“CROSS-PLATFORM TUTORIAL APPLICATION”

A Project Report Submitted In Partial Fulfillment of the Requirements For the Degree of

BACHELOR OF TECHNOLOGY

in

Computer Science and Engineering

by

Niteesh dubey (CSE)–1604610055
Athar sameen khan (CSE)-1604610033
Mayank mani (CSE)-1604610049
Mayank Animesh (CSE)-1604610048
Abhayanand(CSE)-1604610000

Under the precious supervision of
Mrs. NIKITA MAM
Assistant Professor, Computer Science and Engineering Department

MAHARANA INSTITUTE OF PROFESSIONAL STUDIES KANPUR NAGAR
(Affiliated to Dr. A.P.J Abdul Kalam Technical University,Lucknow)
August , 2020

CERTIFICATE

This is to certify that the seminar report is goldeous opportunity on the topic of **“ cross-platform tutorial application”** submitted by **Niteesh dubey (CSE)–1604610055, Athar sameen kh (CSE)-1604610033, Mayank mani (CSE)-1604610049, Mayank Animesh (CSE)-16 Abhayanand(CSE)-1604610000** in partial fulfilment for the award of degree of **Bachelor of Technology** in **Computer Science & Engineering** has been found satisfactory and is approved for submission.

Ms. Nikita Mam

Assistant Professor

Dept. of CSE

MPGI

ACKNOWLEDGEMENT

It gives us a great sense of pleasure to present the report of the B.Tech project undertaken during B.Tech 4 th year. I owe special debt of gratitude to **Mrs Nikita Mam** (Head of department in Computer Science and Engineering).

The satisfaction that accompanies the successful completion of any task would be incomplete without the mention of people whose ceaseless cooperation made it possible, whose constant guidance and encouragement crown all efforts with success.

We are grateful to our project guide **Mrs. Nikita Mam** for the guidance, inspiration and constructive

suggestions that help us in the preparation of this project.

We also thank our colleagues who have helped in successful completion of the project.

Niteesh dubey (CSE)–1604610055
Athar sameen khan (CSE)-1604610033
Mayank mani (CSE)-1604610049
Mayank Animesh (CSE)-1604610048
Abhayanand(CSE)-1604610000

TABLE OF CONTENTS:

TITLE PAGE No.

CHAPTER 5 PROJECT DESCRIPTION

CHAPTER 6 PROJECT DESIGN

CHAPTER 7 TESTING

CHAPTER 8 SNAPSHOTS

INTRODUCTION

1. ABSTRACT:

This system facilitates education among users of an Internet. Now-a-days organizations are growing fast and are increasing in size also. So these organizations are divided into departments. In the fast growing world the information is needed as fast as possible. This can be accomplished by passing the information quickly. Quick passing of education is not possible in manual systems. This is because the information is passed through persons of one department to another department. This leads to the inconsistency and delay in delivering information. So we need a system which provide everything in one app. This can be achieved by developing a cross-platform tutorial application.

It will work just like a classroom for users to learn different things in all in single platform. All users have to register using this system to get a unique id and select a strong password as per the given choices. This system will work as host as well as for their clients and all information will be stored at central server. To establish fast communication between content's and users id will be available directly under the profile of particular users, each users have there own ID.

2. OBJECTIVES: Existing System

For tutorial app, users were using other application such as solo, google classroom and many more, but it was not possible for the users to get everything in a single application. It was not able to provide a professional environment for professional workers. These system were provided with full features as per the organization requirements and send feeds and notification to the users to give updates with there activity.

Present Application is limited with there features for there users. Users have to download the different apps to perform different task to get there requirements complete. Sometimes information that you need will not be available. This application will take time to update the content it causes loss of information also. There by causing loss of users time also. Thus the present system stated is **time taking**, **insecure** and **costly**

Proposed System

As the system will be installed at any particular server which will act as host and its client's means working users can access this system using particular IP address or by using its registered domain name. Each users will use their id and password to login and access their panel to perform activities such as chat, send messages and documents, blog, book of different course, videos etc. The concept of session will able to show the history details for each items such as your performance and completion of task, that you have covered in the given application.

Users will be able to see the name, performance course completion, your favrouties and different blogs on different categories. It will also enable them to schedule reminder on particular date and time and pre-defined format will provide availability to perform tasks me as per the time you have set in the application. As organization grows in size in terms of departments and functionalities, it requires a quick and efficient system to achieve instant communication b/w developers of the application and the users. The proposed system **"cross-platform tutorial application"** serves organization's needs in a consistent and transparent manner. It should cater the needs of information sharing. It allows the users to exchange their views thru mails and send electronic files thru attachments. It should have all traditional things such as sent items, inbox, drafts etc. The users are allows to send mails to multiple users using to, cc and bcc too. Thus the system caters spontaneous needs of the organization.

3. DOCUMENT CONVENTION:

The following documentation convention are followed in preparing this report:

- All keywords related to the academics are formatted in bold.

- SRS - Software Requirement Specification
- DFD – Data Flow Diagram

4. INTENDED AUDIENCE AND READING SUGGESTION

This document is created for:

- The members of the organization for their review and monitoring progress of the project.
- For their instant communication to take place in a secure and in an authenticated manner.
- The software development team for their use in analyzing the requirements.

5. PROJECT SCOPE:

An cross-platfrom app will be applicable everywhere, and any one can use it ‘who wants to learn and grow there knowledge’. It will be more efficient and easier way to have a content on the apps . As all the content will be present in a centralized way, user can learn, earn and grow by performing practices given by the app.

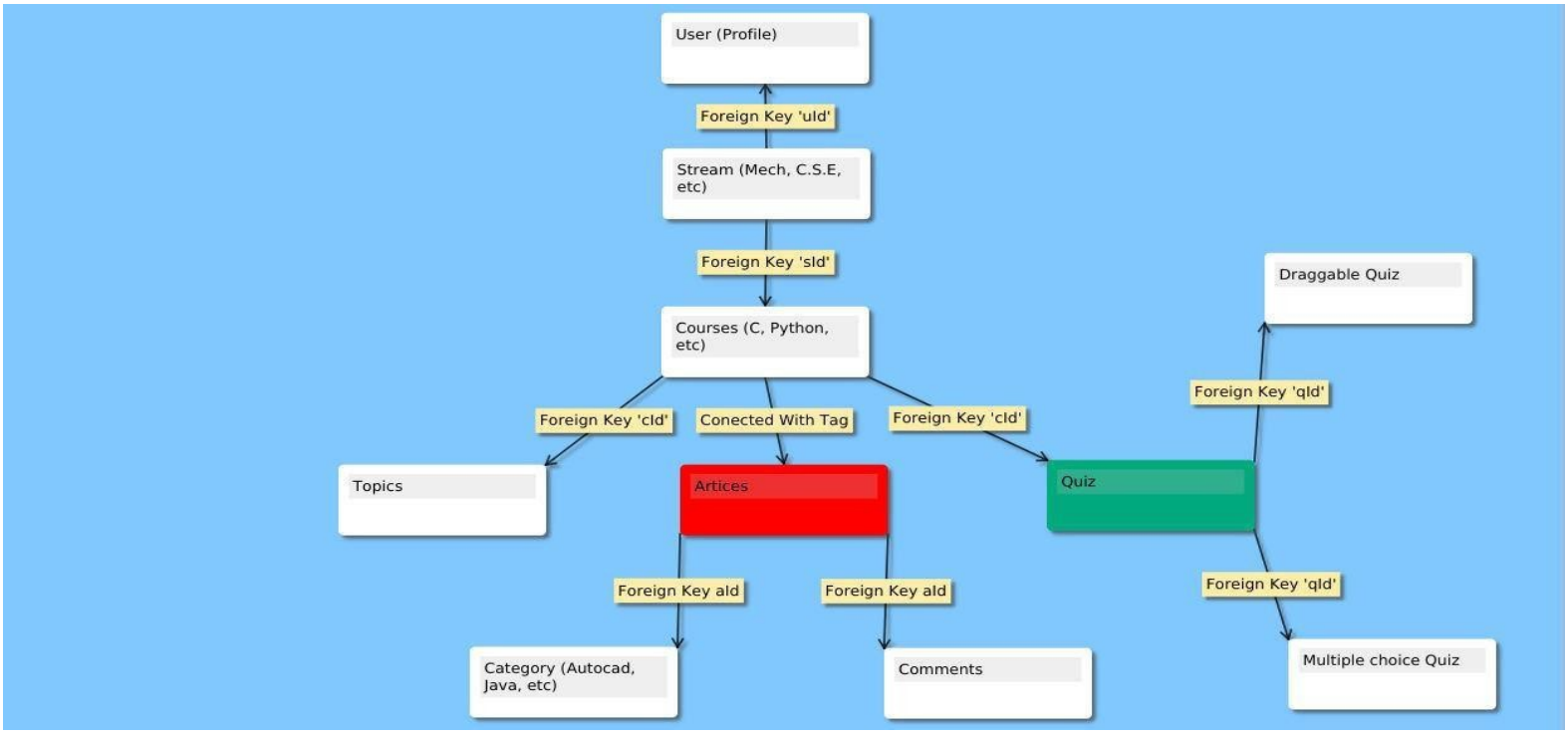
OVERALL DESCRIPTION

1) - PROJECT PERSPECTIVE:

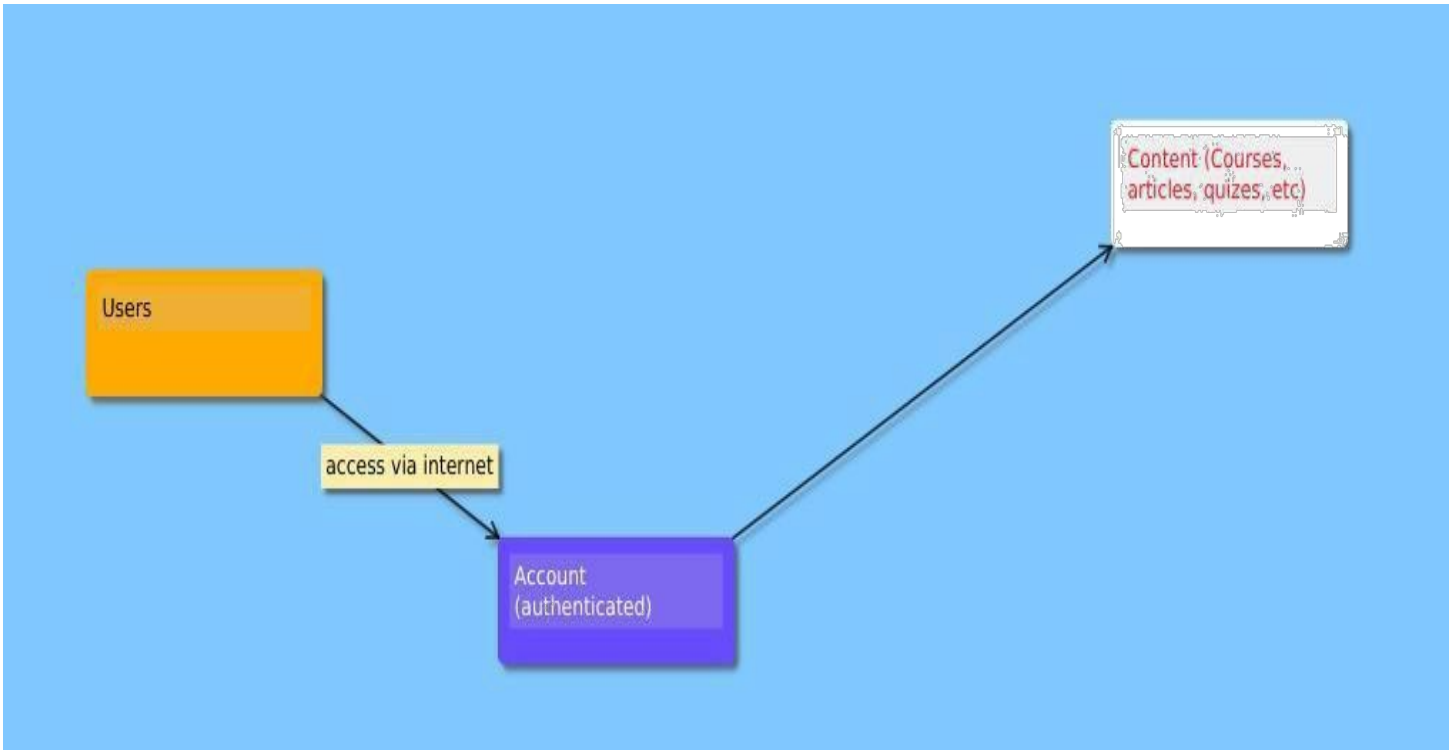
An cross-platfrom app will be applicable everywhere, and any one can use it ‘who wants to learn and grow there knowledge’. It will be more efficient and easier way to have a content on the apps . As all the content will be present in a centralized way, user can learn, earn and grow by performing practices given by the app.

2) - PRODUCT PERSPECTIVE:

Following is the arrangment of the screen connected to each other and transfering the data and showing data to each screen from database by the help of foreign keys.



In earlier apps, user should have to download different apps for each different streams(mech, C.S.E, etc) in the mobile. It wastes time.



After implementing the Tutlify(cross-platform app) user will be able
to connect to his account to access all the content from the app which is available in one single platform.

User:

In user module we have six options.

1. ***Check Courses.***
2. ***Create blogs or posts or articles.***
3. ***Delete posts.***
4. ***View posts.***
5. ***Search posts and courses.***
6. ***Send feedback*.**

🔔 When the user selects the “Send Mail” option, then the form will be opened for sending mail. Then the user can send mail to any user who has a user id in the organization.

🔔 In “Check Blogs” option, the user can check for the blogs, if the user has any blogs. The blogs are displayed. He will check the blogs and can delete the nnecessary blogs. If the user has no blogs then the message “No blogs” will be displayed.

➢ When the User selects the option “Generate Contact”, he will generate a new Contact, if any.

➢ When the administrator selects the option “Delete Contact”, he will delete an existing user, if any.

➢ When the administrator selects the option “Search Contact”, he will search an existing user, if any.

➢ When the user select the “View Contact” option, then the no. of contacts that are added, are displayed.

- **Registration:**

  This module is responsible for capturing and submitting the details of all theemployee of the System so that they can get the facility of mail, chat, search etc.

- **Login:**

  In this module User enter the User id and password is checked and only validuser id andpassword will getentry into member’s zone. This is a securityfeature to avoid entry of unauthorized users.

- **Inbox:**

  User and Admin can check their mail and also download the attachment fileswith their mail box.

- **Create Projects:**

  You can write any posts and also attach any document or image and send to your destinations.

  You can write any posts and also attach any document or image and send to your destinations.

- **Chat:** This is the Administrator’s module by which he keep the eye on whole site and maintain and upgrade the site’s service for sake of users. Administrator canshow banner ads of others, which help the site in revenue.

- ## Security Management Process:

This module provides functionality to change the user id and the password.

- ## Logout:

The Admin and User can successfully logout and their session will bedestroyed.

- ## Help:

Provide the functionality to get help about the any query.

4) - OPERATING ENVIRONMENT

- Web Presentation: HTML, CSS
- Client – side Scripting: JavaScript
- Programming Language: PHP
- Backend Database: MY SQL
- Operating System: Windows , LINUX

HARDWARE REQUIREMENTS

- Hard Disk -------- 20GB
- KEYBOARD -------- 108 Standard
- RAM Capacity -------- 128MB
- Pentium processor -------- 233 MHZ or above

5) -ASSUMPTIONS AND DEPENDENCIES:

There are no assumptions made in this project.

REQUIREMENT ANALYSIS

1) - HARDWARE REQUIREMENT:

- Hard Disk -------- 20GB
- KEYBOARD -------- 108 Standard
- RAM Capacity -------- 128MB
- Pentium processor -------- 233 MHZ or above

2) - SOFTWARE REQUIREMENT

- Web Presentation: Flutter Framework, Dart
- Client – side Scripting: Dart
- Programming Language: Javascript
- Backend Database: Firebase
- Operating System: Android, IOS

3) - FUNCTIONAL REQUIREMENT:

This application will have following functionalities:

1. Provide Different Streams:

Users will able to access all the streams from the single application by just clicking the selected streams provided in the app.

2. Object Scanner :

This is the features provided by the application in which a given application identifies the object and give the related information, related to that object.

3. Quizes:

A given application provides the quizes to check the ability of the user and rank them according to there scores.

4. Notify you about the topics to learn:

Provides the notification based on the daily performance of the user and gives them a suggestion which topics to cover and provides the daily articles.

5. Notify you about the topics to learn:

Provides the notification based on the daily performance of the user and gives them a suggestion which topics to cover and provides the daily articles.

6. Competet with your friends to earn more badges:

Provides the feature to connect with your friend by using a device hotspot or internet and competet with each other by playing some the games and quizes to earn the badges.

7. Logout:

The Admin and User can successfully logout and their session will be destroyed.

8. Help:

Provide the functionality to get help about the any query.

4) - NON FUNCTIONAL REQUIREMENT:

• SAFETY REQUIREMENT:

This software will ease the process of instant communication with members of same or different departments of that organization.

• SECURITY REQUIREMENTS:

This software will:

➢ Authenticate each user, who logs in.

➢ When the user performs any action, authorize him/her to perform the actions allowed for the user and displays an error message if found to be unauthorized.

• PERFORMANCE REQUIREMENT:

This software should be able to handle the following tasks:

It should be able to handle the content of the different users posts in firebase Google cloud.

CHAPTER 4

CHAPTER 4
FEASIBILITY STUDY

Feasibility Study is one of the fundamental aspects in design of system. The major feasibility studies that are conducted are Economic Feasibility, Social Feasibility and Political Feasibility. Economic Feasibility is also referred as Cost-Benefit Analysis. Here a study is done to ascertain that whether building the system will be beneficial for the organization in terms of the resources that are needed. When we talk of resources it includes both capital and human resources. In short the organization judges that building the system will be cost effective or not.

Social and Political Feasibility tests are basically conducted to judge the social and the political effects of the system. For example if by building a system it causes the displacement of hundreds of workers then the system is not feasible socially. If a system causes political uproar due to its presence it might not be politically feasible to build the system. However as in our case neither of the above facts are valid it is assumed that the system is economically, socially and politically feasible and hence we can proceed with the process of development of the system.

Cost Estimation

As part of this project, the costs and benefits associated with the proposed system compared and the project is economically feasible only if tangible or intangible benefits outweigh costs. The system development costs will be significant. So the proposed system is economically feasible.

Cocomo (Constructive Cost Model) is a regression model based on LOC, i.e. **number of Lines of Code** It is a procedural cost estimate model for software projects and often used as a process of reliably predicting the various parameters associated with making a project such as size, effort, cost, time and quality. It was proposed by Barry Boehm in 1970 and is based on the study of 63 projects, which make it one of the best-documented models.

The key parameters which define the quality of any software products, which are also an outcome of the Cocomo are primarily Effort & Schedule:

- **Effort:** Amount of labor that will be required to complete a task. It is measured in person-months units.
- **Schedule:** Simply means the amount of time required for the completion of the job, which is, of course, proportional to the effort put. It is measured in the units of time such as weeks, months.

Different models of Cocomo have been proposed to predict the cost estimation at different levels, based on the amount of accuracy and correctness required. All of these models can be applied to a variety of projects, whose characteristics determine the value of constant to be used in subsequent calculations. These characteristics pertaining to different system types are mentioned below.

Boehm's definition of organic, semidetached, and embedded systems:

1. **Organic** – A software project is said to be an organic type if the team size required is adequately small, the problem is well understood and has been solved in the past and also the team members have a nominal experience regarding the problem.

2. **Semi-detached** – A software project is said to be a Semi-detached type if the vital characteristics such as team-size, experience, knowledge of the various programming environment lie in between that of organic and Embedded. The projects classified as Semi-Detached are comparatively less familiar and difficult to develop compared to the organic ones and require more experience and better guidance and creativity. E.g.: Compilers or different Embedded Systems can be considered of Semi-Detached type.

3. **Embedded** – A software project with requiring the highest level of complexity, creativity, and experience requirement fall under this category. Such software requires a larger team size than the other two models and also the developers need to be sufficiently experienced and creative to develop such complex models.

All the above system types utilize different values of the constants used in Effort Calculations.

Topic :- Constructive Cost Model (COCOMO)

Comparison of three COCOMO modes

Mode	Project Size	Nature of Project	Innovation	Deadline of the Project	Development Environment
Organic	Typically 2 – 50 KLOC	Small Size Projects, experienced developers.	Little	Not tight	Familiar And In house
Semi-Detached	Typically 50 – 300 KLOC	Medium size project, average previous experience on similar projects.	Medium	Medium	Medium
Embedded	Typically over 300 KLOC	Large projects, complex interfaces, very little previous experience.	Significant	Tight	Complex Hardware / Customer interfaces required

Types of Models:

COCOMO consists of a hierarchy of three increasingly detailed and accurate forms. Any of the three forms can be adopted according to our requirements. These are types of COCOMO model:

1. Basic COCOMO Model
2. Intermediate COCOMO Model
3. Detailed COCOMO Model

The first level, **Basic COCOMO** can be used for quick and slightly rough calculations of Software Costs. Its accuracy is somewhat restricted due to the absence of sufficient factor considerations.

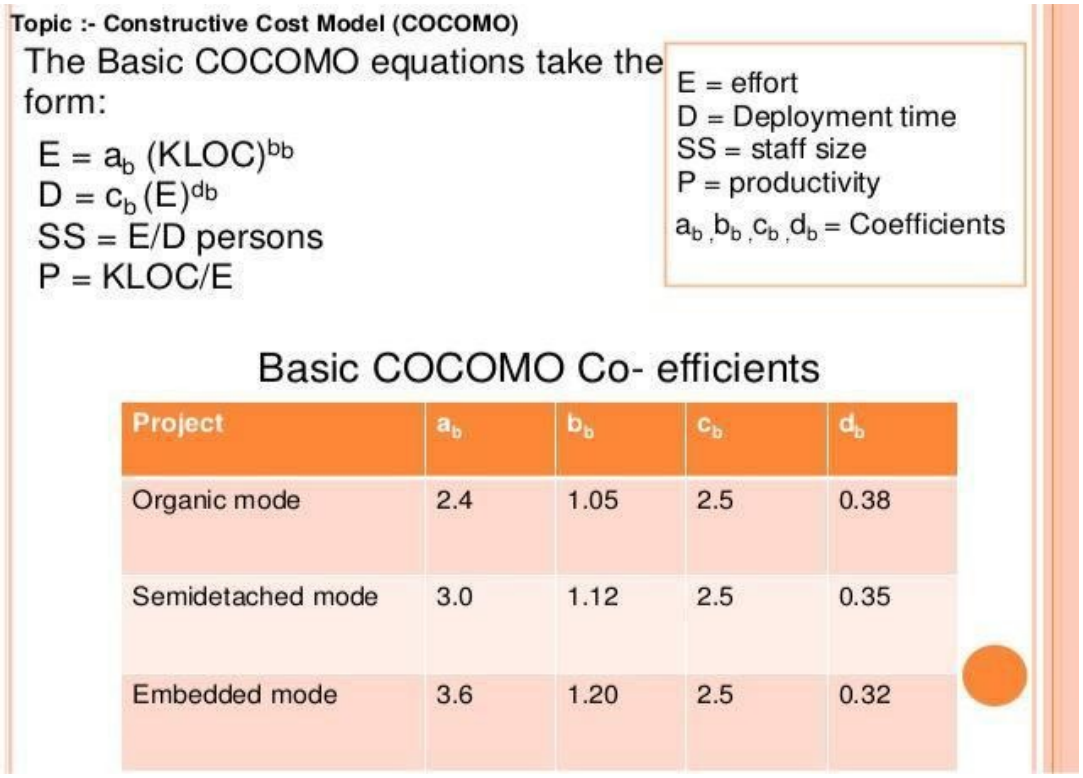
Intermediate COCOMO takes these Cost Drivers into account and **Detailed COCOMO** additionally accounts for the influence of individual project phases, i.e. in case of Detailed it accounts for both these cost drivers and also calculations are performed phase wise henceforth producing a more accurate result. These two models are further discussed below.

BASIC MODEL:

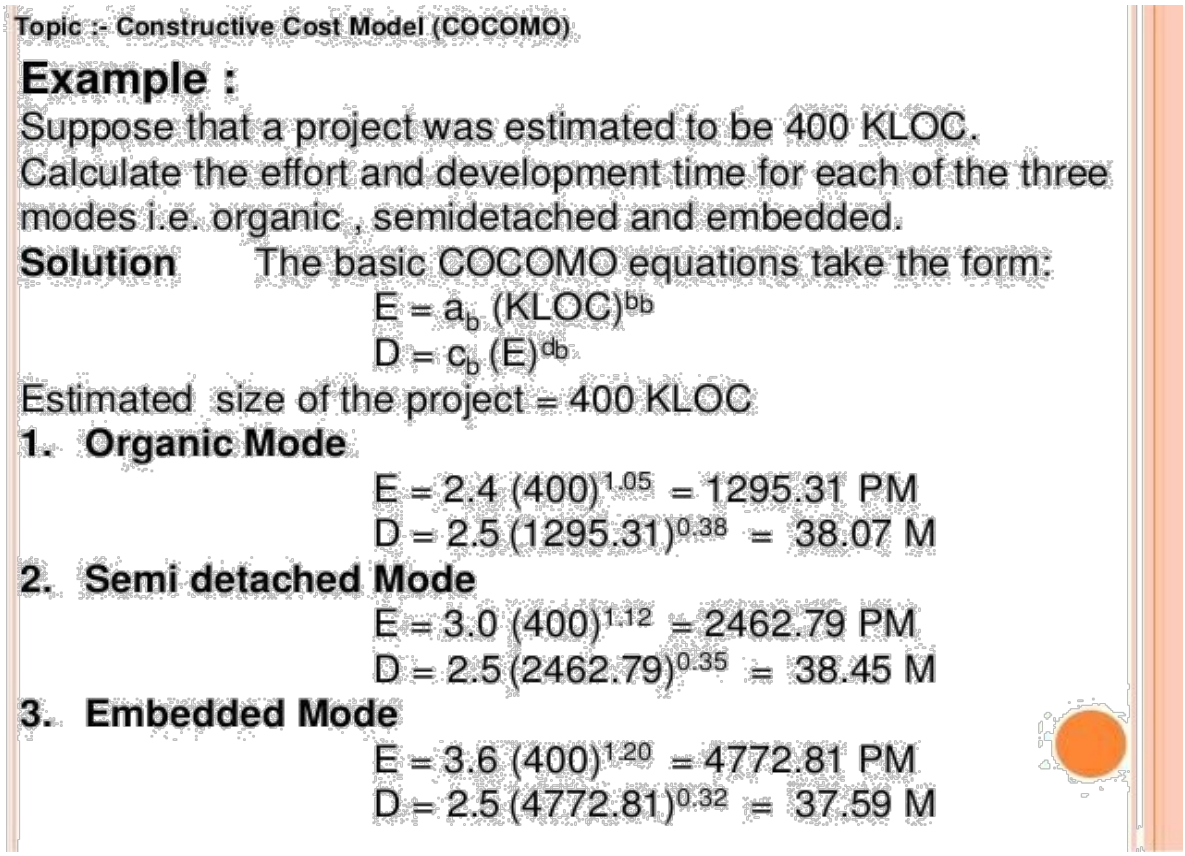
E=a *KLOC^b

The above formula is used for the cost estimation of for the basic COCOMO model, and also is used in the subsequent models. The constant values a and b for the Basic Model for the different categories of system.

The effort is measured in Person-Months and as evident from the formula is dependent on Kilo-Lines of code. These formulas are used as such in the Basic Model calculations, as not much consideration of different factors such as reliability, expertise is taken into account, henceforth the estimate is rough.



Project	a_b	b_b	c_b	d_b
Organic mode	2.4	1.05	2.5	0.38
Semidetached mode	3.0	1.12	2.5	0.35
Embedded mode	3.6	1.20	2.5	0.32



In our project, there are 5 KLOC , so:

Effort= a*(KLOC)^b

E= 2.4*(5)^1.05

E=13.0055 PM

Development Time= c*E^d

DT =2.5*(13.0055)^0.38

DT =6.62686753 M

Feasibility Report

After analyzing the existing system, the users are in need of automation of existing manual system. The users have the capacity to stand the cost of developing new system and are willing to do that. The product will be of utmost use and the level of ease has been increased to a great extent.

The objective is to determine whether or not the proposed system is feasible. The three tests of feasibility have been carried out.

1) -Technical Feasibility

In Technical Feasibility study, one has to test whether the proposed system can be developed using existing technology or not. It is planned to implement the proposed system using Java. It is evident that the necessary hardware and software are available for development and implementation of the proposed system. Hence, the solution is technically feasible.

2) - Economic Feasibility:

As part of this, the costs and benefits associated with the proposed system compared and the project is economically feasible only if tangible or intangible benefits outweigh costs. The system development costs will be significant. So the proposed system is economically feasible.

3) - Operational Feasibility:

It is a standard that ensures interoperability without stifling competition and innovation among users, to the benefit of the public both in terms of cost and service quality. The proposed system is acceptable to users. So the proposed system is operationally feasible.

CHAPTER 5

CHAPTER 5

PROJECT DESCRIPTION

---

1)-PROJECT CATEGORY:

Tutorial application is a cross-platform app which has the ability to run on different devices having the different **Operating systems, It has the ability to run on Android, IOS Web, Desktop, Linux and Mac.**

Project uses two different techs:

- REST API's (Repersentative state transfer application interface).
- Application uses the firebase owned by the google and run on the Google Cloud, uses the serverless technology.

2) - MODULES:

- **Registration**
- **Login**
- **New User**
- **courses**
- **Timeline**
- **Feeds**
- **messaging**
- **Scanner**
- **Create Blog**
- **Trash**
- **Update Profile**
- **Logout**

But, the "tutorial application" consists of mainly 4 modules.

1. User Profile Management
2. Courses
3. Blogs
4. Feeds

User Profile Management

This module facilitates new user registration, sign in of existing user, password recovery and user profile management.

Courses:

This modules consist of different list of courses and there sets quizes as this courses are divided into different categories based on the selection of the users and these categorys comes under the different streams as the application ask form the user to select the streams and then the aplication provide the different category based on the streams you have selected and provide the different blogs based on the categories which you are currently on.

Blogs:

This module provides the articles based on the stream and category and courses you are working on.

Feeds:

It is an important feature of this system. With this feature, this system provide the notifications of different tasks and the reminder of the tasks that user have set on the application.

3) **- INPUT OF PROJECT:**

Login page get the input of user id and password.

- Create the new user id for your Profile yourself
- Create Blog
- Get updated articles
- Dynamically change the streams
- Change password
- Add image
- create blogs

4) - OUTPUT OF PROJECT:

- View Feeds
- Updates from developer
- blogs
- Trash
- View Profile
- Help
- Home

5) - BENEFITS OF PROJECT:

- Fully works as a online
- Reducing the time
- Easy to customize when to learn and where to learn.
- Centralized maintain all information.
- Easy to Searching the information.
- Easy to managing the whole performance.

6) - FUTURE SCOPE:

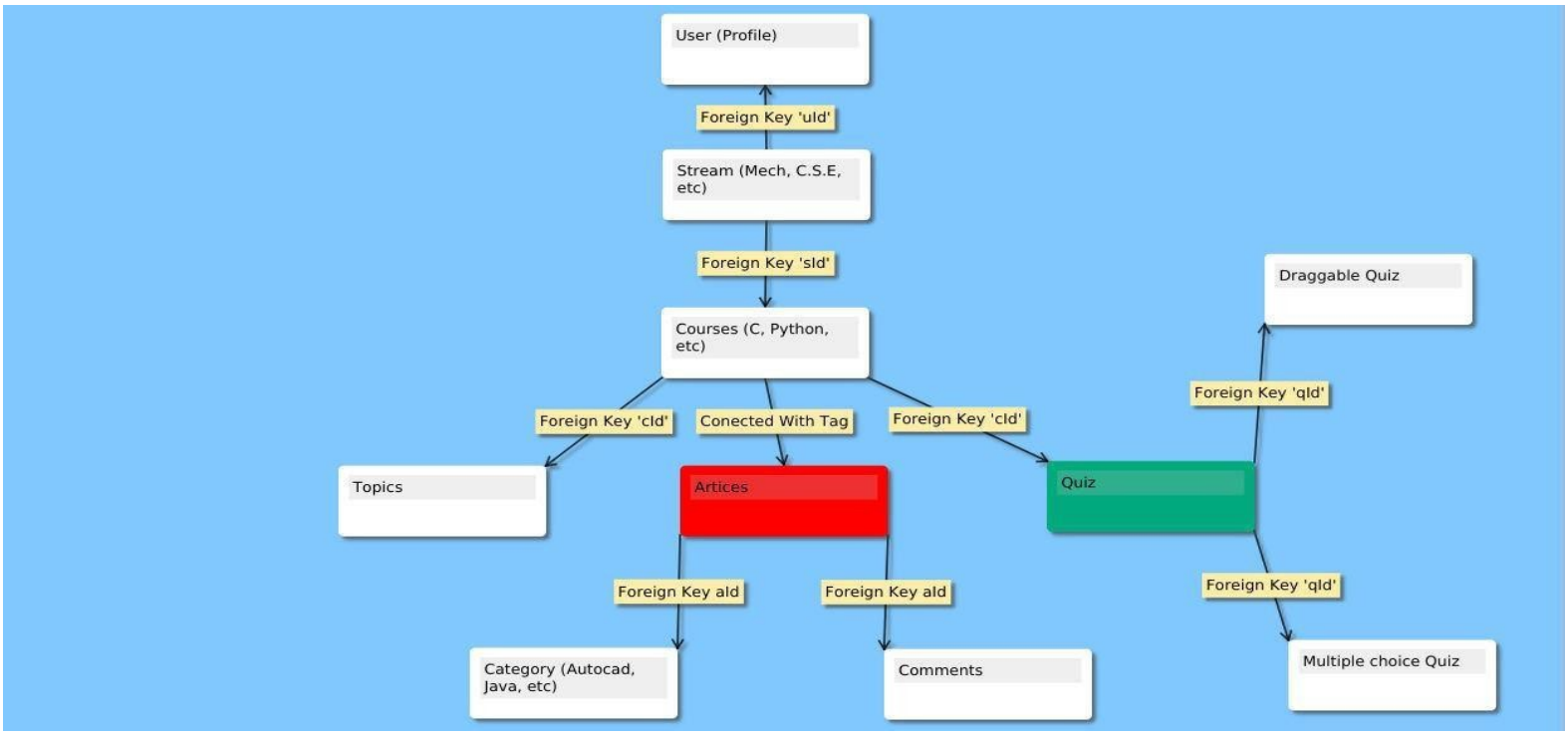
This project is created for those user who want to learn the given things in brief and get updated related to that techs as this application charges nothing so its a free app for users, but if you want to use all the features then you have to pay for that features. There are some advances which can be done on this app , as these are object identifier , it easily identify the object and give you the tech related content on that object which have been identified similar to google glass.

CHAPTER 6

CHAPTER 6

PROJECT DESIGN

This chapter describes the design decisions made to meet the aims that are specified in the specifications required. This chapter will also bring in some of the ideas and research outlined in the previous chapter.



Overview of system

➢ Here is the user profile which conected to the streams as, user can select the streams based on there requirements..

➢ Streams provide you a different course based on the stream you have selected and the content related to.

➢ User will get the notification based on the streams you have selected.

➢ It provides you a articles quizes and topics based on the courses you have selected.

Firebase: Is a famous serverless backhend sytem which runs on the cloud and the cloud is supported by the google.

Firebase Messaging Cloud: (FMC) Is a distributed system which is handle by the google and firebase management team, as FMC provides the **realtime push notification** the given app of the users as these notifications are different for different users.

1) - DATA FLOW DIAGRAM (DFD):

Data flow diagrams model the flow of data into, through, and out of an information system:

- Show the processes that change or transform data.
- Show the movement of data between processes.
- Represent a system as a network of processes which transform data flowing between them.

The user screen flow shows what a user of the community will see. After successfully logging on, the user will be given various links (such as search users, search boards, view course, blog etc.), and be able to select options from there, or go back to their home.

Data flow diagram is graphical representation of flow of data in an information system. It is capable of depicting incoming data flow, outgoing data flow and stored data. The DFD does not mention anything about how data flows through the system.

There is a prominent difference between DFD and Flowchart. The flowchart depicts flow of control in program modules. DFDs depict flow of data in the system at various levels. DFD does not contain any control or branch elements.

Types of DFD

Data Flow Diagrams are either Logical or Physical.

- **Logical DFD** – This type of DFD concentrates on the system process, and flow of data in the system. For example in a Banking software system, how data is moved between different entities.
- **Physical DFD** – This type of DFD shows how the data flow is actually implemented in the system.

➢ DFD COMPONENTS:

DFD can represent Source, destination, storage and flow of data using the following set of components –

- **Entities** – Entities are source and destination of information data. Entities are represented by a rectangles with their respective names.

- **Process** – Activities and action taken on the data are represented by Circle or Round-edged rectangles.

- **Data Storage** – There are two variants of data storage - it can either be represented as a rectangle with absence of both smaller sides or as an open-sided rectangle with only one side missing.

- **Data Flow** – Movement of data is shown by pointed arrows. Data movement is shown from the base of arrow as its source towards head of the arrow as destination.

➢ **Levels of DFD:**

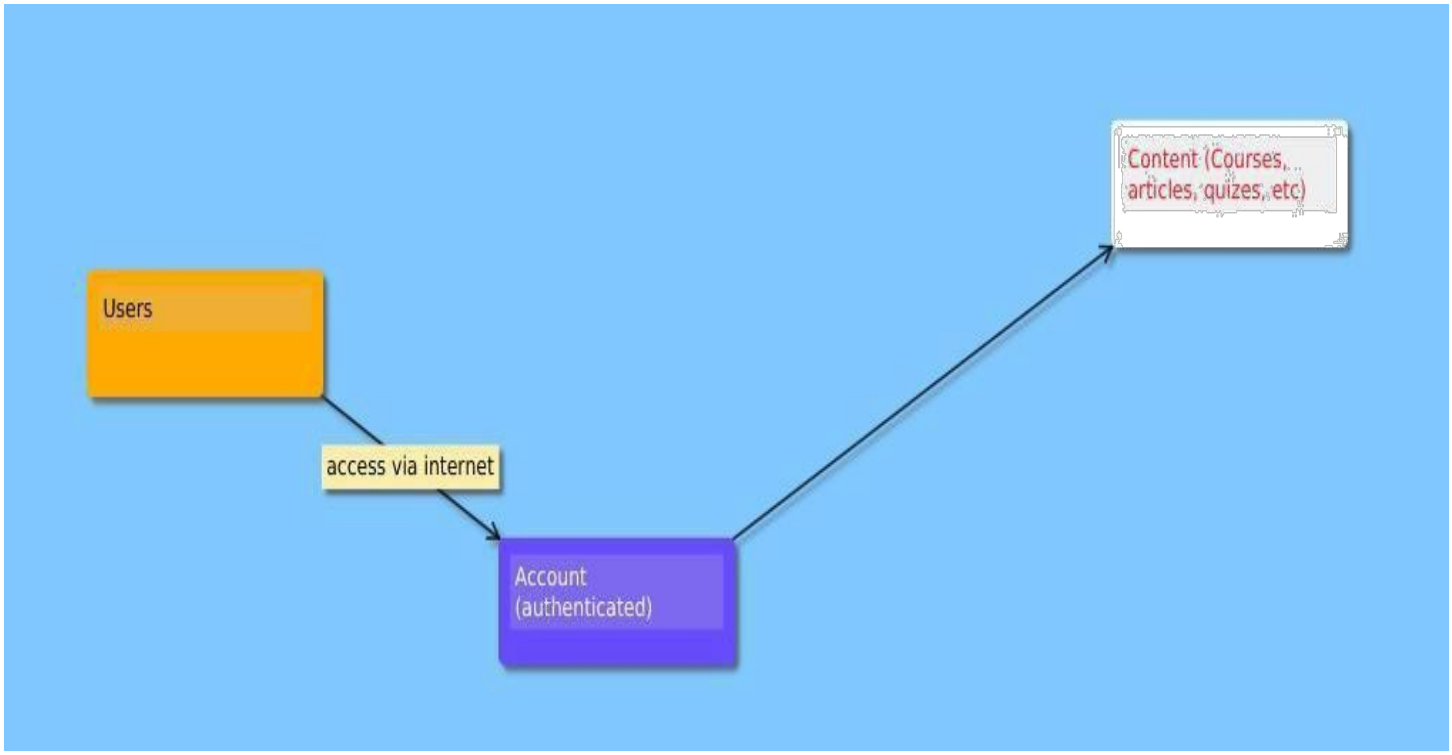
- **Level 0** – Highest abstraction level DFD is known as Level 0 DFD, which depicts the entire information system as one diagram concealing all the underlying details. Level 0 DFDs are also known as context level DFDs.

- **Level 1** – The Level 0 DFD is broken down into more specific, Level 1 DFD. Level 1 DFD depicts basic modules in the system and flow of data among various modules. Level 1 DFD also mentions basic processes and sources of information.

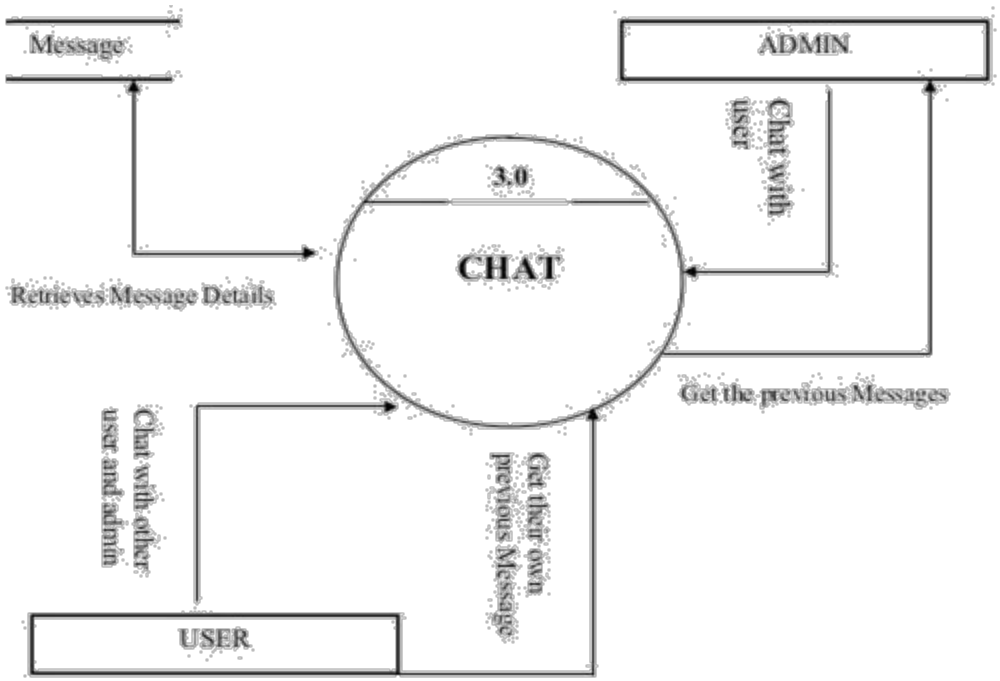
- **Level 2** – At this level, DFD shows how data flows inside the modules mentioned in Level

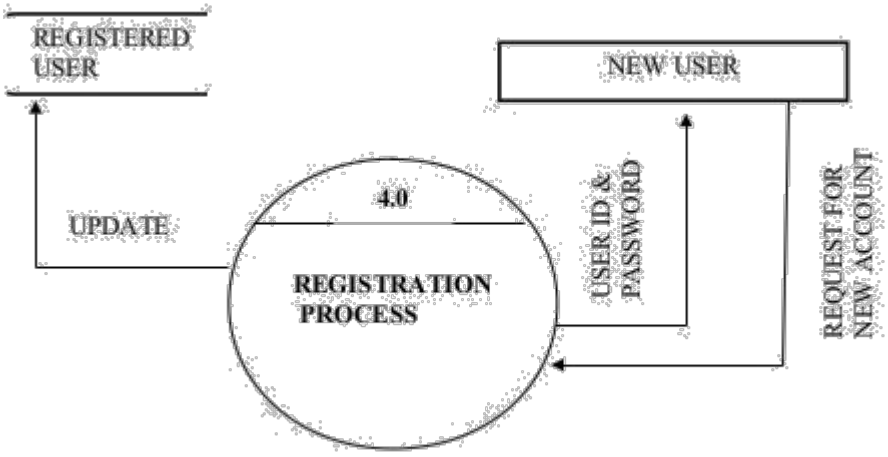
Higher level DFDs can be transformed into more specific lower level DFDs with deeper level of understanding unless the desired level of specification is achieved.

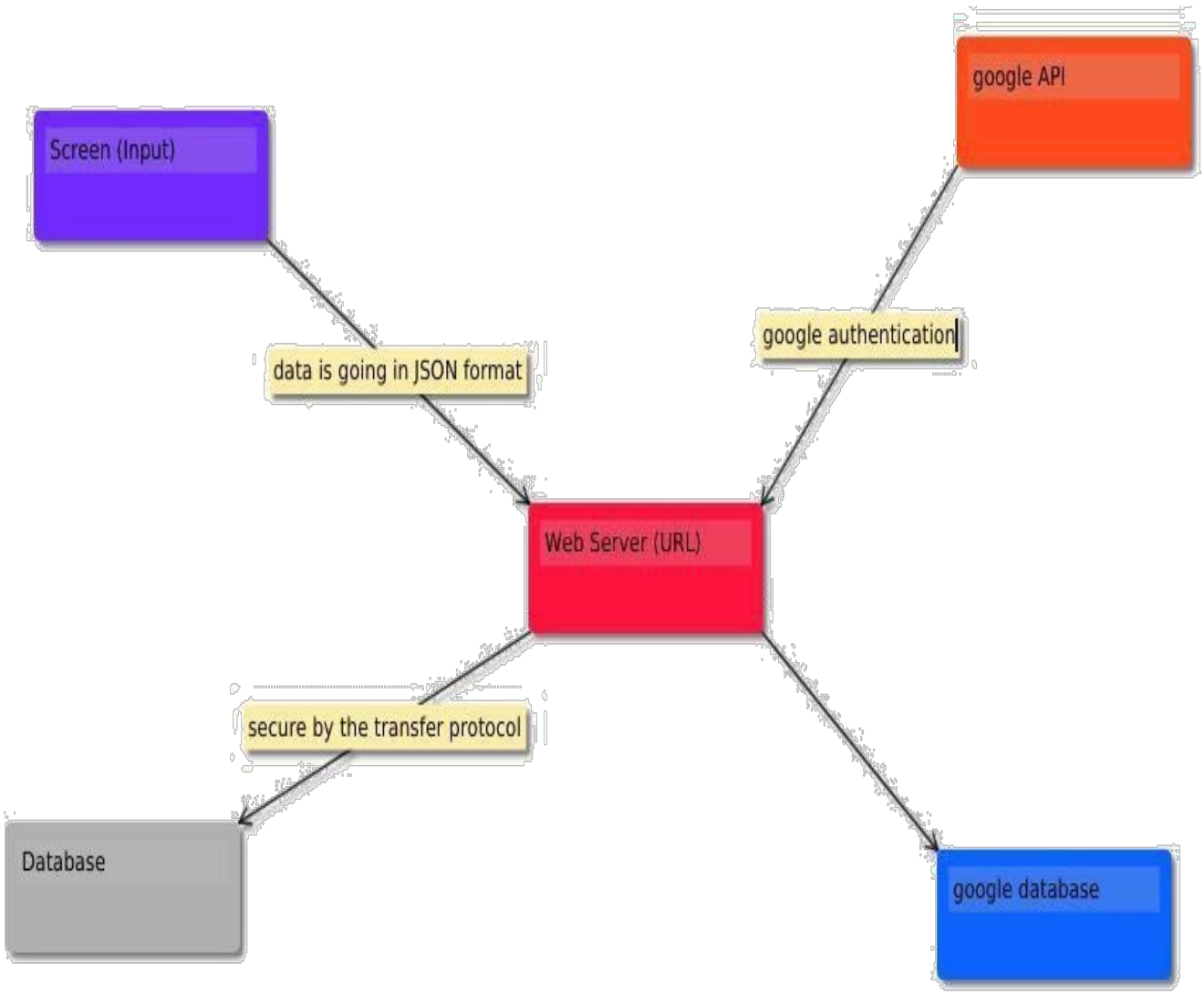
LEVEL-0 DFD:

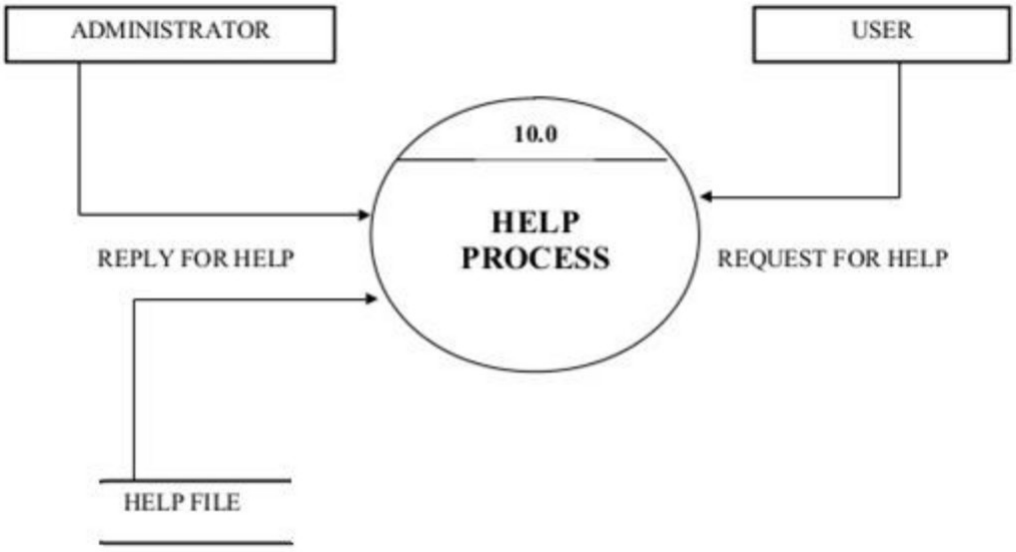


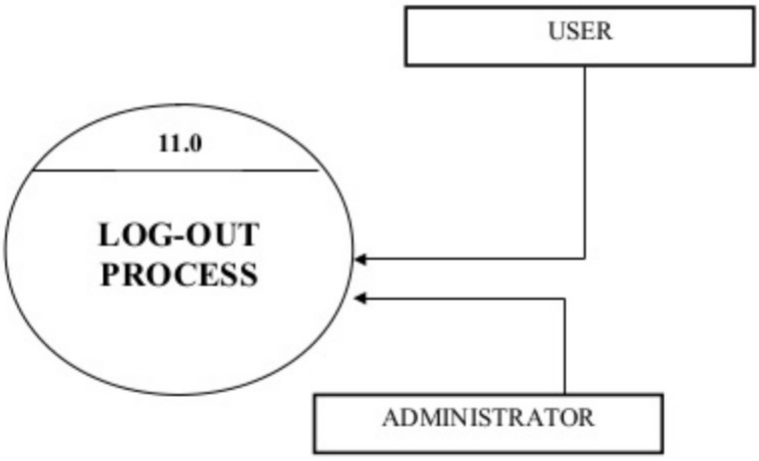
LEVEL -1 DFD:


Message
ADMIN
Chat with user
3.0
CHAT
Retrieves Message Details
Get the previous Messages
Chat with other user and admin
Get their own previous Message
USER


REGISTERED USER
NEW USER
4.0
UPDATE
REGISTRATION PROCESS
USER ID & PASSWORD
REQUEST FOR NEW ACCOUNT


google API
Screen (Input)
google authentication
data is going in JSON format
Web Server (URL)
secure by the transfer protocol
Database
google database


ADMINISTRATOR
USER
10.0
HELP
PROCESS
REPLY FOR HELP
REQUEST FOR HELP
HELP FILE


USER
11.0
LOG-OUT
PROCESS
ADMINISTRATOR

2) – E - R DIAGRAM: Definition:

An entity-relationship (ER) diagram is a specialized graphic that illustrates theInterrelationships between entities in a database.

ERdiagrams often use symbols to represent three different types of information. Boxesare commonly used to represent entities. Diamonds are normally used torepresent relationships and ovals are used to represent attributes.

Entity Relationship (ER) diagrams:

The ER or (Entity Relational Model) is a high-level conceptual data model diagram. Entity-Relation model is based on the notion of real-world entities and the relationship between them. ER modeling helps you to analyze data requirements systematically to produce a well-designed database. So, it is considered a best practice to complete ER modeling before implementing your database.

ER diagrams are a visual tool which is helpful to represent the ER model. It was proposed by Peter Chen in 1971 to create a uniform convention which can be used for relational database and network. He aimed to use an ER model as a conceptual modeling approach.

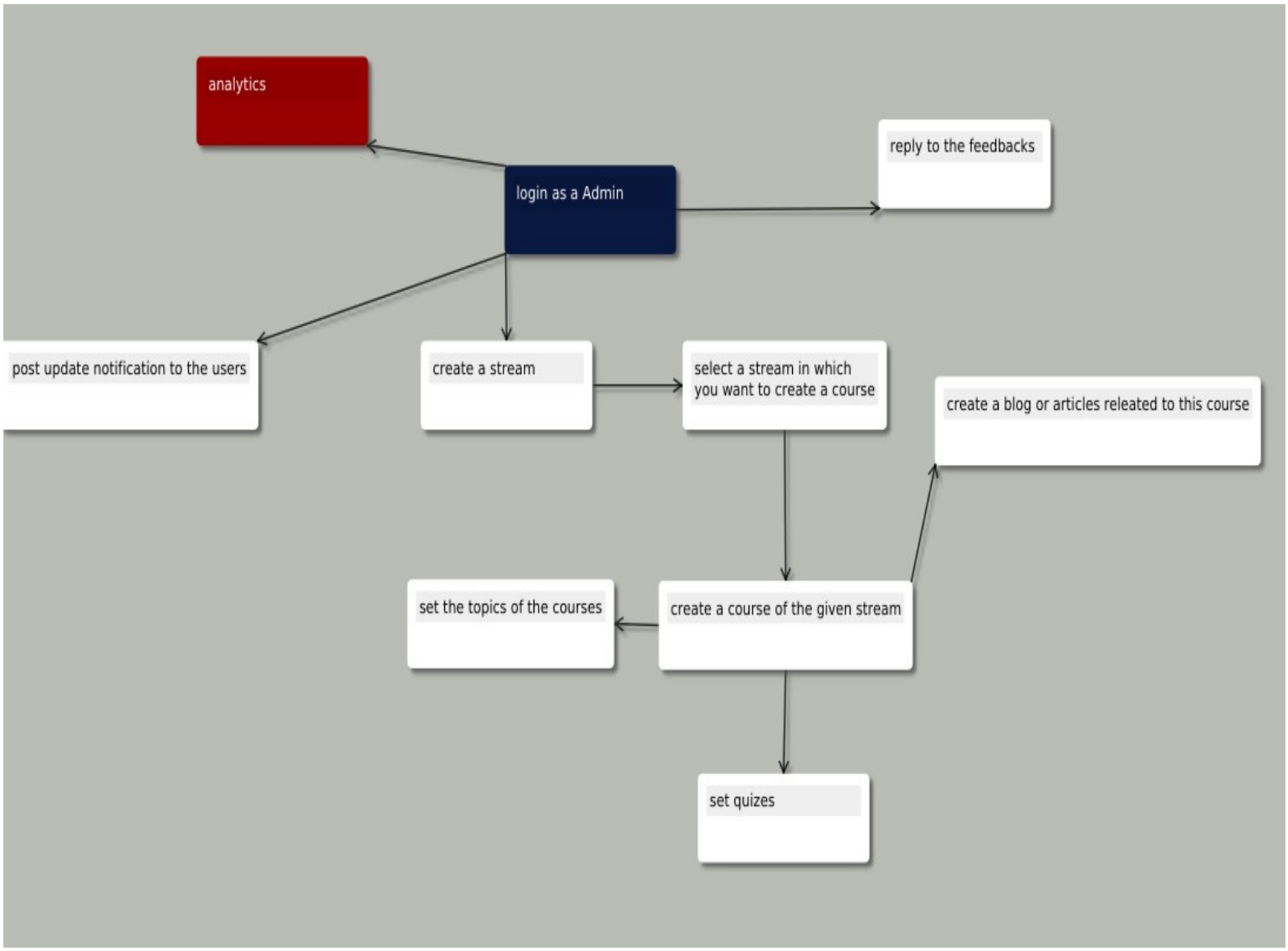
Entity relationship diagram displays the relationships of entity set stored in a database. In other words, we can say that ER diagrams help you to explain the logical structure of databases. At first look, an ER diagram looks very similar to the flowchart. However, ER Diagram includes many specialized symbols, and its meanings make this model unique

This model is based on three basic concepts:

- Entities
- Attributes
- Relationship

3) **– ACTIVITY DIAGRAM:**

Activity Diagram of the Administrator

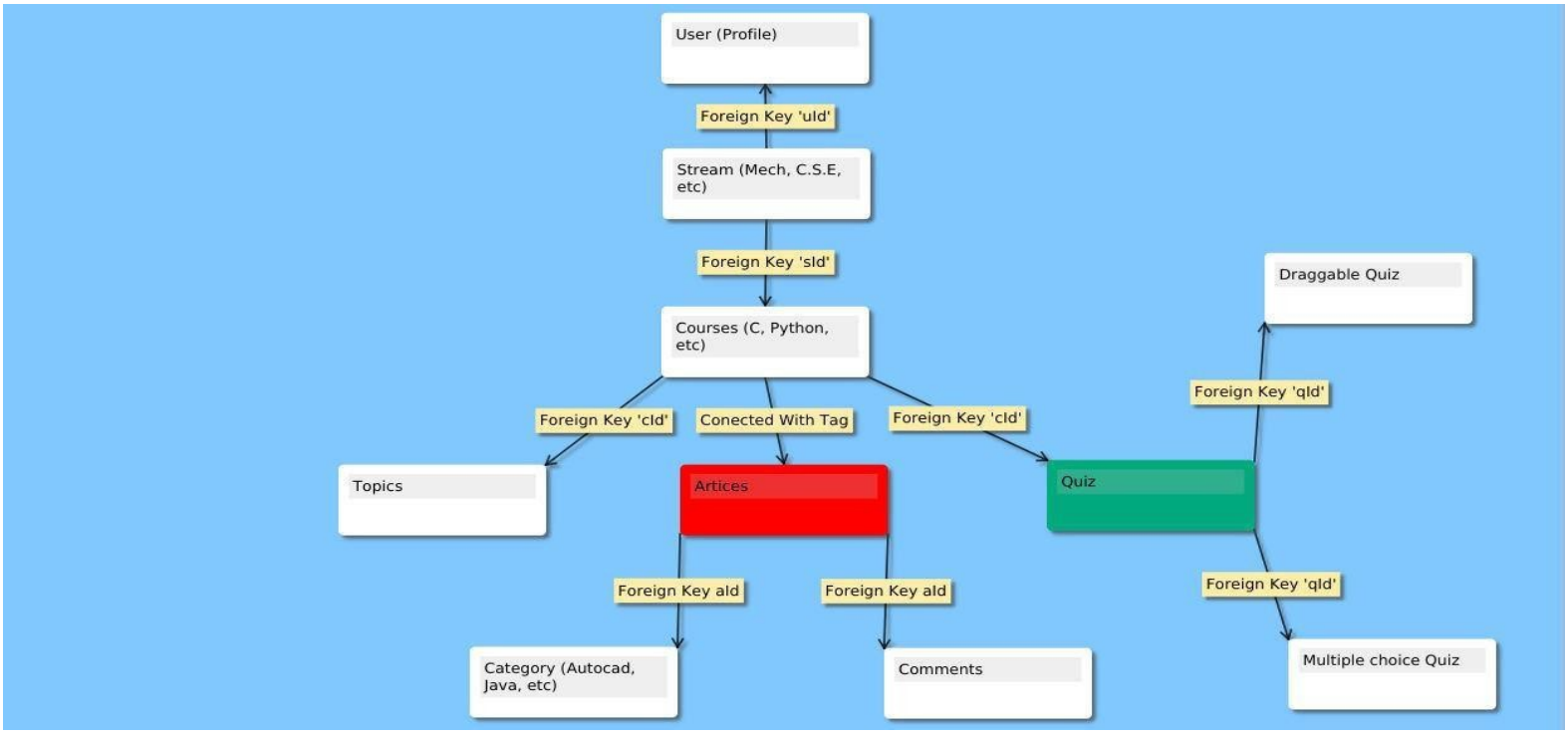


The administrator has all the privilege of accessing all the users feedbacks. He is also authorized to create new accounts for the users, delete the existing users and can also delete the courses, articles, blog, streams etc from

the user accounts. First the administrator should know the password of the server host (i.e. he should be the root user).

The above figure is the activity diagram of the administrator and explains all the specified functionalities in a pictographic view. To establish as an administrator, he should prove his authentication and then is allowed to perform the administration activities.

Activity Diagram of the Users



It is an activity diagram of the general user. First of all, the user encounters a welcome page that prompts him to enter the valid user name (or valid user- id) and the corresponding password. After a successful user login, he is allowed to select his streams from the screen. He then can see the list of the courses from the screen – you can create a blog related to the corse you are learning . He can also organize contents of the different articles and blogs

Apart from creating blogs or article, he is also allowed to save the other user blogs and articles. And set your own quizes for other users and create challenges. Once the user is logged out, he is only allowed through the login page.

4) – STRUCTURE CHART:

A **Structure Chart** (SC) in software engineering and organizational theory is a chart which shows the breakdown of a system to its lowest manageable levels. They are used in structured programming to arrange program modules into a tree. Each module is represented by a box, which contains the module's name. The tree structure visualizes the relationships between modules.

A structure chart is a top-down modular design tool, constructed of squares representing the different modules in the system, and lines that connect them. The lines represent the connection and or ownership between activities and subactivities as they are used in organization charts.

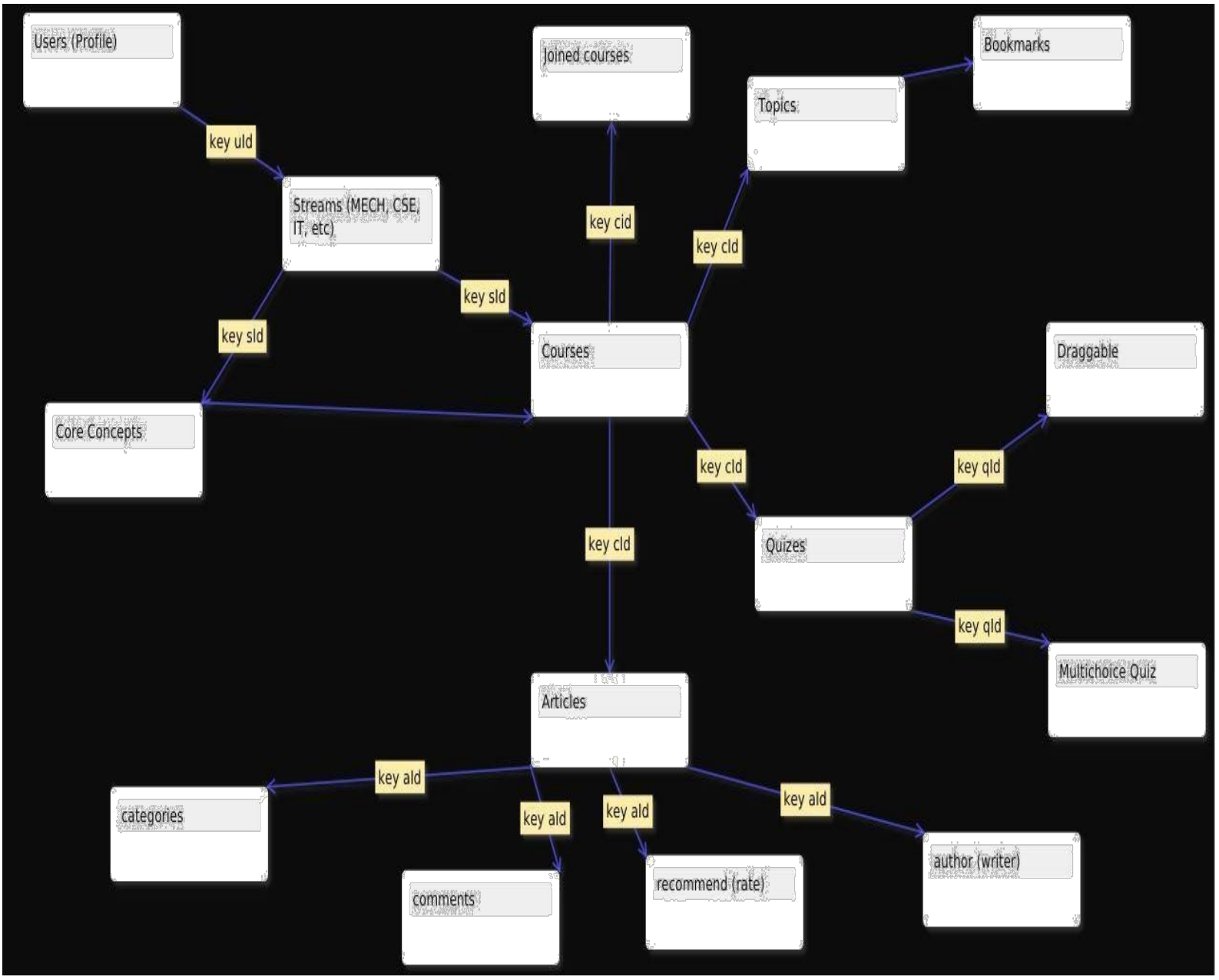
In structured analysis structure charts, according to Wolber (2009), "are used to specify the high-level design, or architecture, of a computer program. As a design tool, they aid the programmer in dividing and conquering a large software problem, that is, recursively breaking a problem down into parts that are small enough to be understood by a human brain. The process is called top-down design, or functional decomposition. Programmers use a structure chart to build a program in a manner similar to how an architect uses a blueprint to build a house. In the design stage, the chart is drawn and used as a way for the client and the various software designers to communicate. During the actual building of the program (implementation), the chart is continually referred to as "the master-plan".

- the size and complexity of the system
- number of readily identifiable functions and modules within each function
- whether each identifiable function is a manageable entity or should be broken down into smaller components.

A structure chart is also used to diagram associated elements that comprise a run stream or thread. It is often developed as a hierarchical diagram, but other representations are allowable. The representation must describe the breakdown of the configuration system into subsystems and the lowest manageable level. An accurate and complete structure chart is the key to the determination of the configuration items (CI), and a visual representation of the configuration system and the internal interfaces among its CIs. During the configuration control process, the structure chart is used to identify CIs and their associated artifacts that a proposed change may impact.

According to Wolber (2009), "a structure chart can be developed starting with the creating of a structure, which places the root of an upside-down tree which forms the structure chart. The next step is to conceptualize the main sub-tasks that must be performed by the program to solve the problem. Next, the programmer focuses on each sub-task individually, and conceptualizes how each can be broken down into even smaller tasks. Eventually, the program is broken down to a point where the leaves of the tree represent simple methods that can be coded with just a few program statements".

The structure chart of intranet mailing system is given in next page.



CHAPTER 7

CHAPTER 7

TESTING

Testing is the process of running a system with the intention of finding errors. Testing enhances the integrity of a system by detecting deviations in design and errors in the system. Testing aims at detecting error-prone areas. This helps in the prevention of errors in a system. Testing also adds value to the product by conforming to the user requirements.

The main purpose of testing is to detect errors and error-prone areas in a system. Testing must be thorough and well-planned. A partially tested system is as bad as an untested system. And the price of an untested and under-tested system is high.

The implementation is the final and important phase. It involves user-training, system testing in order to ensure successful running of the proposed system. The user tests the system and changes are made according to their needs. The testing involves the testing of the developed system using various kinds of data. While testing, errors are noted and correctness is the mode.

1. OBJECTIVES OF TESTING:

Testing is a process of executing a program with the intent of finding errors. A Successful test case is onethat uncoversan as- yet-undiscovered error.

The various types of testing on the system are:

Unit Testing.

Integration Testing

System testing

User Acceptance Testing

➢ Unit Testing:

Unit testing focuses efforts on the smallest unit of software design. This is known as module testing. The modules are tested separately.

The test is carried out during programming stage itself.

In this step, each module is found to be working satisfactory as regards to the expected output from the module.

- ## INTEGRATION TESTING:

Data can be lost across an interface. One module can have an adverse effect on another, sub functions, when combined, may not be linked in desired manner in major functions. Integration testing is a systematic approach for constructing the program structure, while at the same time conducting test to uncover errors associated within the interface. The objective is to take unit tested modules and builds program structure. All the modules are combined and tested as a whole.

- ## SYSTEM TESTING:

System testing is the stage of implementation. This is to check whether the system works accurately and efficiently before live operation commences. Testing is vital to the success of the system. The candidate system is subject to a variety of tests: on line response, volume, stress, recovery, security and usability tests. A series of tests are performed for the proposed system is ready for user acceptance testing.

- ## USER ACCEPATANCE TESTING:

User acceptance of a system is the key factor for the success of any system. The system under consideration is tested for the user acceptance by constantly keeping in touch with the prospective system users at the time of developing and making changes whenever required.

2. VALIDATION:

At the culmination of the integration testing, Software is completely assembled as a package. Interfacing errors have been uncovered and corrected and a final series of software test begin in validation testing. Validation testing can be defined in many ways, but a simple definition is that the validation succeeds when the software functions in a manner that is expected by the customer. After validation test has been conducted, one of the three possible conditions exists.

The function or performance characteristics confirm to specification and are accepted.

A deviation from specification is uncovered and a deficiency lists is created.

Proposed system under consideration has been tested by using validation test and found to be working satisfactory.

3. OUTPUT TESTING

After performing the validation testing, the next step is output testing of the proposed system, since no system could be useful if it does not produce the required output in a specific format. The output format on the screen is found to be correct; the format was designed in the system design time according to the user needs. For the hard copy also; the output comes as per the specified requirements by the user. Hence output testing did not result in any correction for the system.

4. TEST CHARACTERS:

➢ *BLACK BOX TESTING*:

The method of Black Box Testing is used by the software engineer to derive the required results of the test cases:

1. Black Box Test
2. A Black Box test for the internal logic structure of the software.
3. Black box testing was performed to find errors in the following categories:-

- Incorrect or missing functions
- Graphics error.
- Errors in data in binary format.
- Error in data in integer format.
- File error.
- Pointer error.
- Memory access error.
- Variable error.
- Performance error

➢ *WHITE BOX TESTING:*

White Box methods the software engineer can derive the following test cases:

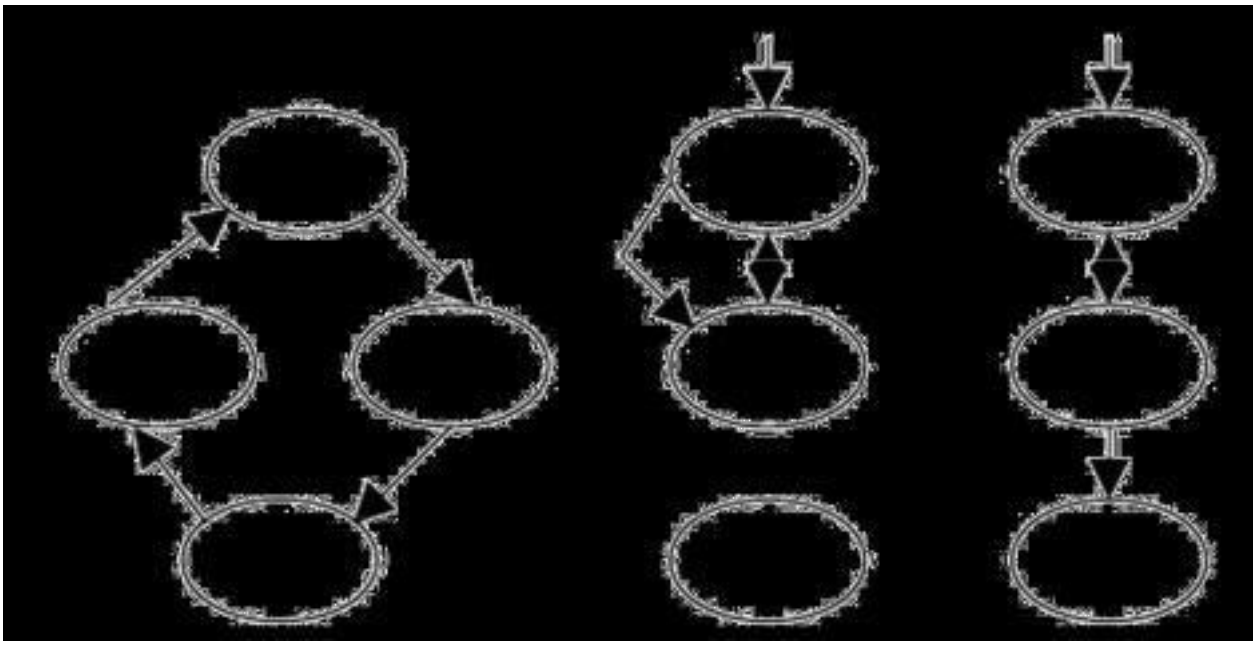
1. Optimizing the code
2. Finding hidden errors

In White Box Testing efforts were made to handle the following:-

- Number of input parameters equal to number of arguments.
- Parameters and arguments attributes match.
- Any references to parameters not associated to build in functions correct.
- Input only arguments altered.
- Global variable definition consistent across module.
- Files attributes correct.
- Format specifications matches I/O specification.
- Files opened before use.
- File closed while working is going on.
- I/O errors handled.
- Any textual errors in output information.

5.TEST DATA IMPLEMENTATION AND THEIR RESULT ON:

The quality depends truly on the various predefined testing norms and on the performances of the software over those norms. There are various standards existing in the software industry the engineered end product strives to achieve viz. ISO 9002 SEI CMM Level5 etc. These s t a respective testing norms predefined in them vide the various test cases and parameters using the CASE topologies. Generally, software is tested both on a stand-alone mode as well afterintegrating all the modules in the system vide deferent available testing methods/norms.



FLOW GRAPH

The following Flow Graph methodology was used while testing the software:

Here each circle represents one or more non branching procedural language or source codes methodology was selected. While performing Loop Testing simple loops, concatenated loops, nested and unstructured loops were tested thoroughly.

➢ TEST CASES:

• LOGIN:

Sr. No.	Input Values	Test case	Conditional being checked	Result
1	UserID	Empty	Please input data values	Successful
2	UserID	Wrong UserID	Wrong userID or Password	Successful
3	Password	Empty	Please input data values	Successful
4	Password	Ifwrong Password	Wrong userID or Password	Successful

6. SYSTEM TESTING:

System testing is a critical element of quality assurance and represents the ultimate review of analysis, design and coding. Test case design focuses on a set of techniques for the creation of test because that meet overall testing objective. When a system is developed it is hoped that it performs properly. The main purpose of testing an information system is to find the errors and correct them.

System testing is the process of checking whether the developed system is working according to the objective and requirement. All testing is to be conducted in accordance to the test conditions specified earlier. This will ensure that the test coverage meets the requirements and that testing is done in a systematic manner.

The process of analyzing the software item to detect the differences between existing or required condition and evaluate the features of the software items. The thorough testing of the system before release of the software needs to be done vide the various test cases and modes so that the software becomes devoid of bugs and uses minimum space requirements as well as minimum time to perform. The test cases were selected beforehand with expected results defined and actual results recorded for comparison. The selection of test cases is done by

“White Box Testing” technique to check the internal programming logic and efficiency and vide ”Black Box Testing” technique to check software requirement fulfillment with intension of finding maximum number of errors with minimum effort and time. Although test cases are a design by considering the cyclomatic complexity, conditional test, still the software code is not in its optional form, as all other possible alternative parts in the software are not considered. At the integration level, the software will be passing to the third party tests which would further enhance the software optimality and efficiency

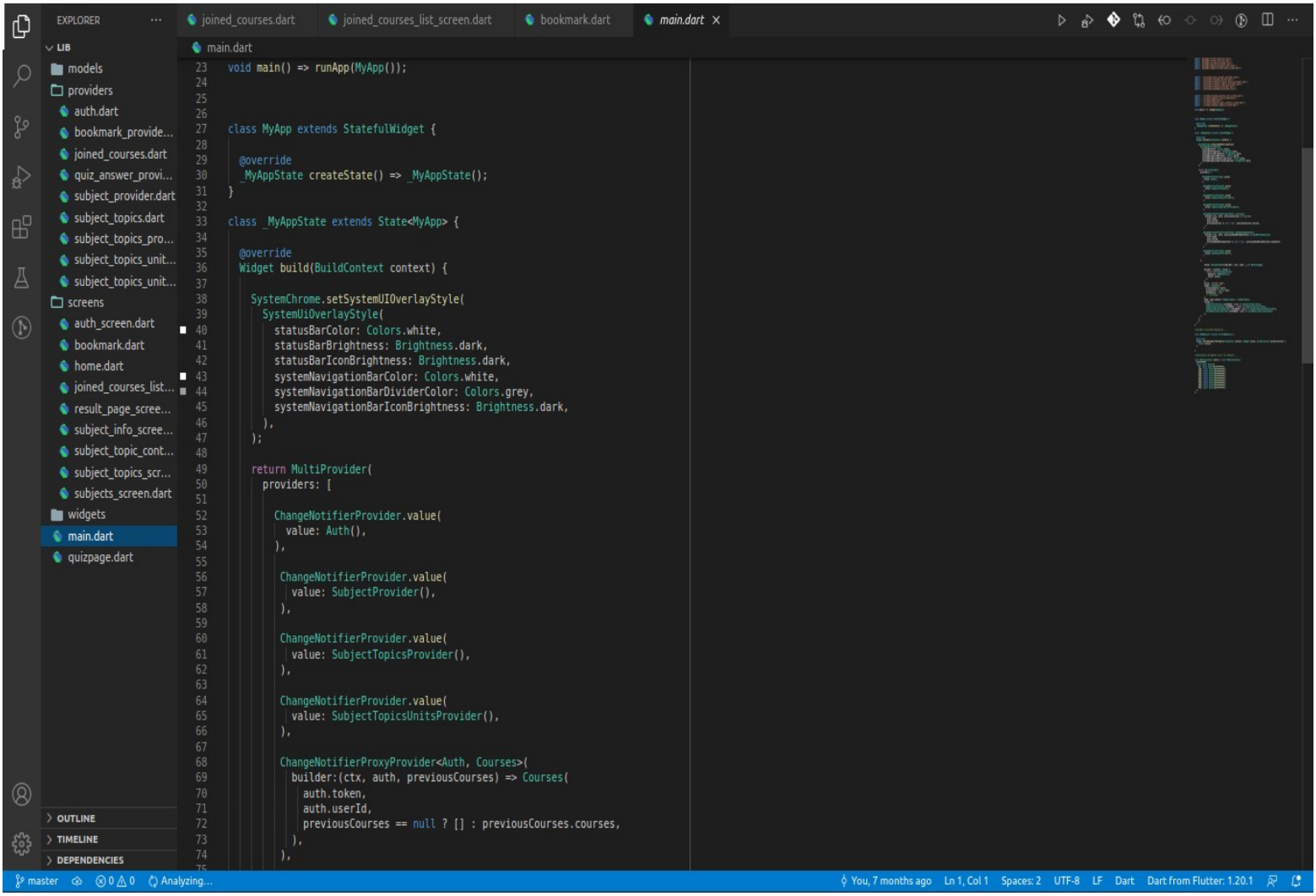
CHAPTER 8

CHAPTER 8

SNAPSHOTS

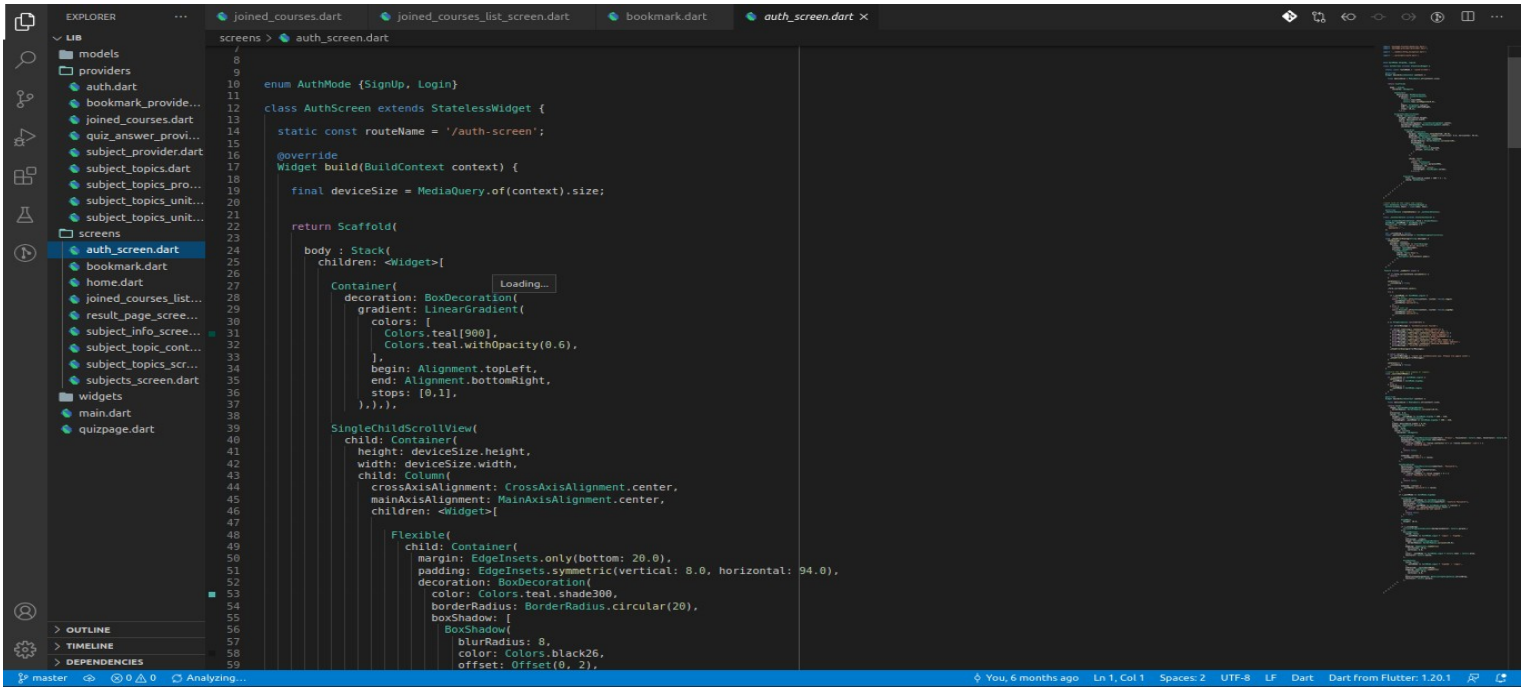
1)- SOURCE CODE SNAPSHOTS

These are main source code of the given application:



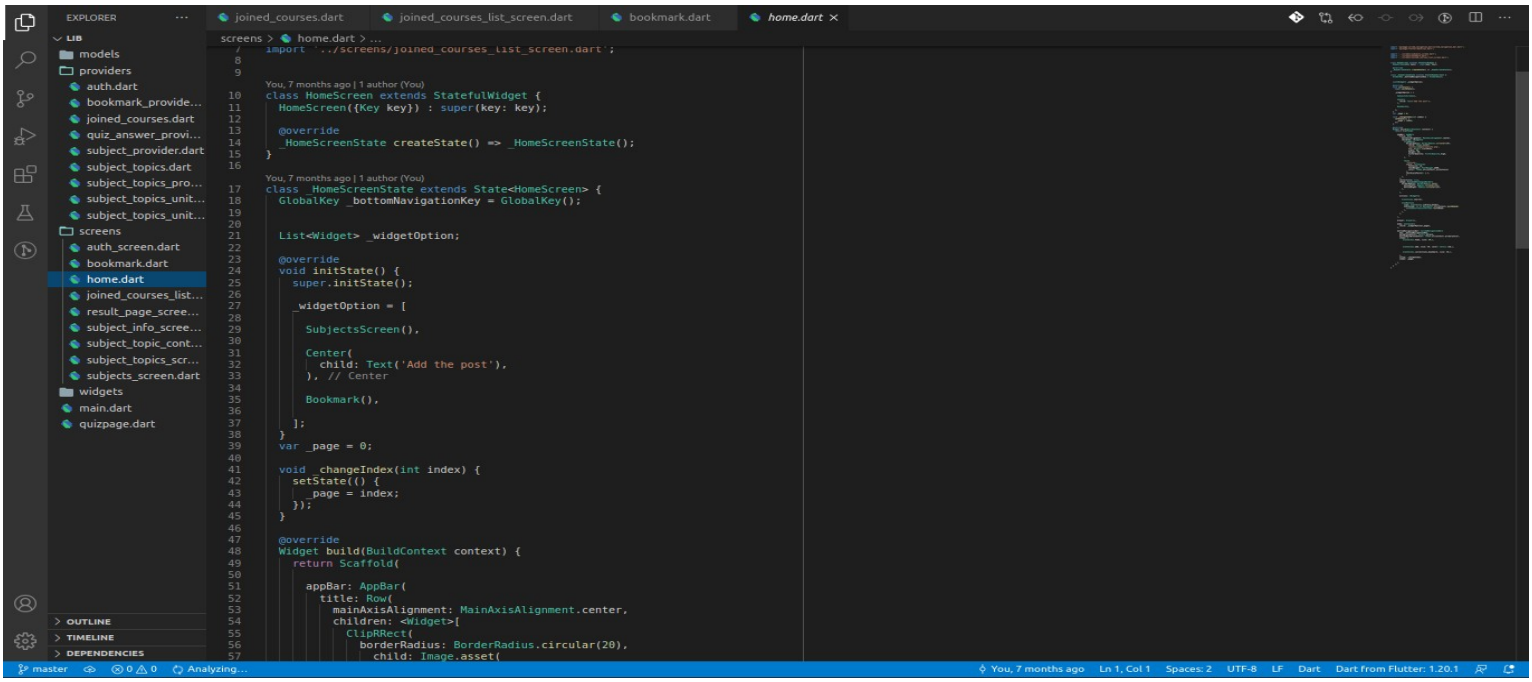
Main.dart file:

1- Auth.dart file:



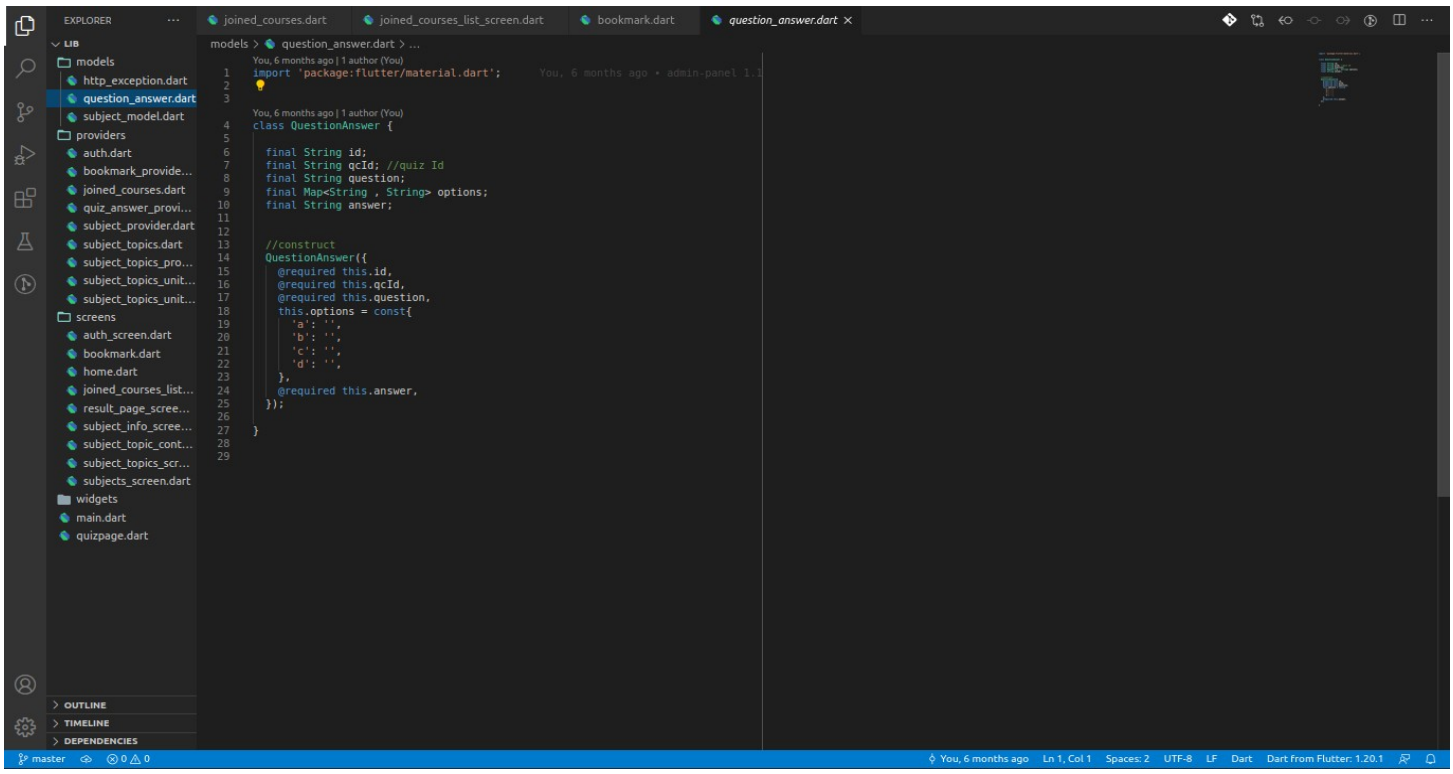
```dart

enum AuthMode {SignUp, Login}

class AuthScreen extends StatelessWidget {

  static const routeName = '/auth-screen';

  @override
  Widget build(BuildContext context) {

    final deviceSize = MediaQuery.of(context).size;


    return Scaffold(

      body : Stack(
        children: <Widget>[

          Container(                Loading...
            decoration: BoxDecoration(
              gradient: LinearGradient(
                colors: [
                  Colors.teal[900],
                  Colors.teal.withOpacity(0.6),
                ],
                begin: Alignment.topLeft,
                end: Alignment.bottomRight,
                stops: [0,1],
              ),),),

          SingleChildScrollView(
            child: Container(
              height: deviceSize.height,
              width: deviceSize.width,
              child: Column(
                crossAxisAlignment: CrossAxisAlignment.center,
                mainAxisAlignment: MainAxisAlignment.center,
                children: <Widget>[

                  Flexible(
                    child: Container(
                      margin: EdgeInsets.only(bottom: 20.0),
                      padding: EdgeInsets.symmetric(vertical: 8.0, horizontal: 94.0),
                      decoration: BoxDecoration(
                        color: Colors.teal.shade300,
                        borderRadius: BorderRadius.circular(20),
                        boxShadow: [
                          BoxShadow(
                            blurRadius: 8,
                            color: Colors.black26,
                            offset: Offset(0, 2),
```

2- Home.dart file:



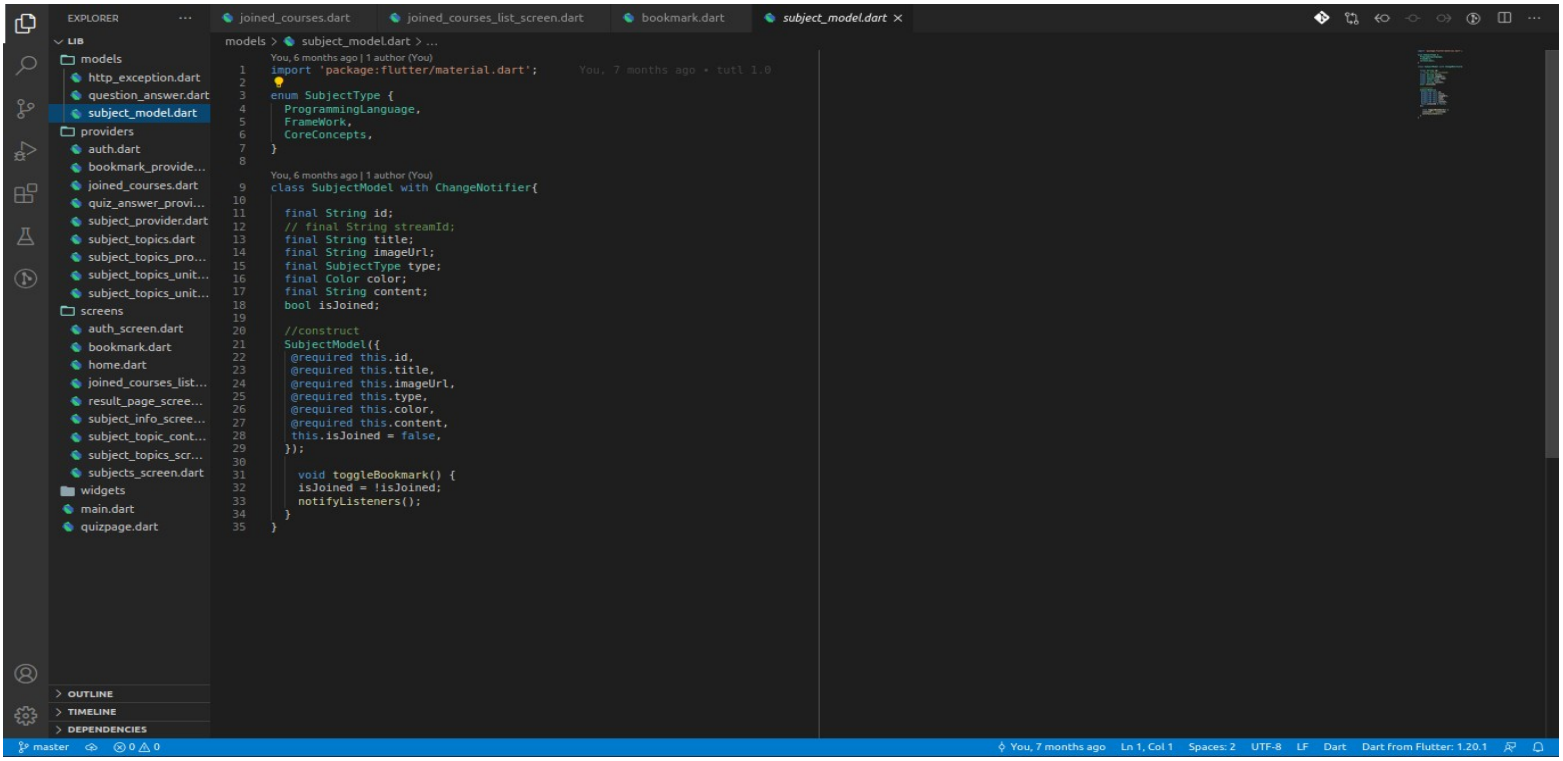
```dart
import '../screens/joined_courses_list_screen.dart';


You, 7 months ago | 1 author (You)
class HomeScreen extends StatefulWidget {
  HomeScreen({Key key}) : super(key: key);

  @override
  _HomeScreenState createState() => _HomeScreenState();
}

You, 7 months ago | 1 author (You)
class _HomeScreenState extends State<HomeScreen> {
  GlobalKey _bottomNavigationKey = GlobalKey();


  List<Widget> _widgetOption;

  @override
  void initState() {
    super.initState();

    _widgetOption = [

      SubjectsScreen(),

      Center(
        child: Text('Add the post'),
      ), // Center

      Bookmark(),

    ];
  }
  var _page = 0;

  void _changeIndex(int index) {
    setState(() {
      _page = index;
    });
  }

  @override
  Widget build(BuildContext context) {
    return Scaffold(

      appBar: AppBar(
        title: Row(
          mainAxisAlignment: MainAxisAlignment.center,
          children: <Widget>[
            ClipRRect(
              borderRadius: BorderRadius.circular(20),
                child: Image.asset(
```

3- QuestionAnswer.dart file model:



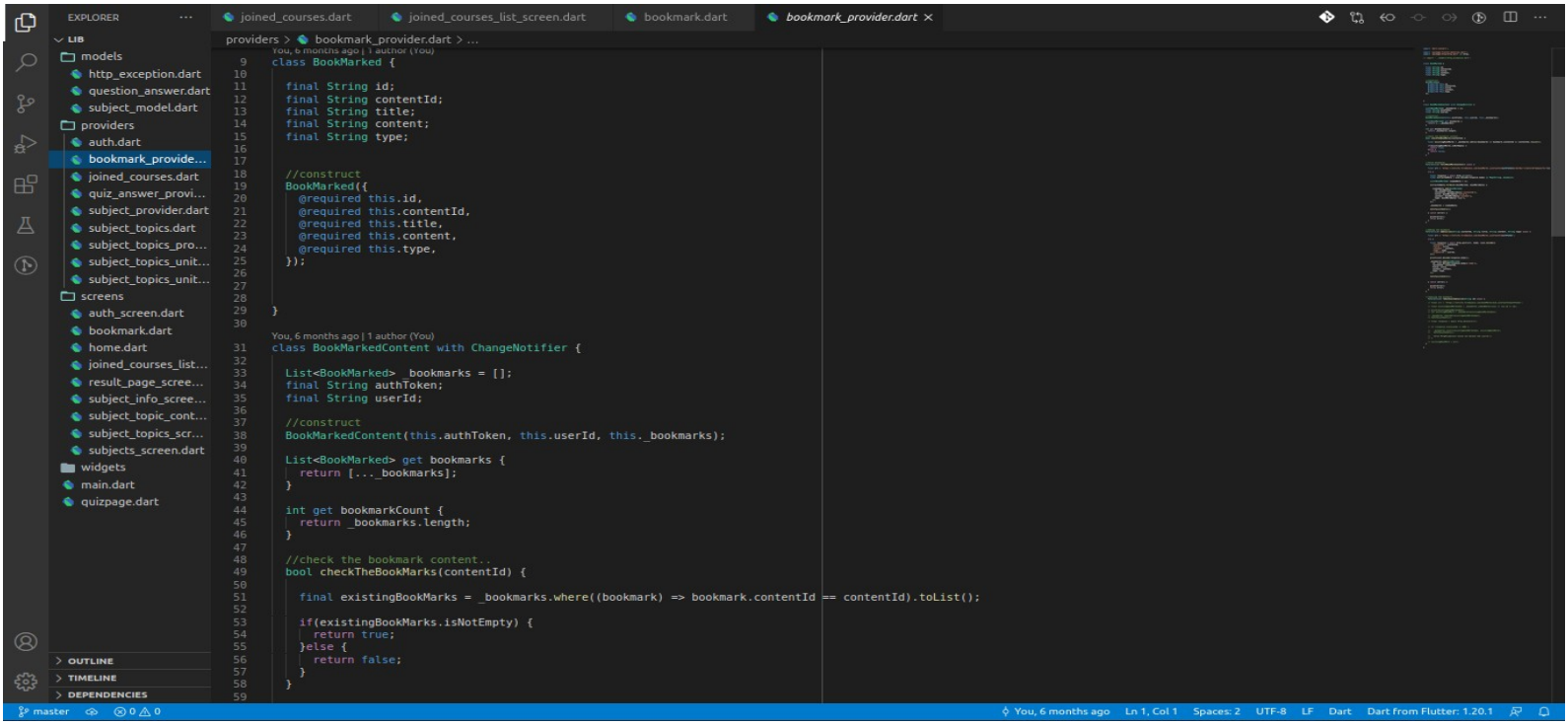
```dart
import 'package:flutter/material.dart';

class QuestionAnswer {

  final String id;
  final String qcId; //quiz Id
  final String question;
  final Map<String , String> options;
  final String answer;


  //construct
  QuestionAnswer({
    @required this.id,
    @required this.qcId,
    @required this.question,
    this.options = const{
      'a': '',
      'b': '',
      'c': '',
      'd': '',
    },
    @required this.answer,
  });

}
```

4 – CourseType.dart file model:

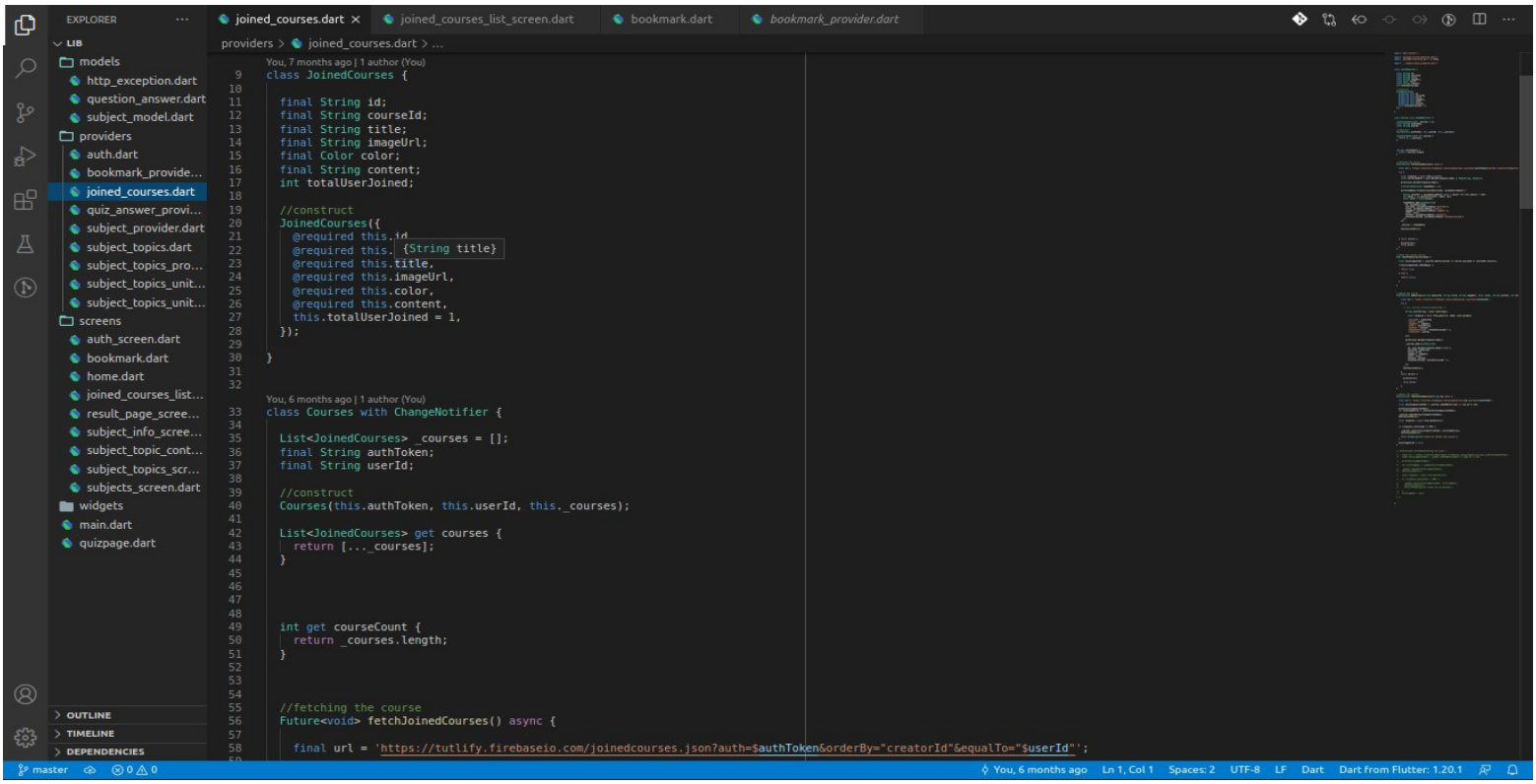


```dart
import 'package:flutter/material.dart';

enum SubjectType {
  ProgrammingLanguage,
  FrameWork,
  CoreConcepts,
}

class SubjectModel with ChangeNotifier{

  final String id;
  // final String streamId;
  final String title;
  final String imageUrl;
  final SubjectType type;
  final Color color;
  final String content;
  bool isJoined;

  //construct
  SubjectModel({
    @required this.id,
    @required this.title,
    @required this.imageUrl,
    @required this.type,
    @required this.color,
    @required this.content,
    this.isJoined = false,
  });

    void toggleBookmark() {
    isJoined = !isJoined;
    notifyListeners();
  }
}
```

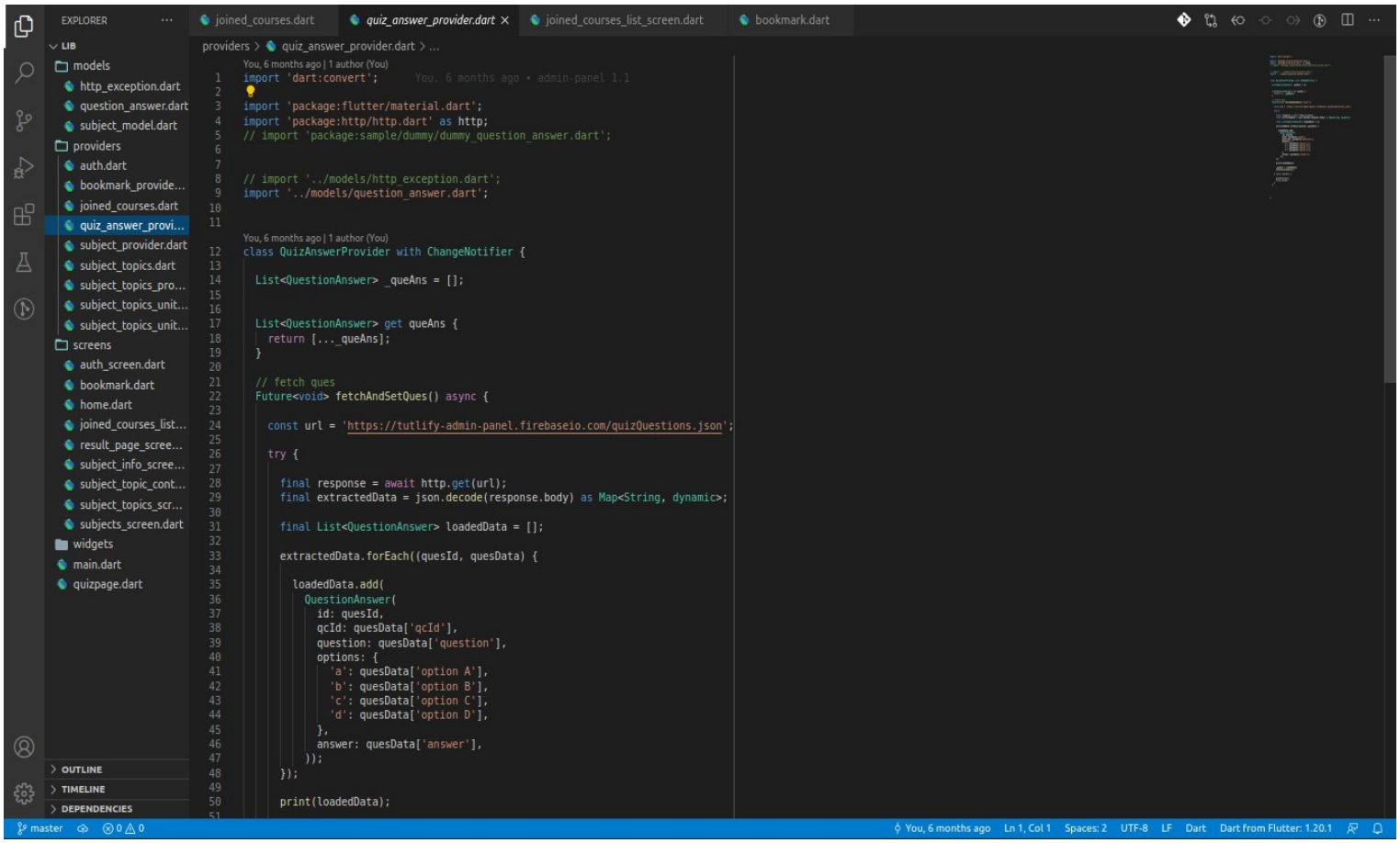
5 – Bookmark.dart file:



6 – JoinedCourses.dart file :



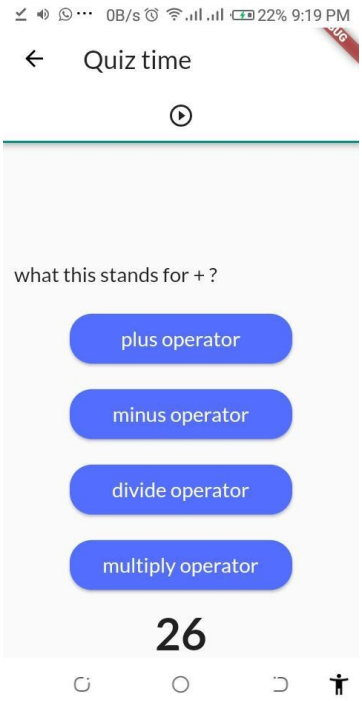
7- Quiz_provider.dart file:



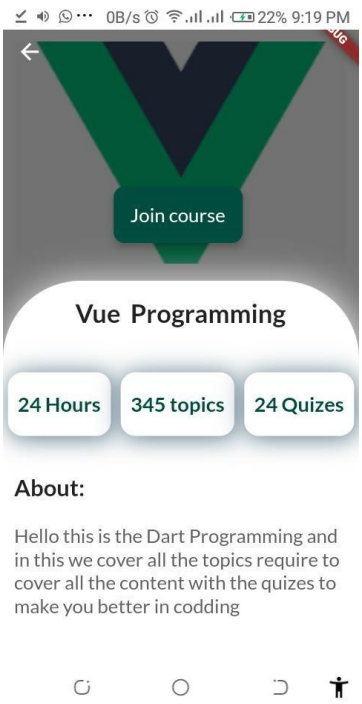
2)-PROJECT SNAPSHOT:

HOME PAGE:

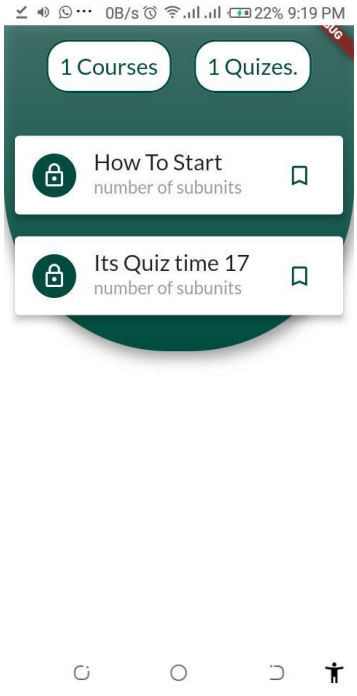
QUIZ PAGE:



COURSE PAGE:



COURSE TOPICS AND QUIZES PAGE:



REFERENCES:

1. For RFC2060, https://www.rfc-archieve.org.
2. For **FLUTTER**, https://flutter.io
3. About **FIREBASE**, https://firebase.com.
4. **FLUTTER tutorial**: https://flutter.dev/docs
5. **GITHUB** : https://github.com/egnimos
6. **PROJECT SOURCE CODE**: https://github.com/egnimos/sample-tutl-
7. **JSON (JAVASCRIPT OBJECT NOTATION)**: https://www.json.org/json-en.html