co-ordination and combined efforts of several sources of knowledge.

We are eternally thankful to our subject teacher **Ms. Aparna Dutt**, for her valuable advice and direction, under whom we executed this project. Her constant guidance and willingness to share his vast knowledge made us understand this project and its manifestations in great depths and helped us to complete this project. This project would not have been possible without her guidance. Although there may be many who remain unacknowledged in this humble note of gratitude, there are none who remain unappreciated.

TABLE OF CONTENTS

1. Problem Statement
2. Software Process Model
3. Software Requirement Specification (SRS)
4. Use Case Diagram
5. Data Flow Diagram (DFDs)
6. Entity Relationship Diagram (ERD)
7. Database Management System Executed in SQL
8. Timeline Chart
9. Graphical User Interface (GUI)
10. Cyclomatic Complexity
11. Pseudocode
12. Risk Management
13. System Testing
14. Test Cases
15. References

CERTIFICATE

This is to certify that software engineering project report entitled "**Village Development**" is the work carried out by **Vineet, Komal** of B.Sc. (Hons.) Computer Science 4th semester, PGDAV College, University of Delhi.

Teacher in Charge

Principal

DECLARATION

I hereby declare that this Project Report titled “Mommy’s World” submitted to the Department of Computer Science, PGDAV College (University Of Delhi) is a record of original work done by the team under the guidance of **Dr. Aparna Datt**. The information and data given in the report is authentic to the best of the team knowledge. This Project Report is not submitted to any other university or institution for the award of any degree, diploma or fellowship or published any time before.

PROBLEM STATEMENT

Villagers face many problems because they don't know what the village panchayat does and they don't have any data in writing. At present, there is no platform /software to connect villagers to their gram panchayat all over India. A platform/portal where villagers could share their problems suggests new ideas and villagers should be aware of the current development status of the village. They should also know how much budget is being spent for which work. It tracks the growth of Infrastructure development. It shares the details of the awareness program organized by the Panchayat. It also has a complaint box - where villagers can share their problems and it gathers the details of all the meetings that will be held by the Gram Panchayat. It reduces the corruption rate and helps in the maintenance and development of the village. All the manual work will be converted into computerized, so the load of employees will decrease. The database will be stored in the computer rather than in the register/manually. It is easy to store information in the database and quickly accessible also.

CURRENT SYSTEM:

The current system is a manual one in every panchayat displaying the list of available schemes provided by the Government. It invites applications from eligible users based on certain criteria. All the applications are verified at the panchayat level and then approved applications are sent to the District level for approval. All the approved users avail the benefits of the scheme.

The following are the disadvantages of the current system:

- It is not secure to maintain important information manually
- More manual hours need to generate the required reports
- It is tedious to manage historical data which needs much space to keep
- Data is not in sync in the case of a manual system.

PROPOSED SYSTEM:

The proposed system is a software application that avoids more manual hours that need to be spent in record keeping and generating reports. This application keeps the data in a centralized way which is available to all the users simultaneously. It is very easy to manage historical data in the database. No specific training is required for the employees to use this application. They can easily use the tool that decreases manual hours spent on normal things and hence increases the performance.

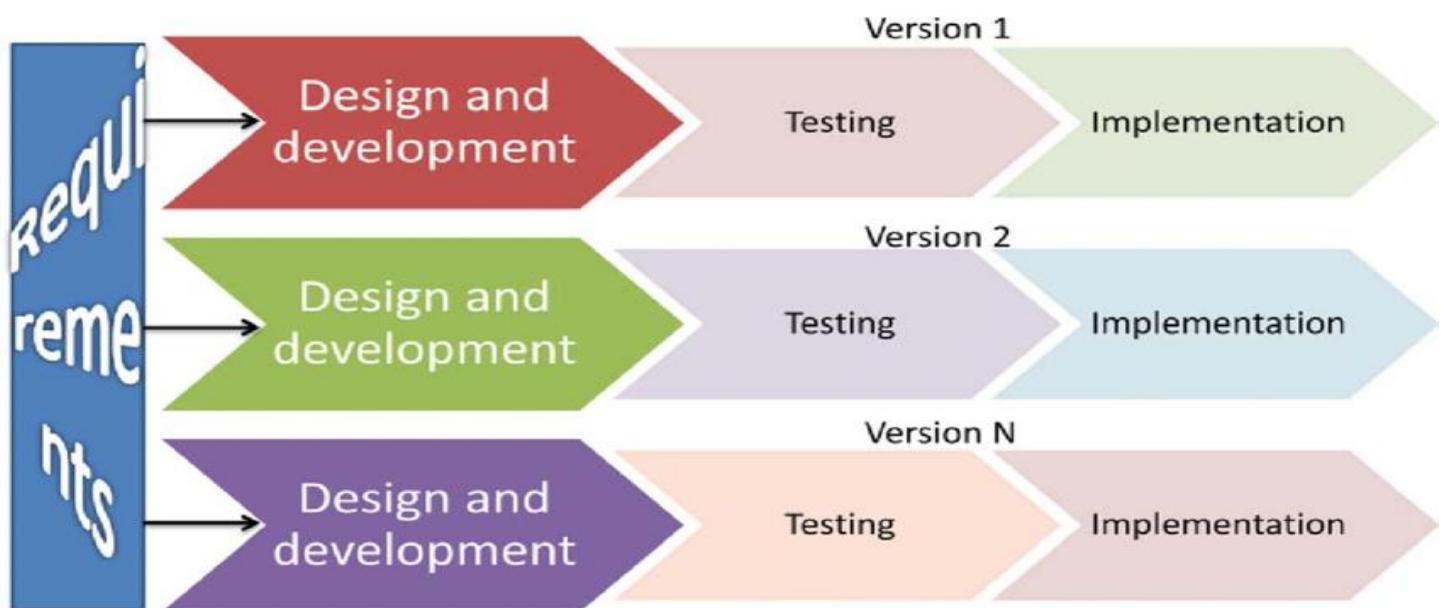
The following are the advantages of the proposed system:

- Easy to track the status of applications at any level at any point in time
- Can generate required reports easily
- Easy to manage historical data in a secure manner
- A centralized database helps in avoiding conflicts
- Easy to use GUI that does not require specific training.

SOFTWARE PROCESS MODEL

INCREMENTAL MODEL

The incremental model combines elements of the waterfall model applied in an iterative fashion. When an incremental model is used, the first increment is often a core product. That is, basic requirements are addressed, but many supplementary features remain undelivered. The core product is used by the customer. As a result of use and/or evaluation, a plan is developed for the next increment. The plan addresses the modification of the core product to better meet the needs of the customer and the delivery of each increment until the complete product is produced.



ADVANTAGES OF THE INCREMENTAL MODEL

- Customer feedback is received after the delivery of each module.
- The risk of requirement changes is reduced.
- More flexible.
- Easy to test and debug.
- Give quick results

WHY DID WE USE THE INCREMENTAL MODEL?

The reason for using this model is that the concept of our software is vast and there are a lot of modules that are required to be built. The use of the increment model will allow us to analyze our project after the completion of each module. Thus, we can change our requirements at the start of each module. Moreover after our first increment; we can launch our core product for the users. After use, evaluation, and feedback from the users, we can modify our software and take it to a more advanced level.

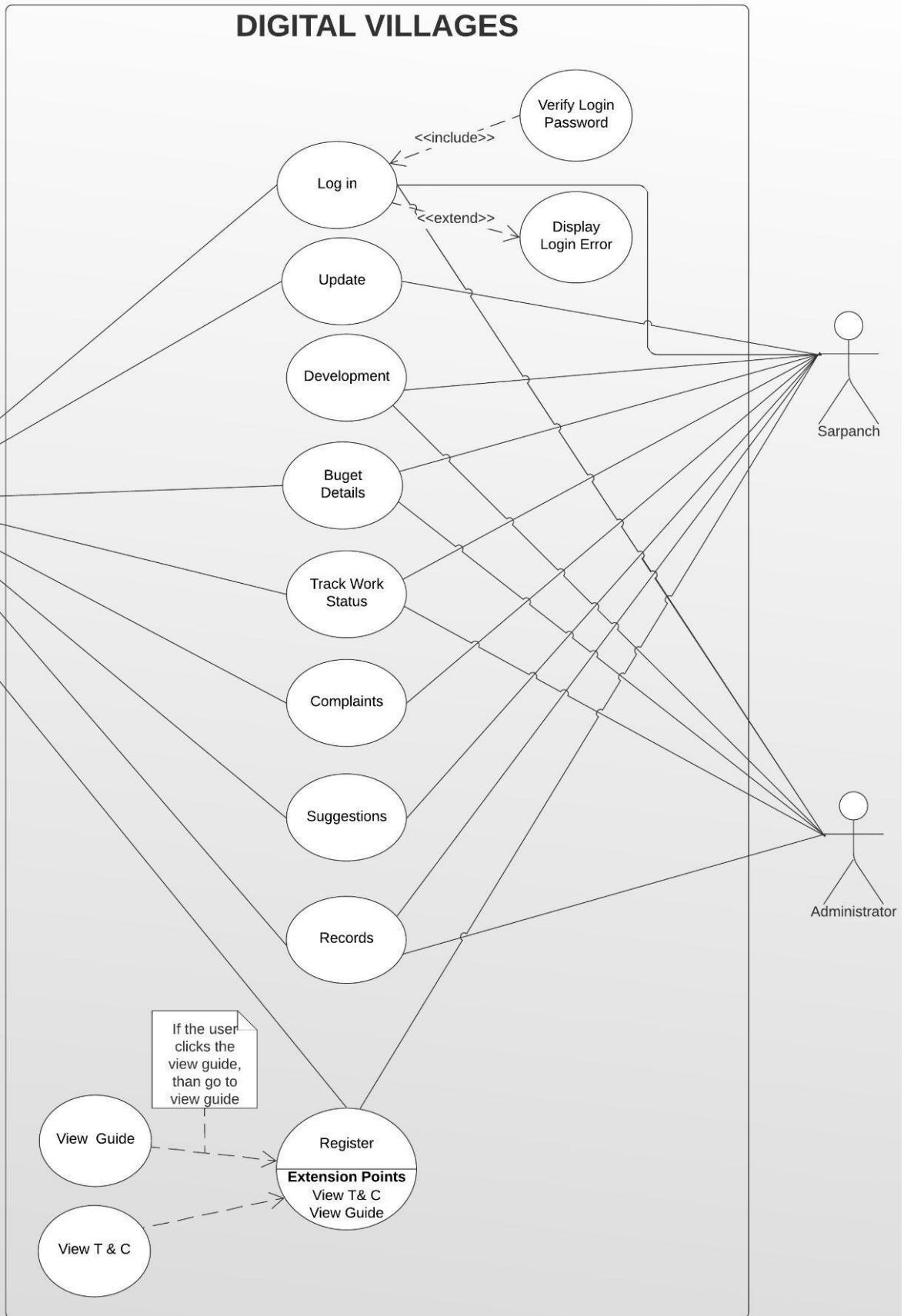
SOFTWARE REQUIREMENT SPECIFICATION

It is the means of translating ideas [inputs in the minds of people] into a formal document [output] which consists of a set of formally specified requirements that are complete and consistent.

Functional Requirements

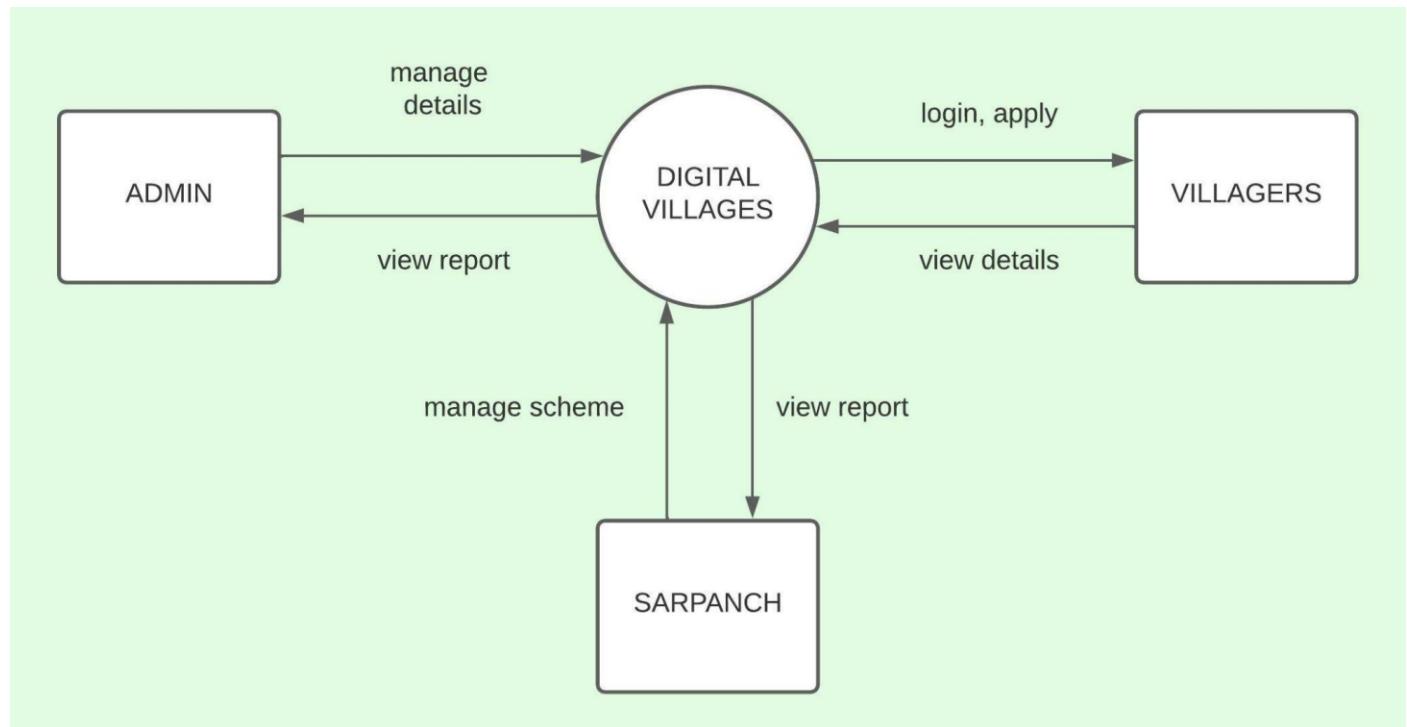
- **Register Module:** In this module, the villagers and sarpanch registered themselves.
- **Login Module:** This module describes how villagers, sarpanch, and admin log into the system to access different functionalities.
- **Development Module:** This model will be accessed by Sarpanch and Admin. This model will give a complete flow of construction work and help to maintain budget
- **Budget Details Module:** This Module is accessed by the Sarpanch and Villagers to keep track of how the budget is being spent.
- **Complaint Module:** in this module, villagers will share their problems regarding maintenance, infrastructure, and other problems so that the sarpanch gets the problem and solves them quickly.
- **Suggestion Module:** This module allows villagers to send any type of suggestion (eg: feedback, idea, bugs in the portal, etc.) which helps for the better development of the village, and villagers can also connect to the development.
- **Track work status Module:** this module will be accessed by the villagers, sarpanch, and admin, here the user can track the progress of work like how much work will be done, which currently running, and which work is completed. This module will generate a daily report to keep track of work.
- **Records:** This model will be accessed by the Villagers, Sarpanch and Admin, it will have the record store of all the village sarpanches and engineers, so the villagers have information about their panchayat and workers.
- **Update:** This module will be accessed by the Sarpanch and Villagers, here Sarpanch will update information, and notice and villagers can view updated records.

USE CASE DIAGRAM

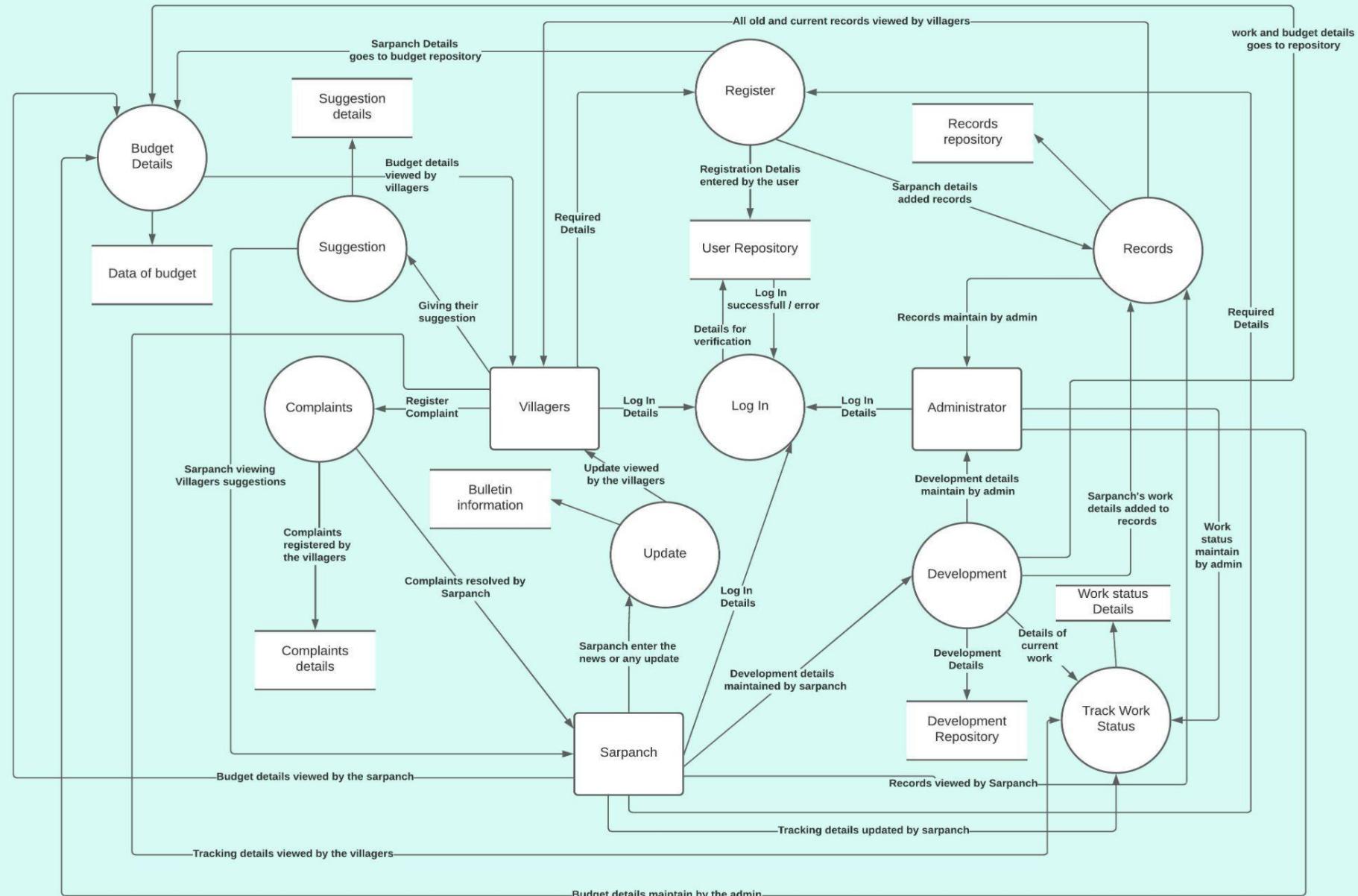


DATA FLOW DIAGRAM

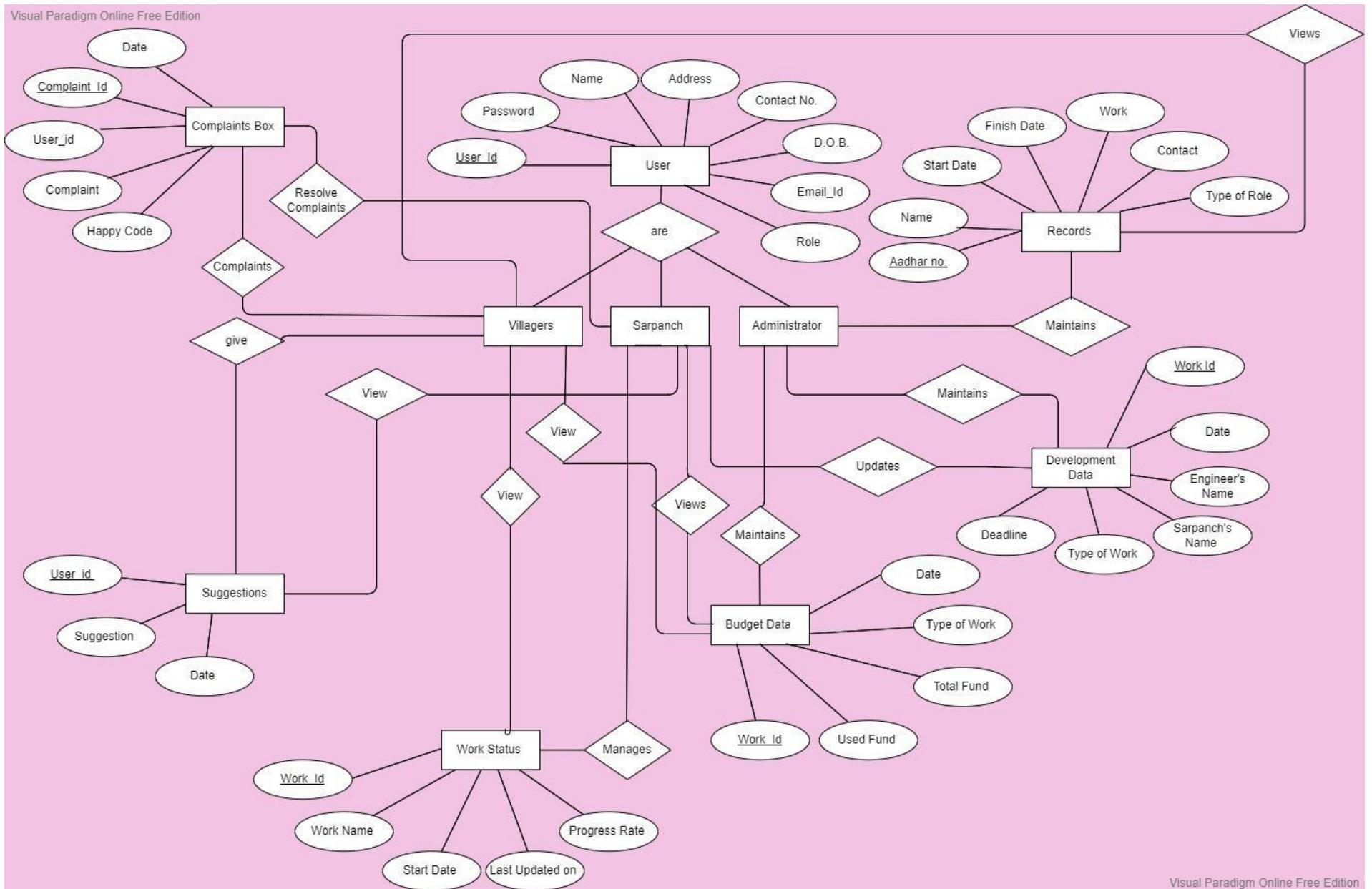
1. LEVEL - 0



LEVEL - 1



ENTITY-RELATIONSHIP DIAGRAM



RELATIONAL DATABASE MANAGEMENT SYSTEM EXECUTED IN SQL

```
mysql> show tables;
+-----+
| Tables_in_village_development |
+-----+
| budget_data
| complaint_box
| development
| records
| suggestion
| user
| work_status
+-----+
```

```
mysql> select * from user;
```

User_Id	Name	Email_Id	Password	Date of Birth	Contact No.	Address	Role
SA10	Pratap Singh	Pratap123@gmail.com	Pratap#123	1965-01-20	9685874123	Alawalpur	Sarpanch
SA11	Bharat Pal	Bharat123@gmail.com	Bharat#123	1970-03-19	8574691230	Katesra	Sarpanch
SA12	Rajesh Yadav	Rajesh123@gmail.com	Rajesh#123	1975-01-30	7854693210	Prithla	Sarpanch
VI10	Mahipal	Mahipal123@gmail.com	Mahi#123	1996-12-10	9876541230	Alawalpur	Villager
VI11	Himanshu	Himanshu123@gmail.com	Him@123	1995-12-11	8527603835	Katesra	Villager
VI12	Krishana	Krish123@gmail.com	Krish#123	1980-11-09	9647851230	Katesra	Villager
VI13	Gajender	Gajender@gmail.com	Gaj#123	1975-03-15	9587461230	Alawalpur	Villager
VI14	Sachin	Sachin123@gmail.com	Sachin#113	1996-05-16	9356287410	Prithla	Villager
VI15	Rahul	Rahul123@gmail.com	Rahul@123	2000-12-11	8596741230	Prithla	Villager

```
mysql> select * from development;
```

Work_Id	Date	Sarpanch's Name	Engineer's Name	Type of Work	Deadline
W101	2018-01-15	Pratap Singh	Rohan	Road Construction	2018-05-16
W102	2019-05-16	Bharat Pal	Mohan Singh	Electricity wiring	2019-10-17
W103	2017-12-10	Rajesh Yadav	Jitesh	Water Supply Pipeline	2018-03-11
W104	2015-01-12	Pratap Singh	Sohan	School Development	2015-12-20
W105	2020-06-12	Bharat Pal	Pushpender	Park development	2020-12-30
W106	2018-05-11	Rajesh Yadav	Vidhi Dixit	Road Construction	2018-11-05

```
mysql> select * from budget_data;
```

Work_Id	Type of Work	Date	Total Fund	Used Fund
W101	Road Construction	2017-09-16	9000000	8500000
W102	Electricity wiring	2018-12-31	5000000	5000000
W103	Water Supply Pipeline	2017-12-10	2000000	1800000
W104	School Development	2015-01-12	20000000	19000000
W105	Park development	2020-02-11	1000000	1000000
W106	Road Construction	2017-12-11	8000000	7800000

```
mysql> select * from work_status;
```

Work_Id	Work Name	Start Date	Last Updated on	Progress Rate (%)
W101	Road Construction	2018-01-15	2018-03-11	80
W102	Electricity wiring	2019-05-16	2019-10-13	100
W103	Water Supply Pipeline	2017-12-10	2018-01-11	40
W104	School Development	2015-01-12	2015-12-20	100
W105	Park development	2020-06-12	2020-10-11	75
W106	Road Construction	2018-05-11	2018-09-06	90

```
mysql> select * from complaint_box;
+-----+-----+-----+-----+-----+
| Complaint_Id | User_Id | Date      | Complaint                                | Happy Code |
+-----+-----+-----+-----+
| C101          | VI10    | 2011-05-15 | Related to Drinking Water                | 100AV      |
| C102          | VI11    | 2017-05-11 | Sewage related problem in front of my house | 15DF      |
| C103          | VI14    | 2019-05-13 | Electricity wire breakage                | AB123     |
+-----+-----+-----+-----+
```

```
mysql> select * from suggestion;
+-----+-----+-----+
| User_Id | Date      | Suggestion                         |
+-----+-----+-----+
| VI12    | 2019-01-12 | Provide security at night         |
| VI14    | 2016-04-11 | Provide transport to nearest Market |
| VI15    | 2011-06-30 | Provide infant Vaccinations       |
+-----+-----+-----+
```

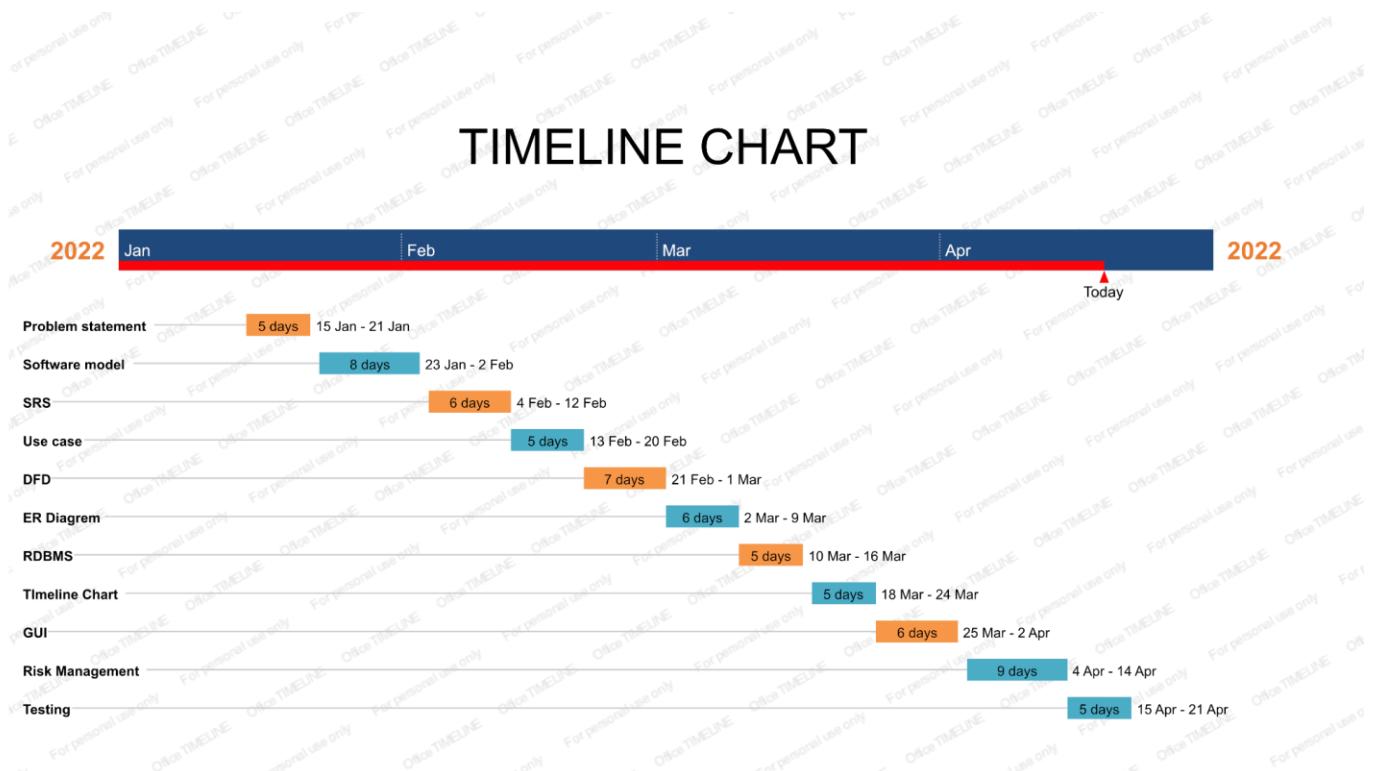
```
mysql> select * from records;
```

Aadhar Number	Name	Start Date	Finish Date	Work	Contact	Role Type
123456789874	Vidhi Dixit	2018-05-11	2018-11-05	Road Construction	9874561230	Engineer
169874123012	Pratap Singh	2015-01-15	2020-01-16	Road Construction, Electricity Wiring	7584962310	Sarpanch
456213987012	Bharat Pal	2011-04-12	2016-03-12	Water Pipeline, Park Development	8574963210	Sarpanch
987654123012	Pushpender	2020-06-12	2020-12-30	Park Development	8574963210	Engineer

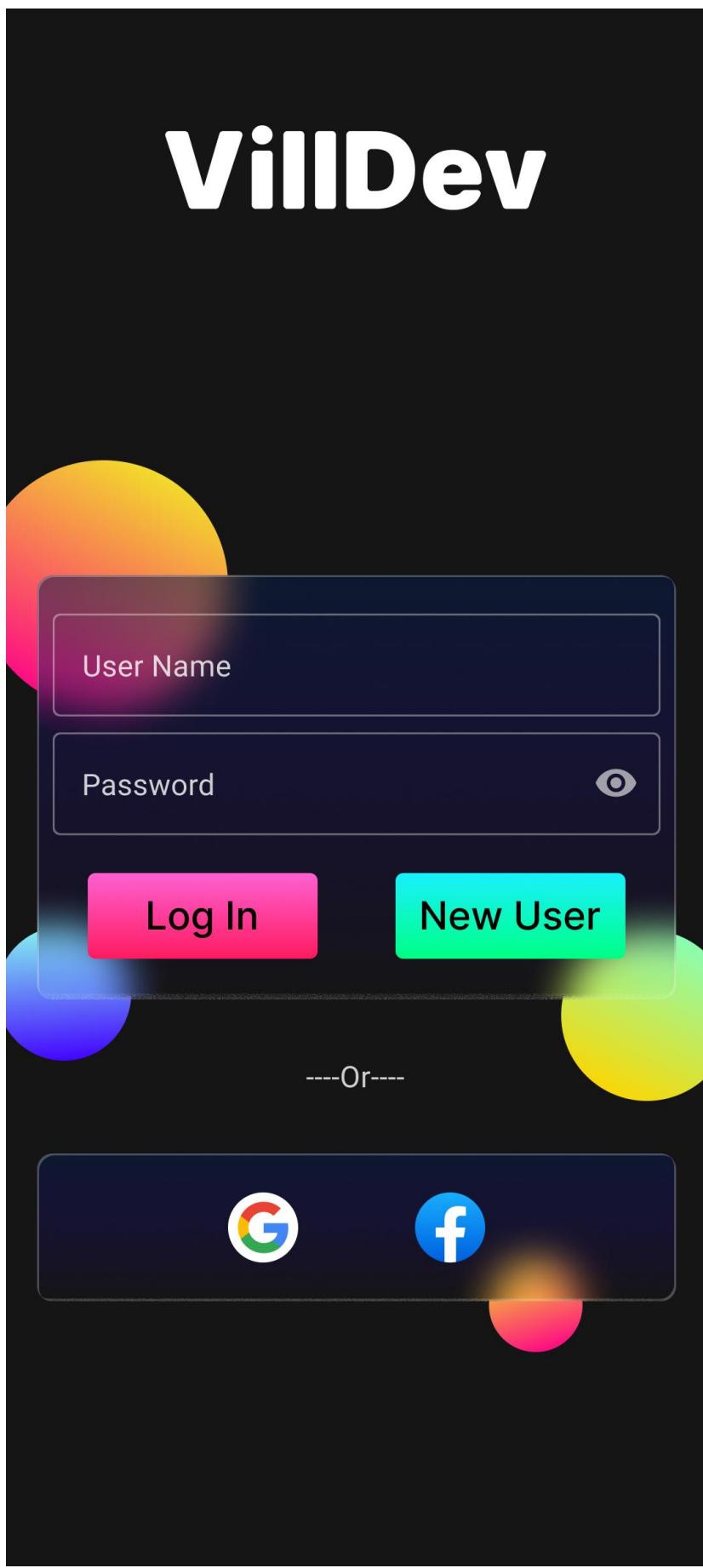
TIMELINE CHART

When creating a software project schedule, the planner begins with a set of tasks. If automated tools are used, the work breakdown is input as a task Network or task outline. Effort, duration, and start date are then input for each task. In addition, tasks may be assigned to specific individuals. A timeline chart, also called a Gantt chart, is generated.

A timeline chart can be developed for the entire project. All project tasks are listed in the left-hand column. The horizontal bars indicate the duration of each task. When multiple bars occur at the same time on the calendar, task concurrency is implied. The diamond indicates milestones.



GRAPHICAL USER INTERFACE



User ID

Password



Confirm Password

Name

Address

Contact No.

Email ID

Next

--User Role--



Submit

--User Role--



Sarpanch

Villager

Submit

Sarpanch



Add Photo



Add Aadhar Card



Valid Proof Of Sarpanch



Submit

Villager



Add Photo



Add Aadhar Card

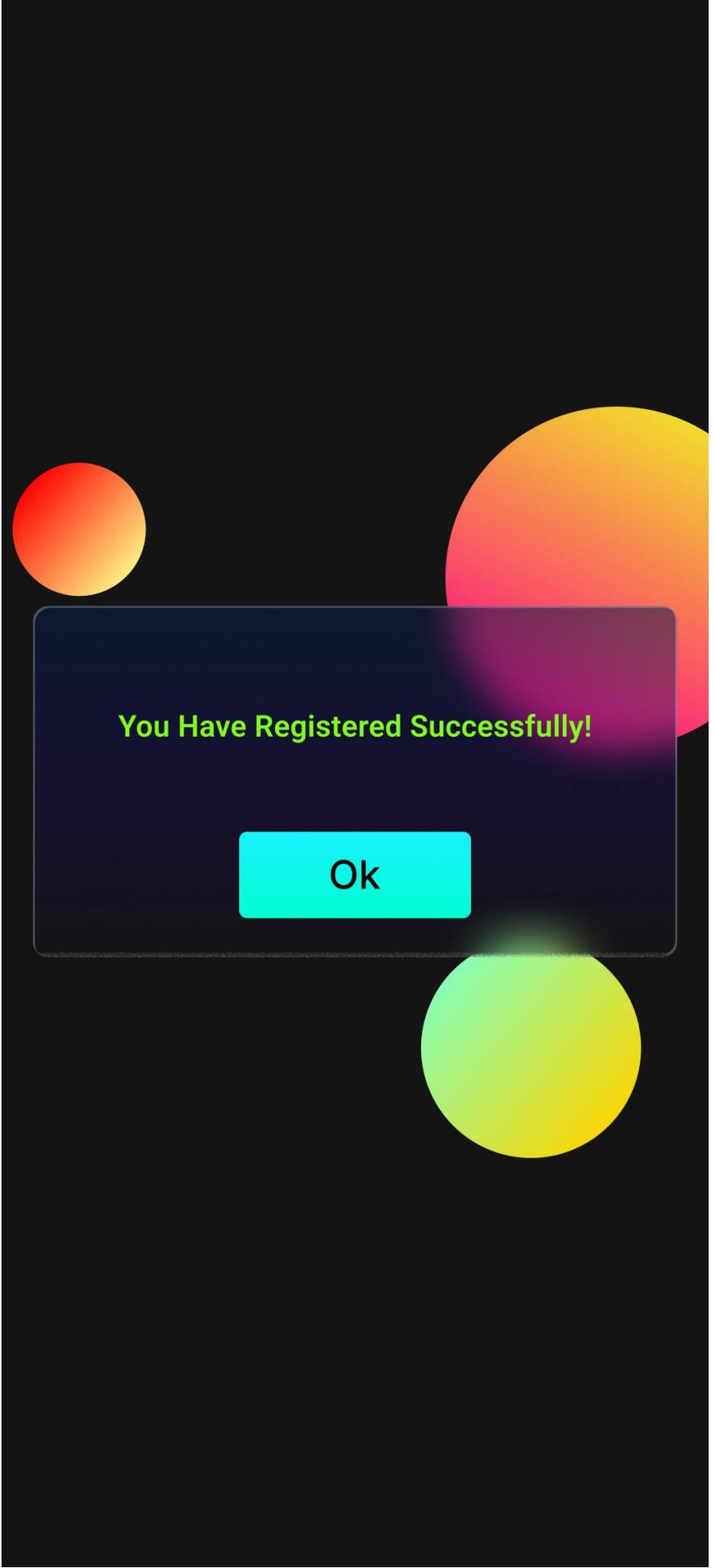


Submit

You Have Registered Successfully!

Onground Verification Will Be Held
Within 24 Hours And Then You Will
Get The Login ID PASSWORD

Ok



You Have Registered Successfully!

Ok



VillDev



DEVELOPMENT



HIRE
ENGINEER



BUDGET DETAILS



WORK STATUS



COMPLAINTS



SUGGESTIONS



RECORDS



UPDATE





VillDev



BUDGET DETAILS



WORK STATUS



COMPLAINTS



SUGGESTIONS



RECORDS



UPDATE



--Select Your Complaint--

- Electricity
- Water
- Road
- Other

Enter Your complaint...

Submit

--Select Your Complaint--

- Electricity
- Water
- Road
- Other

Submit

Complaints

Complaint Subject

User Id

Name

--Happy Code--

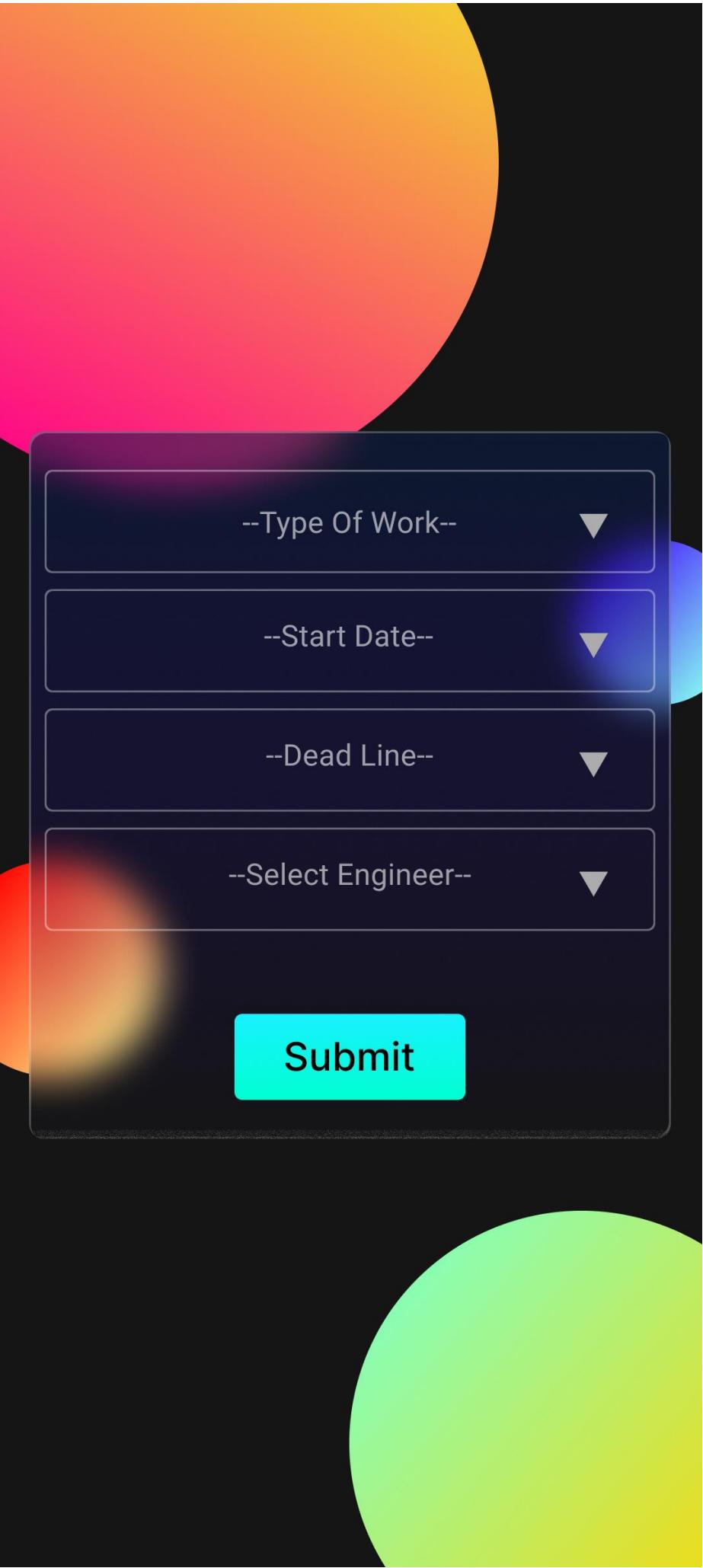
Resolve

Complaints

Complaint Subject

Complaint Resolved Successfully

Ok



Enter Your Suggestion...

Submit

Cyclomatic Complexity

Cyclomatic complexity is a software metric that provides a qualitative measure of the logical complexity of a program. When used in the context of the basis path testing method, the value corresponding to cyclomatic complexity defines the number of independent paths in a basis set of a program and provides us with an upper bound for the number of tests that must be conducted to ensure that all statements have been executed at least once.

Use of cyclomatic complexity:

- Determining the independent path executions thus proven to be very helpful for developers and testers.
- It can make sure that every path has been tested at least once.
- This helps to focus more on uncovered paths.
- Code coverage can be improved.
- The risk associated with the program can be evaluated.
- These metrics being used earlier in the program help in reducing the risks.

Cyclomatic complexity has a foundation in graph theory and is computed in one of the three ways:

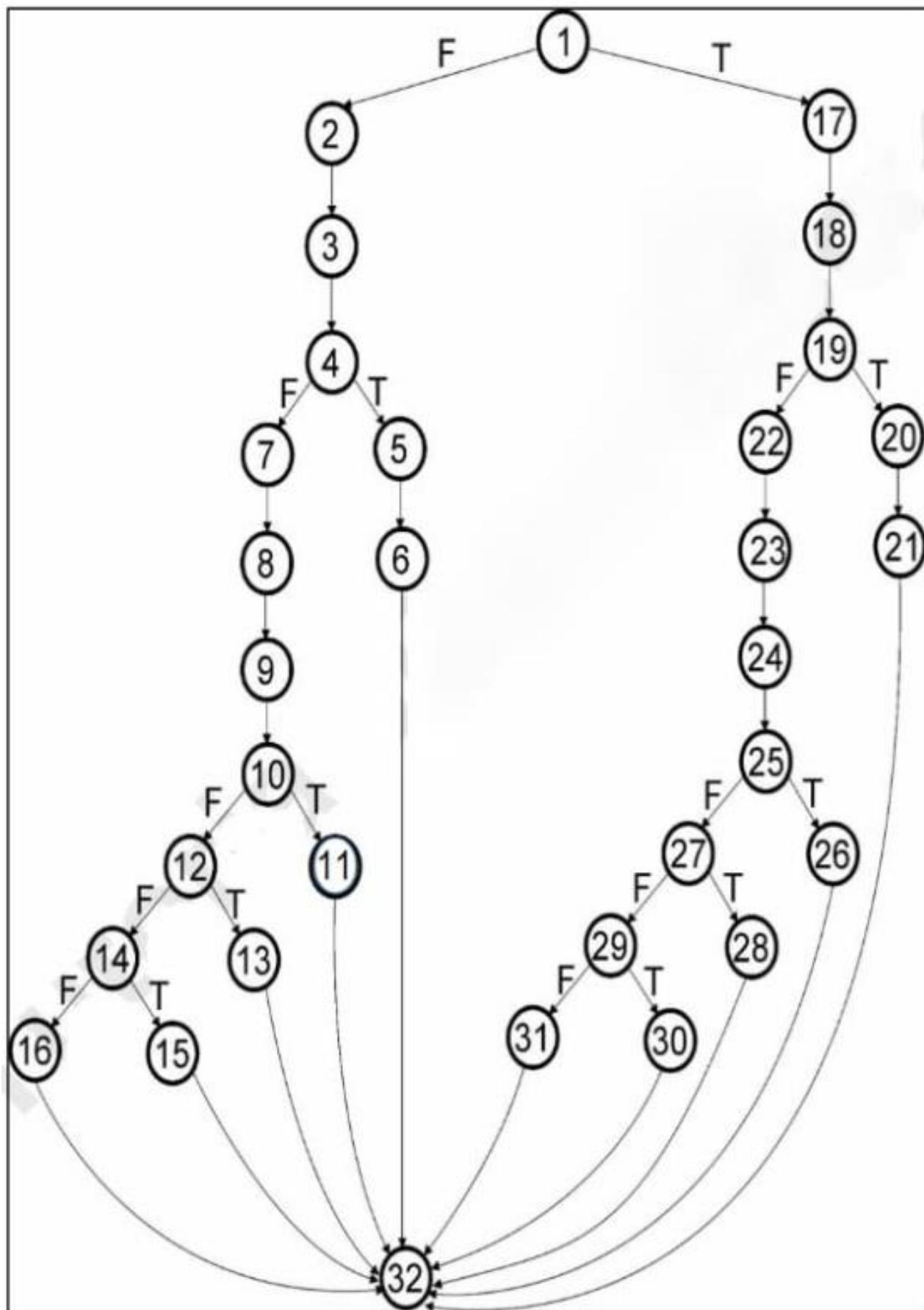
1. The number of regions corresponds to the cyclomatic complexity.
2. Cyclomatic complexity, $V(G)$, for a flow graph G , is defined as $V(G) = E - N + 2$
where E is the number of flow graph edges, and N is the number of flow graph nodes.
3. Cyclomatic complexity, $V(G)$, for flow graph, G is also defined as
$$V(G) = P + 1.$$
Where P is the number of predicate nodes contained in the flow graph G .

Pseudocode

Villagers/Sarpanch Login module

1. {Choose villagers or Sarpanch (0/1) login}
2. if (chosen == villagers) {
3. Enter email
4. If (email syntax is incorrect) {
5. Show (Please enter correct email)
6. Break;}
7. Else {
8. Enter password
9. Search the entered email in database
10. if (username does not exist)
11. Show (Check username or Register)
12. Else if (username exists but password does not match) {
13. Show (Check username or password)
14. Else if (username and password match)
15. Go to event selection screen}
16. Else break;}
17. If (chosen = Sarpanch) {
18. Enter email
19. If (email syntax is incorrect) {
20. Show (Please enter correct email)
21. Break;}
22. Else {
23. Enter password
24. Search the entered email in database
25. if (username does not exist)
26. Show (Check username or Register)
27. Else if (username exists but the password does not match) {
28. Show (Check username or password)
29. Else if (username and password match)
30. Go to event selection screen}
31. Else break;}
32. }

Flow Graph:



Computed cyclomatic complexity

- Region = 10
- $V(G) = E - N + 2$
$$V(G) = 40 - 32 + 2 = 10$$
- $V(G) = P + 1 = 9 + 1 = 10$

Basis Set independent Paths

Path 1: 1-2-3-4-7-8-9-10-11-32

Path 2: 1-2-3-4-7-8-9-10-12-14-16-32

Path 3: 1-2-3-4-7-8-9-10-12-13-32

Path 4: 1-2-3-4-7-8-9-10-12-14-15-32

Path 5: 1-2-3-4-5-6-32

Path 6: 1-17-18-19-22-23-24-25-26-32

Path 7: 1-17-18-19-22-23-24-25-27-28-32

Path 8: 1-17-18-19-22-23-24-25-27-29-30-32

Path 9: 1-17-18-19-22-23-24-25-27-29-31-32

Path 10: 1-17-18-19-20-21-32

Risk Management

Risk: computer crash

- **Mitigation:** The cost associated with a computer crash resulting in a loss of data is crucial. A computer crash itself is not crucial, but rather the loss of data. A loss of data will result in not being able to deliver the product to the customer. This will result in not receiving a letter of acceptance from the customer. Without the letter of acceptance, the customer will receive a failed process mail. As a result, the organization is taking steps to make multiple backup copies of the software in development and all documentation associated with it, in multiple locations

,

- **Monitoring:** When working on the product or documentation, the staff member should always be aware of the stability of the computing environment they're working in. Any changes in the stability of the environment should be recognized and taken seriously.
- **Management:** The lack of a stable-computing environment is extremely hazardous to a software development team. In the event that the computing environment is found unstable, the development team should cease work on that system until the environment is made stable again, or should move to a system that is stable and continue working there.

Risk: Late Delivery:

- **Mitigation:** The cost associated with a late delivery is critical. A late delivery will result in a late delivery of a letter of acceptance from the customer. Without the letter of acceptance, the group will receive a failing grade for the course. Steps have been taken to ensure a timely delivery by gauging the scope of project based on the delivery deadline.
- **Monitoring:** A schedule has been established to monitor project status. Falling behind schedule would indicate a potential for late delivery. The schedule will be followed closely during all development stages.
- **Management:** Late delivery would be a catastrophic failure in the project development. If the project cannot be delivered on time the development team will not pass the course. If it becomes apparent that the project will not be completed on time, the only course of action available would be to request an extension to the deadline form the customer.

Risk: Technology Does Not Meet Specifications

- **Mitigation:** In order to prevent this from happening, meetings (formal and informal) will be held with the customer on a routine business. This ensures that the product we are producing, and the specifications of the customer are equivalent.
- **Monitoring:** The meetings with the customer should ensure that the customer and our organization understand each other and the requirements for the product.
- **Management:** Should the development team come to the realization that their idea of the product specifications differs from those of the customer, the customer should be immediately notified, and whatever steps necessary to rectify this problem should be done. Preferably a meeting should be held between the development team and the customer to discuss at length this issue.

Risk: End Users Resist System

- **Mitigation:** In order to prevent this from happening, the software will be developed with the end-user in mind. The user interface will be designed in a way to make use of the program convenient and pleasurable.
- **Monitoring:** The software will be developed with the end-user in mind. The development team will ask the opinion of various outside sources throughout the development phases. Specifically, the user-interface developer will be sure to get a thorough opinion from others.
- **Management:** Should the program be resisted by the end-user; the program will be thoroughly examined to find the reasons that this is so. Specifically, the user interface will be investigated and if necessary, revamped into a solution.

Risk: Database is not Stable

- **Mitigation:** In order to prevent this from happening, developers who are in contact with the database, and/or use functions that interact with the database, should keep in mind the possible errors that could be caused due to poor programming/error checking. These issues should be brought to the attention of each of the other members that are also in contact with the database.
- **Monitoring:** Each user should be sure that the database is left in the condition it was before it was touched, to identify possible problems. The first notice of database errors should be brought to the attention of the other team members.
- **Management:** Should this occur, the organization would call a meeting and discuss the addition of new topics, or removal of unnecessary topics into the documentation.

Risk: Poor Quality Documentation

- **Mitigation:** In order to prevent this from happening, members who are in charge of developing the documentation will keep in contact with each developer on the team. Meetings will be held routinely to offer documentation suggestions and topics. Any topic deemed missing by a particular developer will be discussed and it will be decided whether or not to add that particular topic to the documentation. In addition, beta testers will be questioned about their opinion of the documentation.
- **Monitoring:** Throughout development or normal in and out of house testing, the development team and or beta testers will need to keep their eyes open for any possible documentation topics that have not been included.
- **Management:** Should this occur, the organization would call a meeting and discuss the addition of new topics, or removal of unnecessary topics into the documentation.

SYSTEM TESTING

- Testing is the process of executing a program with the intent of finding an error. A good test case is one that has a high probability of finding undiscovered errors.
- Software testing is a critical element of software quality assurance and represents the ultimate review of specification, design, and coding.
- The purpose of product testing is to verify and validate the various work products viz. units, integrated units, and final products to ensure that they meet their requirements.

Testing for our project:

Acceptance testing:

Alpha testing:

When software testing is performed internally within the organization. Our project will be tested by the developed website.

Beta testing:

When the software testing is performed from the limited number of people who's the college students of university of Delhi who need to search, view and download course question paper and credit points.

Black Box Testing:

- Black Box testing is also called functional testing.
- Black Box Testing is a test case design method that focuses on the functional requirements of the software that is it enables the software engineer to derive a set of input conditions that fully exercise all functional requirements for a program.
- Test the artifacts from the external point of view.
- Specifications are used to test data that is what type of input should be given to the unit or module should be specified.
- We can check the functionality on the basis of the output generated and the input, not looking at the internal coding.
- It attempts to find errors in the following categories
 - Incorrect or missing functions
 - Interface errors
 - Errors in data structure or External database access
 - Behavior or performance error
 - Initialization and termination errors

White Box Testing:

- It is also called glass box testing.
- White Box testing is test case design method that uses the control structure of the procedural design to derive test cases.
- Using White Box Testing method, the software engineer can derive test cases that
 - Guarantee that all independent paths within a module have been exercised at least once.
 - Exercise all logical decisions on their true and false sides.
 - Execute all loops at their boundaries and within their operational bounds.
 - Exercise internal data structures ensure their validity.
- Test the artifacts from the internal point of view.
- It cannot detect the absence of features.
- For security purposes the Email of the user is required in case he/she forgets his/her password and wants to retrieve that.

Test cases

Test Cases are good in revealing the presence of faults. Successful implementation of test cases implies that there is no error in the program. Test cases should be minimum as they are expensive in case of money & effort. The primary objectives of test cases are to ensure that if there is an error or fault in the program it is exercised by the test cases. An ideal test case set is one that succeeds only if there are no errors in the program. One possible ideal set of the test cases is one that includes all possible I/P to the program and is called exhaustive testing. A test case is good if it detects an undiscovered error in Program.

Sr No.	Test Cases	Feature	Description	Steps to execute	Expected Results	Actual result	Status
1	TC-01	User Interface	Check all the text boxes and buttons	Check Page	UI should be perfect • Text boxes and button should be aligned	UI should be perfect • Text boxes and button should be aligned	Pass
2	TC-02	Required Fields	Check the required fields by not filling any data.	1. Enter invalid username 2. Enter correct password 3. Click on Login Button	User should not log in and should show proper error message	User should not log in and should show proper error message	Pass
3	TC-03	User Login	Check When passing a correct username and invalid password	1. Enter valid username 2. Enter incorrect password 3. Click on Login Button	User should not log in and should show proper error message	User should log in and should show proper error message	Error
4	TC-04	User Login	Check when pass correct email and password	1. Enter valid username 2. Enter valid password 3. Click on Login Button	User should log in	User should log in	Pass
5	TC-05	Signup Option for new users	Check whether the signup link for the new user is working	Click Signup link	Clicking signup link takes the user to signup page successfully	Clicking signup link takes the user to signup page successfully	Pass
6	TC-06	Forgot Password	Verify user should get an error message when he/she enters not registered email id.	1. Click on the Forgot password link. 2. Enter unregistered email id and click on the send button.	User should get an error message.	User should get an error message.	Pass
7	TC-07	User Forgot Password.	Verify Forgot Password sends a forgot password link.	1. Click on the Forgot Password link. 2. Enter Email and click on the send button. 3. Now go to mail7.io and enter the email id.	User should get the forgot password link on his/her email id.	User should get the forgot password link on his/her email id.	Pass

REFERENCES

- 1. Pressman, R. S., & Maxim, B. R. (2015). Software Engineering: A Practitioner's Approach 8th edition.**
- 2. Aggarwal, K. K., & Singh, Y. (2007). Software Engineering.**
- 3. <https://www.softwaretestinghelp.com>**
- 4. <https://www.softwaretestingo.com>**

