LAB REPORT

Submitted by

Dev Sharma,Sowndarya,Vishnu
MONIKA MARADANA- RA2111033010158

Under the Guidence of

Prof Siva R

Assistant Professor, Department of Data Science and Business Systems In partial satisfaction of the requirements for the degree of

BACHELOR OF TECHNOLOGY

in

COMPUTER SCIENCE ENGINEERING
with specialization in Data Science

SCHOOL OF COMPUTING

COLLEGE OF ENGINEERING AND TECHNOLOGY
SRM INSTITUTE OF SCIENCE AND TECHNOLOGY
KATTANKULATHUR - 603203

MAY 2023

BONAFIDE CERTIFICATE

Dev Sharma,Sowndarya,Vishnu

Register No. RA21110330100158 Certified to be the Bonafide work done by MONIKA MARADANA of II Year/IV Sem B.Tech Degree Course in the Practical Software Engineering and Project Management 18CSC206J in SRM INSTITUTE OF SCIENCE AND TECHNOLOGY, Kattankulathur during the academic year 2022 – 2023.

LAB INCHARGE
Prof Siva R
Assistant Professor
Department of Computing Technologies
SRMIST – KTR

Head of the Department
Dr. Annie Uthra
Professor and Head,
Department of Computing Technologies
SRMIST-KTR

Date : 15/05/2023

INDEX

ABSTRACT

This abstract introduces a smart fitness tracking app that revolutionizes health monitoring. Leveraging mobile devices and wearables, the app collects vital health metrics, such as heart rate and activity levels. Its user-friendly interface presents personalized dashboards with progress charts and goals, while AI algorithms generate tailored workout recommendations. Social features encourage community engagement and adherence to healthy habits. The app prioritizes privacy and data security, empowering individuals to take control of their fitness goals in a modern, convenient manner.

Abstract:

This application introduces a smart fitness tracking app that revolutionizes health monitoring. Leveraging mobile devices and wearables, the app collects vital health metrics, such as heart rate and activity levels. Its user-friendly interface presents personalized dashboards with progress charts and goals, while AI algorithms generate tailored workout recommendations. Social features encourage community engagement and adherence to healthy habits. The app prioritizes privacy and data security, empowering individuals to take control of their fitness goals in a modern, convenient manner. With its comprehensive features and user-centric approach, the smart fitness tracking app is poised to transform the way people monitor and improve their well-being.

PROBLEM STATEMENT

Despite the growing interest in fitness and well-being, individuals face challenges in effectively monitoring and improving their health due to the lack of a comprehensive and user-friendly solution. Existing fitness tracking apps often lack personalized guidance, social engagement features, and robust privacy measures. There is a need for a smart fitness tracking app that revolutionizes health monitoring by seamlessly integrating with users' lifestyles, leveraging advanced technologies, fostering community engagement, and prioritizing privacy and data security. This app aims to empower individuals to take control of their fitness goals, optimize their routines, and enhance their overall well-being in a modern and convenient manner.

The Smart Fitness Trainer App

THE PROJECT

In bullet points, describe the problem this project aims to solve or the opportunity it aims to develop.

> 75-80 % of students tried to get into fitness and workout but do not possess any prior knowledge of what to do and what to eat.

> To solve this problem our paper proposes an application that gives the correct data on what things you should eat and what workouts you can do.

> This application asks the user their information about their weight, height, and their routines to perfectly create their own personalized Workout schedule.

> This application will hold numerous workout schedules and plans and select the correct one which is suitable to the user.

> And it will recommend the best food you can eat and even provide the online link to buy them.

THE HISTORY

In bullet points, describe the current situation.

> Looking back we have couple of fitness based applications But the problem is it does not understand the users timing and schedule.

>The current situation of having a perfect healthy diet based foods are bread ,salad,or other western based foods.

>We need an application on the foods based on Indian culinary.

standards.

> Adding the links to every food item that we present in the application without any connection of the online grocery store may create an loss for us

>The food on the website might not be budget friendly

APPROACH

List what is needed to complete the project.

>A strong Computer Language to write this app such as C++/Python/Java.
>Data about various workouts.
>Online Links of Healthy and Nutrient food.
>A platform to sell this application such as (play store,app store,TapTap)
>Report system to find any bugs.

BENEFITS

In bullet points, list the benefits that this project will bring to the organization.

> A user friendly app that creates your own personal schedule that you can follow. > Maintaining a very healthy body and meals which is hard for teenagers can be achieved.> Easy to use since its intended to be on your smartphone.
>links to the food page are provided so you don't have to search manually every time.

Selection of Methodology

- **< Summarize their understanding of "Waterfall" or "Agile" Methodology> • We went with the waterfall methodology because it is perfect for app development .**
- **The building of the application starts by COMMUNICATION ,which gathers the neccesary data and information about workouts and schedules.**
- **And even more suggestions and features are planned and removed as per the groundwork goes.**
- **The next phase of the development is PLANNING which is how and when we should develop the app and release it to the masses**

- **The core planning is how the app should look and how user friendly it is to the user.**
- **Our project enters next crucial phase MODELING which is how the UI and the body of the app and style and color will takes place**
- **AS per now we got our data and information the application can be constructed by the means of PYTHON programming language .**
- **The work is done and followed by the waterfall methodology and the project will be published as per the schedule.**

FIG 1: Stakeholder table

Stakeholder Name	Activity/ Area /Phase	Interest Influence	Priority (High/ Medium/ Low)
SFTA technologies	Fitness / Online platform	high high	high
Play store	publishing area	high high	high
youtube	Advertisement	high high	high
Appstore	publishing area	high high	high
Monika Maradana, Soundharya,dev,vishnu	medium	high high	high
	consumers	high high	high
Teenage audience			

System Requirements

> **Modern-android/ios devices: A** modern day mobile is sufficient for the app to run smoothly

> **Machine language:** The app wil be developed and carried on by the new PYTHON programing language.

> **Server room:** Need sufficient amount server rooms with active cloud data base to register the users data such as users timings in order to provide and create the perfect timing for them.

Functional Requirements

> **Integrated bug report :** A bug report feature should be added in order to solve any internal bug that affects the application, this must be the topmost priority. > **Anonymous:** Need to be secret and secure with the users personal data without any data leak.

> **Large amount of workout data:** The application should have plenty of workout plans to give users flexible reports and plans.

> **User Friendly UI:** The user interface and flow of the application should not to be complicated and need to be easy and simple with eye pleasing color with bright icons to identify the options

>**Ties with the online websites:** In order to redirect to the online websites for recommending correct food plans it needs to be provided with the correct Webpage link for the products.

Non-Functional Requirements

> **Minimalistic :** The app needs to use required data and RAM usage inorder to run and not to take too much RAM.

>**Privacy:** The app should ask the user to access in order respect the users privacy to store temp files and cache.

>**Language support:** The app should support fair amount of languages in order to user friend.

> The chance of this application does not show the workout routines if the users schedule too low to the app

>Project Management Plan

Describe the key issues driving the project.

Focus Area Details	
Scope Management	• A project scope statement provides a detailed description of the work that must be done to deliver the output of a project on time and within the allotted budget. • Required workouts are required to build schedule , so a large amount of all manner of workouts are • need in the data understanding the users time and schedule and building the perfect workout routine for the to act do the coding must take much time to create th desire output d
Risk Management	• Potential issues might harm cost . • Technical Issues of the project and quality of ou software device.

Stak
ehol
der

Our stakeholders are Smart fitness tracking applcation wlth pplay store /App store/windows

Step 1: Identify your stake.
Step 2: Prior“rtize your stakeholders. Next prioritize your stakeholders by assessing their level of influence and level of interest. ..
Step 3: Understand your key stakeholders. Stakeholder engagement Is the systematic identification, analysis, planning and implementation of actions designed to influence stakeholder. A stakeholder engagement strategy Identifies the needs of key groups and the sponser plays a-vital role in-ensuring those business needs are met

>Effort and Cost Estimation

Activity Description	Sub-Task	Sub-Task Description	Effort (in hours)	Cost in INR
Design the user screen	designing ui/ux	UI/UXinteract with a product, such as buttons, icons, menu bars, typography, colors, and more	40-45hrs	RS-2000
	codes for frontend and backend	FrontendHTML,CSS,J avascript Backend-Python, NODE js	60-75 hrs	RS-7000
database management , cloud deployment.		will manage data required like music files and deploying the website on cloud for easy access	40-45hrs	RS 10000

Effort (hr)	Cost (INR)
3	450

>Infrastructure/Resource Cost [CapEx]

Infrastructure Requirement	Qty	Cost per qty	Cost per item
Mobile	-	-	-
Cloud for deployment	1	4,00,000	4,00,000
Data centers	1	13,00,000	13,00,00

>Maintenance and Support Cost [OpEx]

Category	Details	Qty	Cost per qty per annum	Cost per item
People	Network, System, Middleware and DB admin Developer , Support Consultant	3	2,000,000	6,000,000
License	Operating System Database Middleware IDE	10	10000	100,000
Infrastructures	Server, Storage and Network	20	20000	400,000

WORK BREAKDOWN STRUCTURE

- **GROUND WORK**
  - Appointing an lead and organizing a program team and data collection team to build the application
  - properly split and define the work load as per the project criteria
  - proper website page and adversities must be done according to the project scope
  - Assess feasibility.

- **PLANNING**
  - analyzing the risk factors
  - schedule and deadline planning
  - budget and brochure
  - report center

- **EXECUTION**
  - programming source codes
  - tracking the progress
  - find problems and improve the process
  - documentation

- **CONTROL**
  - Control scope screep
  - change management processes
  - performing tracking
  - quality management

❖ budget control

❖

❖ **CLOSURE**

❖ patches update

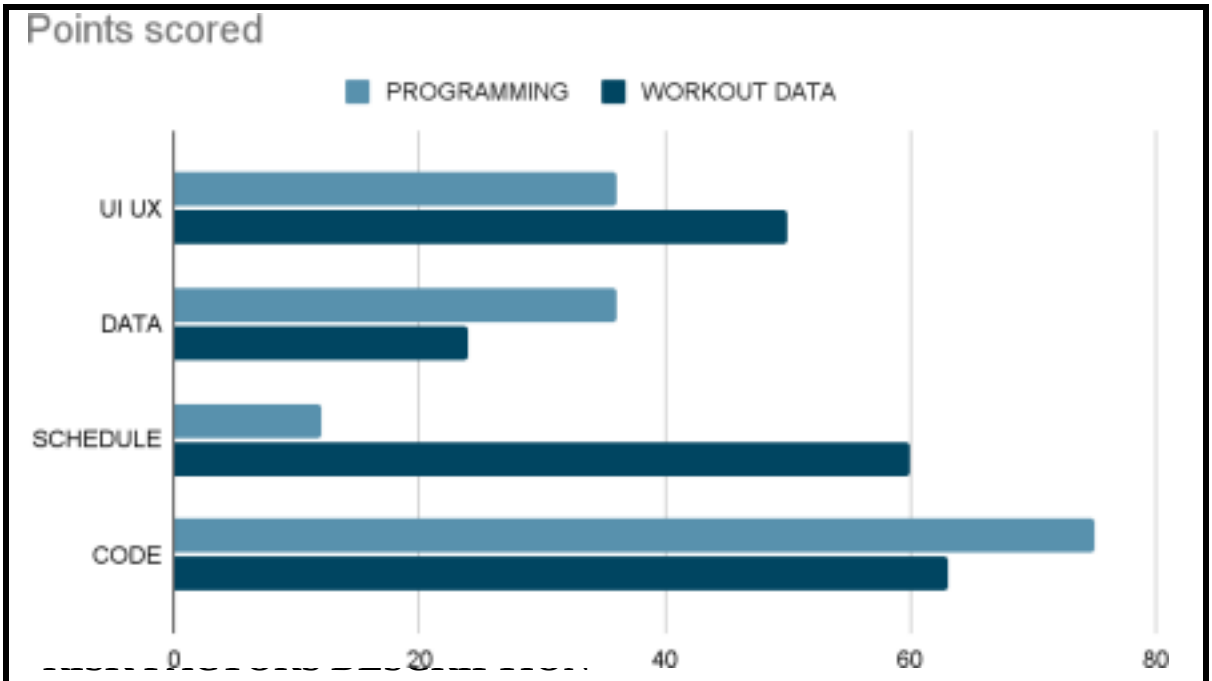
❖ report of the project

❖ releasing the project

❖ proper updates

TIMELINE CHART

FIG2:chart



OCCURENCE OF BUG while programming involves the occurrence of bugs is highly unavoidable since the app involves in constant changing
SCHEDULE PROBLEM the application automatically creates a proper schedule according to the users given data but if the data is not sufficient or the schedule is not suitable for the user it might fail the applications purpose
DATA STORAGE in order to create these perfect schedules it must hold large number of workout data there is risk of data collapse might occur

RISK IDENTIFICATION TABLE

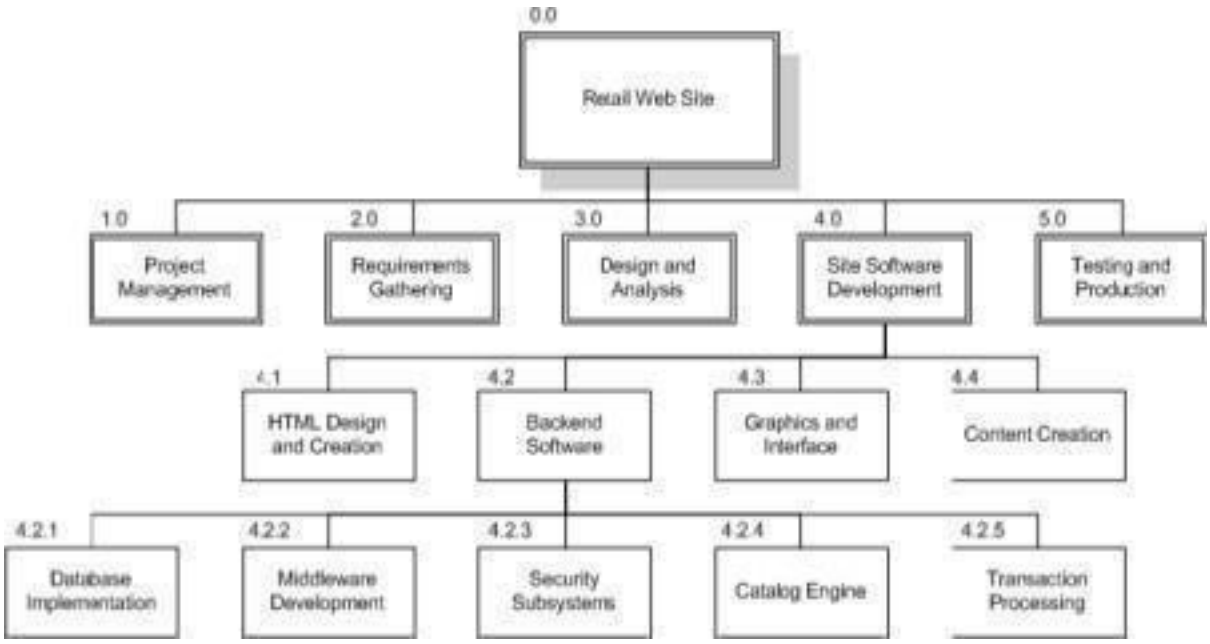


FIG3:site framework

- 0.0 Retail Web Site
- 1.0 Project Management
- 2.0 Requirements Gathering
- 3.0 Analysis & Design
- 4.0 Site Software Development
  - 4.1 HTML Design and Creation
  - 4.2 Backend Software
    - 4.2.1 Database Implementation
    - 4.2.2 Middleware Development
    - 4.2.3 Security Subsystems
    - 4.2.4 Catalog Engine
    - 4.2.5 Transaction Processing
  - 4.3 Graphics and Interface
  - 4.4 Content Creation
- 5.0 Testing and Production

RISK ANALYSIS – SWOT & RMMM

FIG4: pros and cons

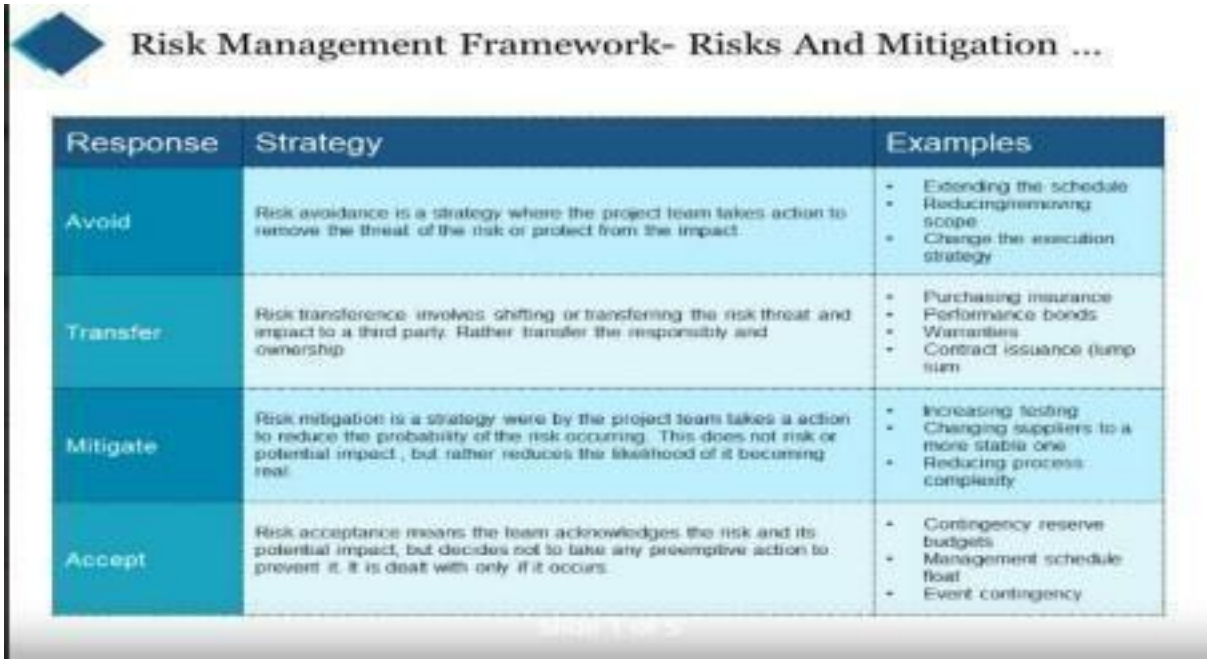
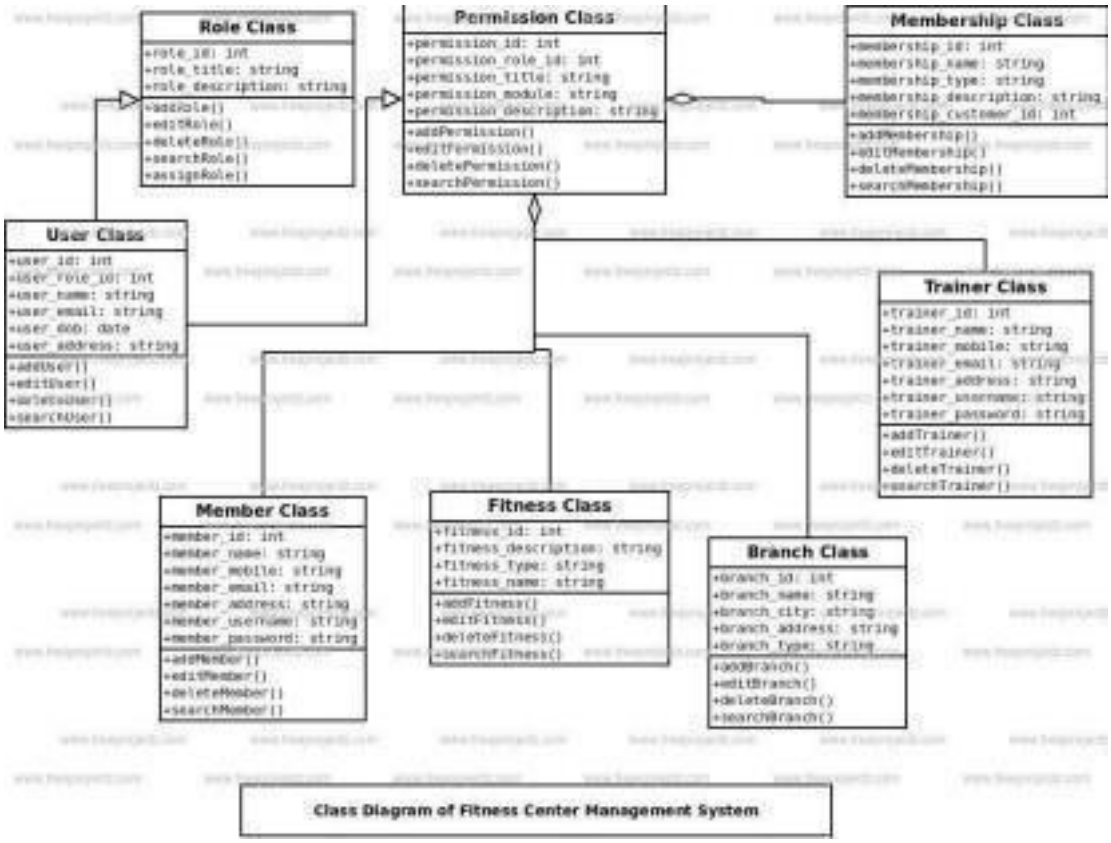


FIG5:risk factors

FIG6: CLASS DIAGRAM



***/ ER Diagram, Notation and Example What is ER Diagram?**

- ER Diagram stands for Entity Relationship Diagram, also known as ERD is a diagram that displays the relationship of entity sets stored in a database. In other words, ER diagrams help to explain the logical structure of databases. ER diagrams are created based on three basic concepts: entities, attributes and relationships.
- ER Diagrams contain different symbols that use rectangles to represent entities, ovals to define attributes and diamond shapes to represent relationships.
- 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. The purpose of ER Diagram is to represent the entity framework infrastructure.

What is ER Model?

- ER Model stands for Entity Relationship Model is a high-level conceptual data model diagram. ER model helps to systematically analyze data requirements to produce a well-designed database.
- ER Model represents real-world entities and the relationships between them. Creating an ER Model in DBMS is considered as a best practice before implementing your database.
- 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.

Why use ER Diagrams?

Here, are prime reasons for using the ER Diagram

- Helps you to define terms related to entity relationship modeling
- Provide a preview of how all your tables should connect, what fields are going to be on each table

- Helps to describe entities, attributes, relationships
- ER diagrams are translatable into relational tables which allows you to build databases quickly
- ER diagrams can be used by database designers as a blueprint for implementing data in specific software applications
- The database designer gains a better understanding of the information to be contained in the database with the help of ERP diagram
- ERD Diagram allows you to communicate with the logical structure of the database to users

Components of the ER Diagram

This model is based on three basic concepts: Entities, Attributes, Relationships

ER Diagram – Notations

- Rectangles represent entity sets.
- Diamonds represent relationship sets.
- Lines link attributes to entity sets and entity sets to relationship sets.
- Ellipses represent attributes
- Double ellipses represent multivalued attributes.
- Dashed ellipses denote derived attributes.
- Underline indicates primary key attributes

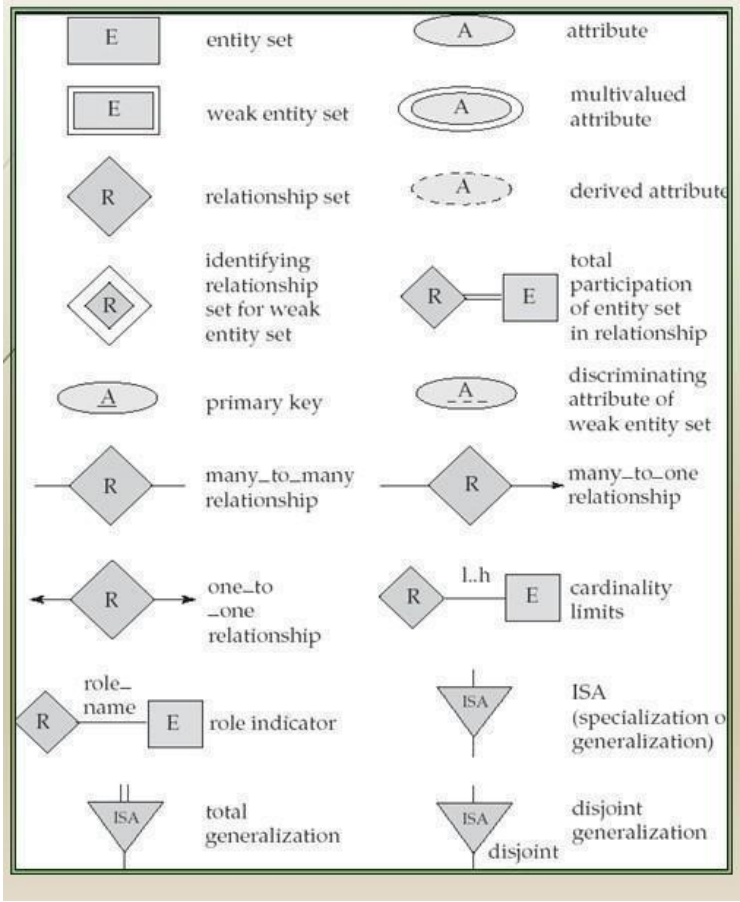
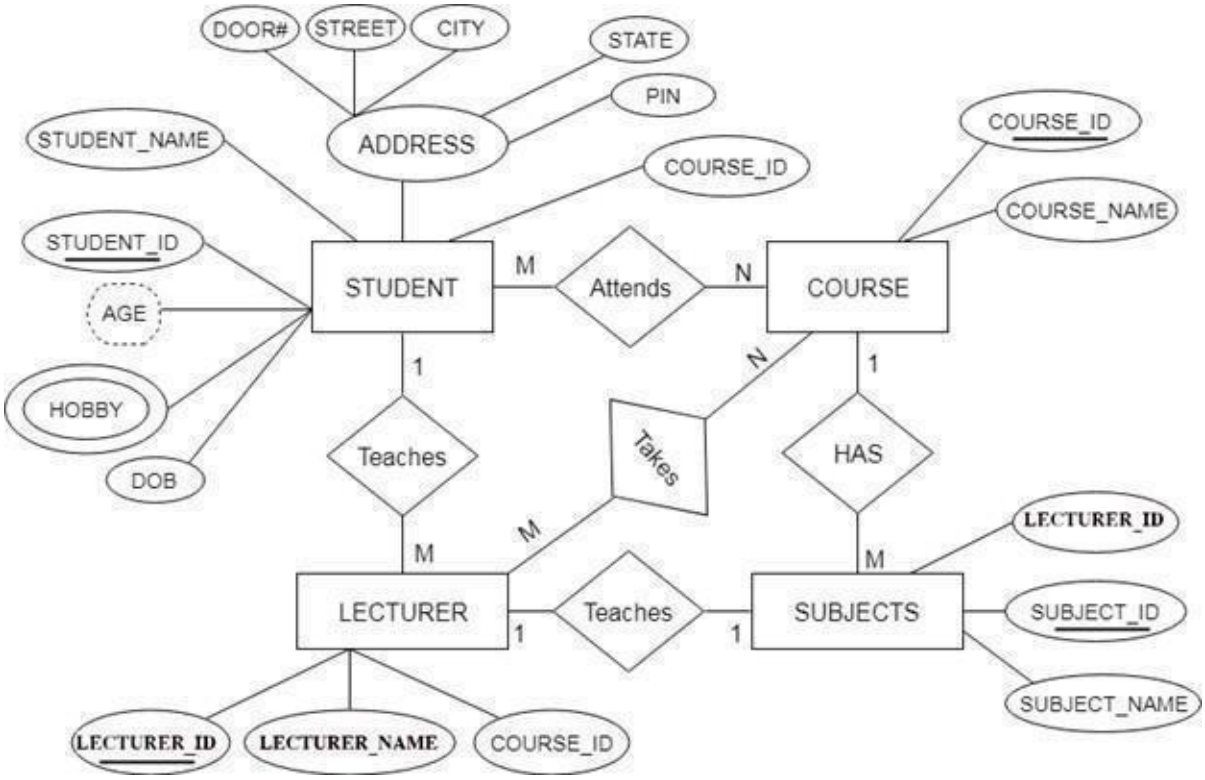


FIG7:ER DIAGRAM

ER Diagram of University Database

FIG8:DATABASE



ADDITIONAL NOTES

- A database can be modeled as a collection of entities, relationship among entities.
- An entity is an object that exists and is distinguishable from other objects. Example: specific person, company, event, plant
- Entities have attributes.

Example: people have names and addresses

- An entity set is a set of entities of the same type that share the same properties. Example: set of all persons, companies, trees, holidays

- Express the number of entities to which another entity can be associated via a relationship set.
- Most useful in describing binary relationship sets.
- We express cardinality constraints by drawing either a directed line (->), signifying "one," or an undirected line (—), signifying "many," between the relationship set and the entity set.

- An entity is represented by a set of attributes, that is descriptive properties possessed by all members of an entity set.

Example: customer = (customer-id, customer-name, customer-street,
customer-city) loan = (loan-number, amount)

- Domain – the set of permitted values for each attribute
- Attribute types:

1. Simple and composite attributes.
2. Single-valued and multi-valued attributes

E.g. multivalued attribute: phone-numbers

3. Derived attributes-Can be computed from other attributes

E.g. age, given date of birth

Cardinality

- For a binary relationship set the mapping cardinality must be one of the following types:

1. One to one

A customer is associated with at most one loan via the relationship borrower. A loan is associated with at most one customer via borrower

2. One to many

A loan is associated with at most one customer via borrower, a customer is associated with several (including 0) loans via borrower

3. Many to one

A loan is associated with several (including 0) customers via borrower, a customer is associated with at most one loan via borrower

4. Many to many

A loan is associated with several (including 0) customers via borrower, a customer is associated with several loans (including 0) via borrower

Weak Entity Set

- An entity set that does not have a primary key is referred to as a weak entity set and represented by double outlined box in E-R diagram.

Example : Consider the entity set payment which got three attributes : payment_number, payment_date and payment_amount. Payment numbers are sequential starting from 1 generally separately for each loan. Although each payment entity is distinct, payments for different loans may share the same payment number. Thus this entity set does not have a primary key.

Discriminator

- The discriminator (or partial key) of a weak entity set is the set of attributes that distinguishes among all the entities of a weak entity set

Example: discriminator of weak entity set payment is the attribute payment_number since for each loan a payment number uniquely identifies one single payment for that loan.

Specialization-Generalization-ISA

- E-R model provides means of representing these distinctive entity groupings
- Process of designating subgroupings within an entity set is called specialization depicted by triangle component labelled ISA ("is a")
- Bottom up design process in which multiple entity sets are synthesized into higher level entity set - Generalization
- ISA relationship may also be referred to as superclass-subclass relationship
- Higher and lower level entity sets are designated by the terms superclass and subclass.
- Specialization and generalization are simple inversions of each other; they are represented in an E-R diagram in the same way.

Total & Partial Participation

- Total participation (indicated by double line): every entity in the entity set participates in at least one relationship in the relationship set

E.g. participation of loan in borrower is total, every loan must have a customer associated to it via borrower

- Partial participation: some entities may not participate in any relationship in the relationship set

Example: participation of customer in borrower is partial

Cardinality limits

- Cardinality limits can also express participation constraints
- Minimum and maximum cardinality is expressed as l..h where l is the minimum and h is the maximum cardinality
- Minimum value of 1 indicates total participation of entity set in relationship set
- Maximum value of 1 indicates entity participates in atmost one relationship set.
- Maximum value of * indicates no limit

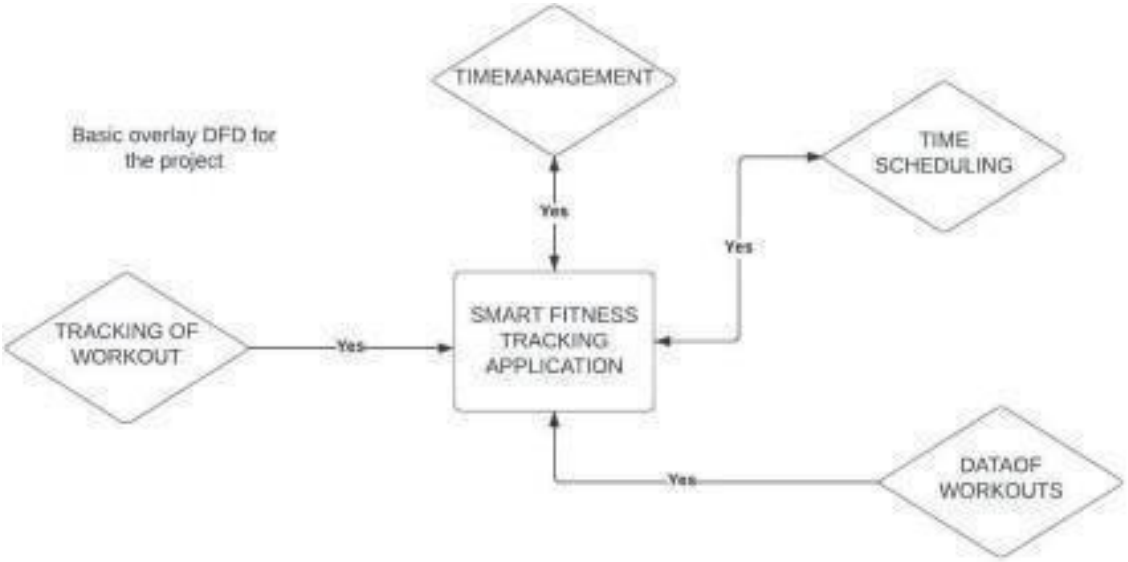
Role indicator

- Entity sets of a relationship need not be distinct
- The labels “manager” and “worker” are called roles; they specify how employee entities interact via the works-for relationship set.
- Roles are indicated in E-R diagrams by labeling the lines that connect diamonds to rectangles.
- Role labels are optional, and are used to clarify semantics of the relationship

Disjoint Generalization

- Disjointness constraint requires that an entity belong to more than one lower level entity set. Example: account entity can satisfy only one condition for account_type attribute ; entity can either be savings or chequing account but not both.ff Signature with date

FIG9:OVERLAY



Data Flow Diagram

The DFD takes an input-process-output view of a system. That is, data objects flow into the software, are transformed by processing elements, and resultant data objects flow out of the software. Data objects are represented by labeled arrows, and transformations are represented by circles (also called bubbles). The DFD is presented in a hierarchical fashion. That is, the first data flow model (sometimes called a level 0 DFD or context diagram) represents the system as a whole. Subsequent data flow diagrams refine the context diagram, providing increasing detail with each subsequent level.

The data flow diagram enables you to develop models of the information domain and functional domain. As the DFD is refined into greater levels of detail, you perform an implicit functional decomposition of the system. At the same time, the DFD refinement results in a corresponding refinement of data as it moves through the processes that embody the application.

A few simple guidelines can aid immeasurably during the derivation of a data flow diagram:

(1) Level 0 data flow diagram should depict the software/system as a single bubble;

(2) Primary input and output should be carefully noted;

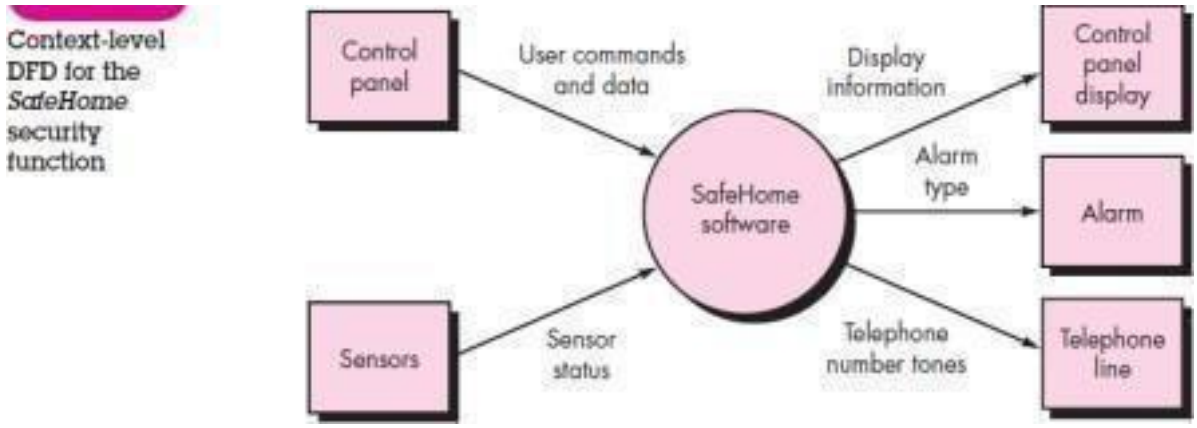
(3) Refinement should begin by isolating candidate processes, data objects, and data stores to be represented at the next level;

(4) All arrows and bubbles should be labeled with meaningful names;

(5) Information flow continuity must be maintained from level to level and (6) One bubble at a time should be refined. There is a natural tendency to overcomplicate the data flow diagram. This occurs when you attempt to show too much detail too early or represent procedural aspects of the software in lieu of information flow.

***/ For Example**

FIG10:DFD Level



0

DFD Level 1

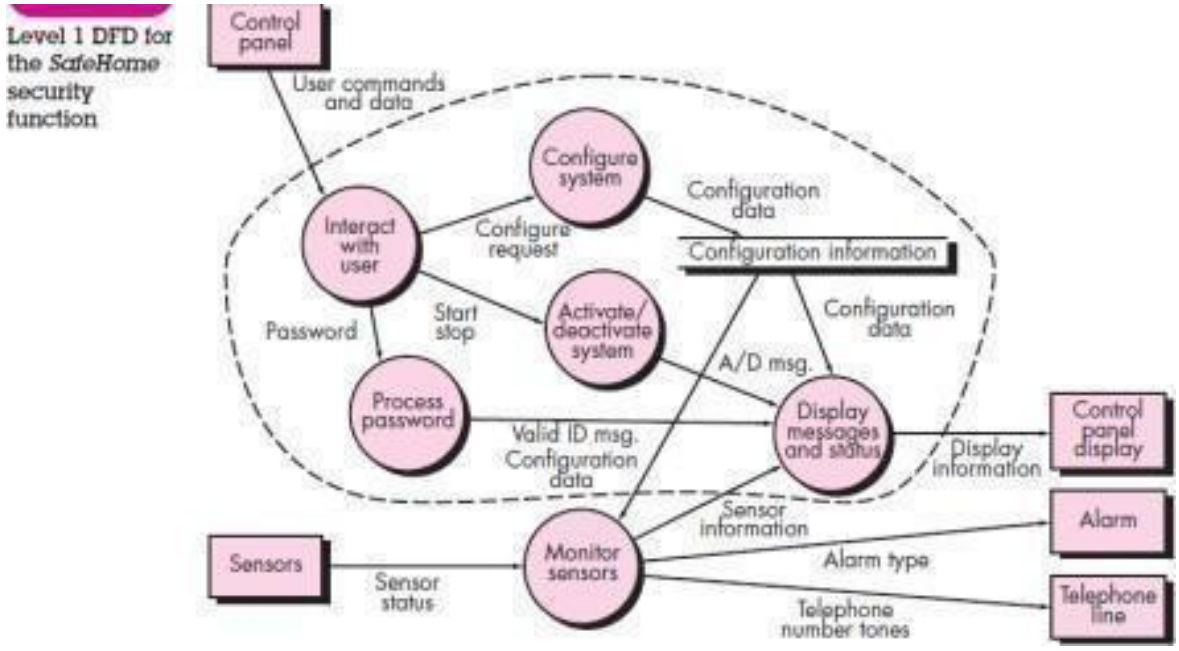
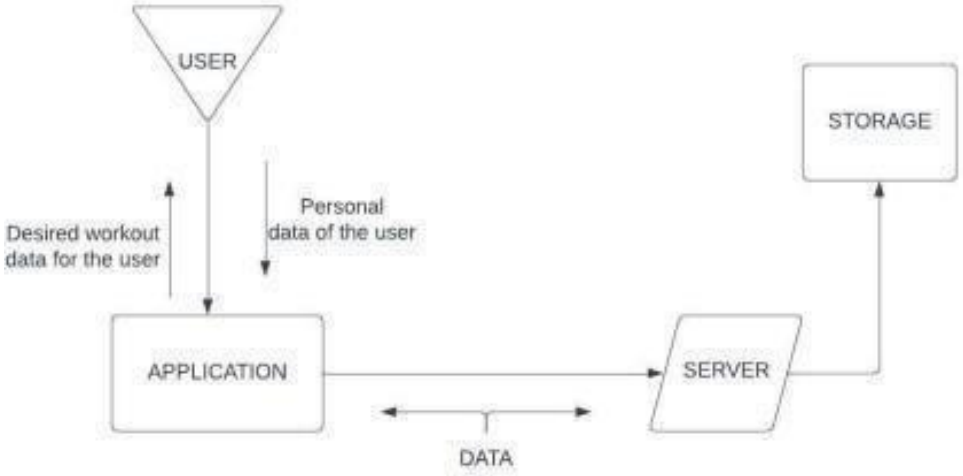
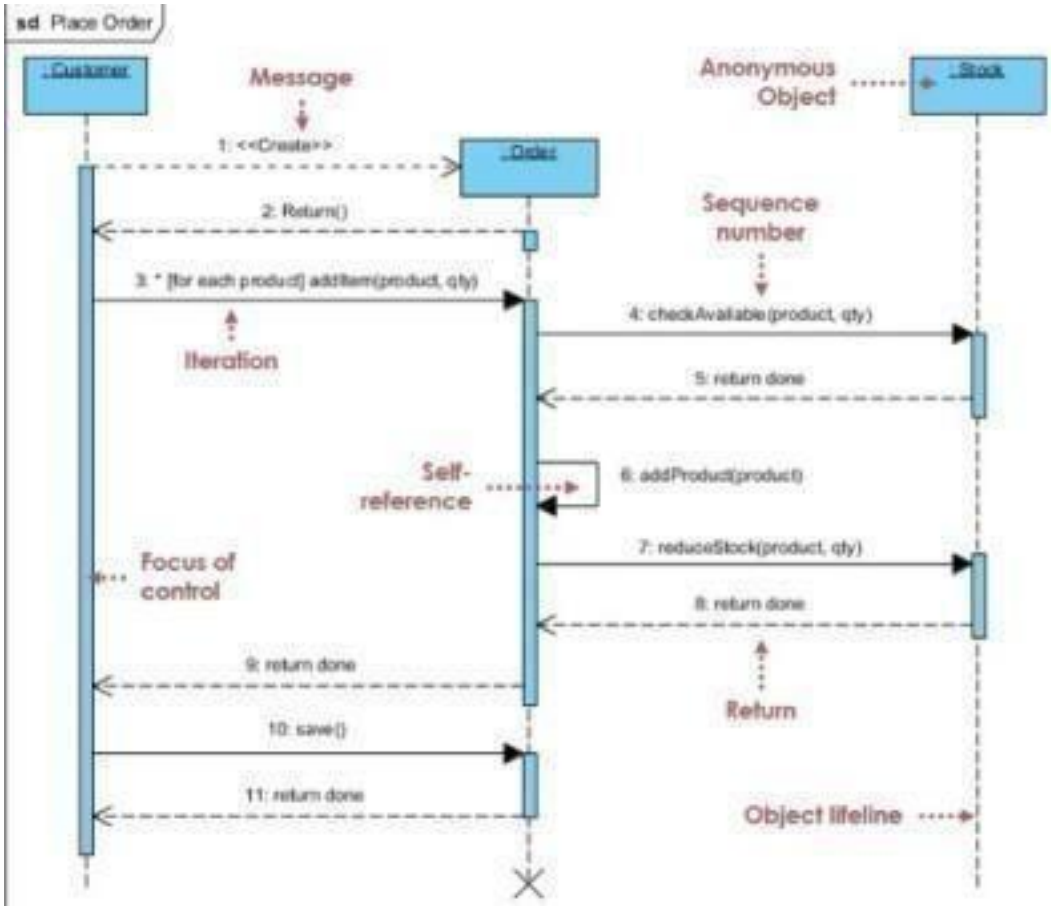


FIG11:<Sequence and Collaboration Diagram>





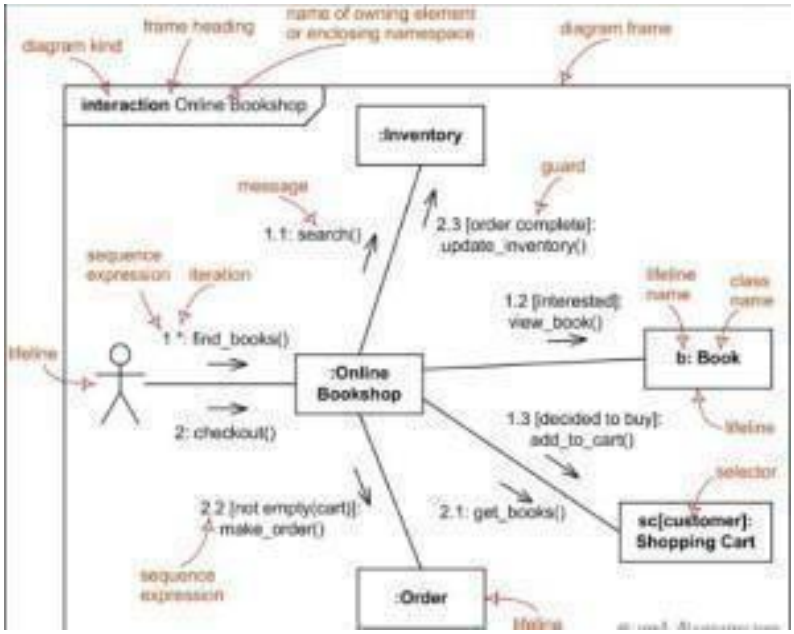


FIG12:Collaboration Diagra

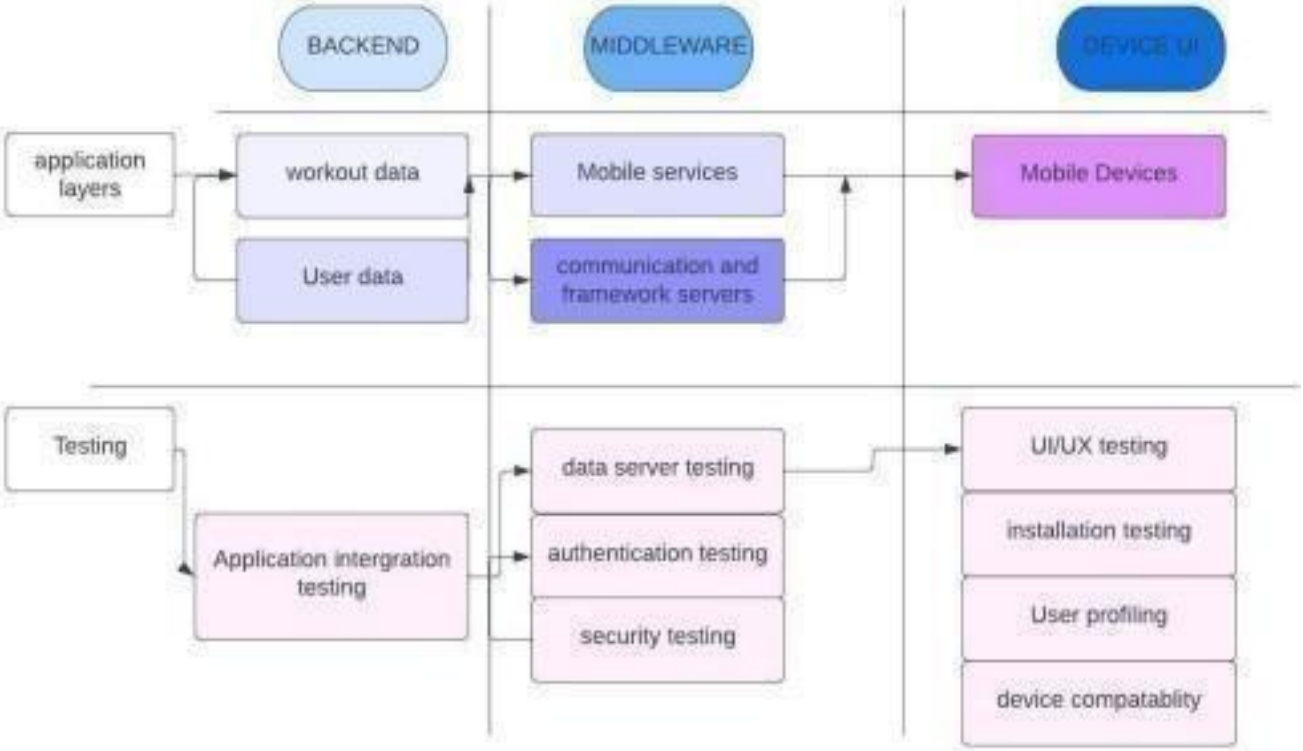


FIG13:FRAMEWORK

Test Plan

Scope of Testing

<<summarize the scope of testing >

Functional: Are all modules covered? Any exception for any modules ? Does automation cover all functional test cases or Regression – Critical Path Test Cases ?

Non-Functional: Are all NFR (Non-Functional Requirements) covered?

Types of Testing, Methodology, Tools

Category	Methodology Tools Required
Functional Requirements	Manual Word Template

TEST CASES: FUNCTIONAL/NON-FUNCTIONAL TEST CASES

Test ID	Test Scenario	Test Case	Execution Steps	Expected Outcome	Actual Outcome	Status	Remarks
1)	Verify User details	Accept Valid details like valid email id and password	Enter the email id on the text box.	User should be able to access all the Gymverse functions	User can access all the Gymverse functions	Pass	Success
2)	Training details	Present the training and diet details to the user for their requirements.	Take all their details. Analyze the diet plan and workout sessions.	User should be able to access all their training details	User gets all their diet and workout plan.	Pass	Success
3)	Test the Java script is properly working.	Check whether the interface is properly working.	Gymverse is working fine or not on every device	User should be able to open the Gymverse in whichever other	User can open the Gymverse in whichever other browsers he/shewa	Running	Running

4)	Users visit the home page	Enter the websit e link tovisit the page	1. Users will type the websi te link. User press enter	Users will go to the root directo ry witha neat user interface	Users go to theroot directory with a neat user interface.	Pass	Success
5)	Test the functi on s of the button s availa ble	Testing whether the buttons are working or not	Users will check by clicki ng onbutt on s anden s ure their worki ng	Users can click various buttons to perform the followin g function	The follow ing functio ns are performe d by theuser after clicking various buttons	Workin g	Working

Introduction:

The GYMVERSE Website is an online platform that provides personalized gym training recommendations based on user goals, fitness levels, and other factors. This report includes functional and non-functional testing to ensure that the website functions as intended and meetsthe user's needs.

Manual Test Case Report:

Category	**Progress Against Plan**	**Status**
Functional Testing	Green	Completed
Non-Functional Testing	Amber	In-Progress

Summary of the current progress:

Functional	**Scenario**	**Test Case Coverage (%)**	**Status**
User Login	• User navigates to the website and clicks on the "Login" button. • User enters a valid username and password. • User clicks on the "Login" button.	20%	Completed
User Registration	1)User navigatesto the website and clicks on the "Register" button. 2)User enters valid registration details. 3)User clicks on the "Register" button.	20%	Completed

Personalized Training Recommendations	• User enters their fitness goals and fitness level information. • User clicks on the "Generate Recommendations" and pays the required package	20%	Completed
Exercise tracking	1)User selects an exercise from the recommended list. 2)User performs the exercise. • User clicks on the "Complete Exercise" button. • User clicks on the "Progress Report" button.	20%	Completed

Non-Functional	Scenario	Test Case Coverage (%)	Status
Website Interface and performance	• Check whether the interface is proper and the user clicks on various links, and buttons. • Check the navigation, load time, speed, and response time of the website.	20%	In-Progress(Need to check the website performance and servers)

Obstaclesto proceeding further on the GYMVERSE website: Lack of accurate exercise recognition and tracking. Limited customization and personalization options for users. Technical challenges in integrating AI algorithms into the website. Data privacy and security concerns in handling user fitness data. User interface and user experience(UI/UX) challenges in making the website user-friendly.

AI technology providers can improve the accuracy of exercise recognition and tracking algorithms through research and development. Gym trainers and fitness experts can collaborate with website developers to provide input on customization and personalization options. Website developers can address technical challenges in integrating AI algorithms, including optimization and testing. Data privacy and security experts can ensure that user fitness data is handled securely and in compliance with relevant regulations. UI/UX designers can work on improving the website's design and user experience to make it more user friendly and intuitive.

DESIGN—

- **Record Daily Food Activity**
- **Invite Friends by sharing location**

Application Uses:

Database : Firebase

Login and Authentication : Firebase, Google, Facebook

Food API : Nutritionix Maps API :

Google Youtube

API : Google

Voice to Text Integration :

Google Step Counter :

In-built Sensor

Animations : WasABeef, AppIntro, ProgressView

Functionalities app consists:

-Navigation Drawer

-Recycler View

-Viewpager

-Toolbars, Menus, Floating Action Buttons

-Co-ordinator Layout

-Fragments and Activities

-User/ saved app state

-Multiple Orientation changes

-Hosting data to cloud (firebase)

-Youtube

-Voice-to-Text

-Map

-GPS

-Real API Data

-Parsing JSON Data

-Async Taks and threads

-Service (Sensors)

-Content provider (Contacts Picker, SMS Manager)

Implementation–

source code for implementation :

```
import sqlite3
from datetime import
datetime from kivy.app
import App
from kivy.uix.boxlayout import
BoxLayout from kivy.uix.button
import Button
from kivy.uix.label import Label
from kivy.uix.screenmanager import ScreenManager, Screen

onnection to the SQLite database conn =
sqlite3.connect('workout_app.db')
c = conn.cursor()

# Create the necessary tables if they don't
exist c.execute('''
  CREATE TABLE IF NOT EXISTS users (
    id INTEGER PRIMARY KEY AUTOINCREMENT,
    username TEXT,
```

```
    password TEXT

c.execute('''
  CREATE TABLE IF NOT EXISTS workouts (
    id INTEGER PRIMARY KEY


class LoginScreen(Screen):
  def__init (self, **kwargs):
    super(LoginScreen, self). init
    (**kwargs)
    layout = BoxLayout(orientation='vertical', padding=10)


    self.username_input =
    TextInput(hint_text='Username')
    layout.add_widget(self.username_input)

    self.password_input =TextInput(hint_text='Password',
    password=True)
    layout.add_widget(self.password_input)

    login_button = Button(text='Login')
    login_button.bind(on_press=self.login)
    layout.add_widget(login_button)

    self.add_widget(layout)

  def login(self, instance):
```

```
        username =
        self.username_input.text
        password =
        self.password_input.text

        # Check if the username and password match a user in the
database c.execute('SELECT id FROM users WHERE username = ?
AND password = ?', (username, password))
        user = c.fetchone()

        if user is not None:
            self.manager.current =
            'menu'
        else:
            print('Invalid username or password')

class MenuScreen(Screen):
    def__init (self, **kwargs):
```

```
        super(MenuScreen, self). init (**kwargs)
        layout = BoxLayout(orientation='vertical', padding=10)

        start_workout_button = Button(text='Start Workout')
        start_workout_button.bind(on_press=self.start_workout)
        layout.add_widget(start_workout_button)

        view_progress_button = Button(text='View Progress')
        view_progress_button.bind(on_press=self.view_progress)
        layout.add_widget(view_progress_button)

        logout_button = Button(text='Logout')
        logout_button.bind(on_press=self.logout)
        layout.add_widget(logout_button)

        self.add_widget(layout)

    def start_workout(self, instance):
        self.manager.current ='workout'

    def view_progress(self, instance):
        self.manager.current = 'progress'

    def logout(self, instance):
        self.manager.current =
        'login'

class WorkoutScreen(Screen): def
    init (self, **kwargs):
        super(WorkoutScreen, self). init (**kwargs) layout
        = BoxLayout(orientation='vertical', padding=10)
        self.timer_label = Label(text='00:00')
        layout.add_widget(self.timer_label)
```

```
start_button = Button(text='Start')
start_button.bind(on_press=self.start_timer)
layout.add_widget(start_button)

stop_button = Button(text='Stop')
stop_button.bind(on_press=self.stop_timer)
layout.add_widget(stop_button)

back_button = Button(text='Back')
back_button.bind(on_press=self.go_back)
layout.add_widget(back_button)

self.add_widget(layout)
```

Conclusion:

In conclusion, the smart fitness tracking app addresses the limitations of existing solutions and presents a comprehensive and user-centric approach to health monitoring. By leveraging mobile devices and wearables, providing personalized dashboards, AI-generated recommendations, and social engagement features, the app

empowers individuals to track their progress, stay motivated, and achieve their fitness goals. Furthermore, the app prioritizes privacy and data security, ensuring that users can monitor their well-being with confidence. With its transformative potential, the smart fitness tracking app is poised to revolutionize the way people approach and improve their overall health and well-being in the modern era.

REFERENCES:

fitness-app · GitHub Topics · GitHub AppSheet tutorial: no-code fitness app | Google Cloud Blog,
How to Build a Fitness Tracker App in 2023 (topflightapps.com)
Fitness Tracker App Development - An Introductory Guide (fatbit.com)