

**Mini Project report on**

**Dynamic Todo List**

**by**

**DEVAL NAYAK [2020510048]**

**Under the guidance of  
Internal Supervisor**

**[ Prof. Sakina Salmani ]**



**Department of Master of Computer Applications**

**Sardar Patel Institute of Technology**

**Autonomous Institute Affiliated to**

**Mumbai University 2020-21**

# **CERTIFICATE OF APPROVAL**

**This is to certify that the following students**

**DEVAL NAYAK [2020510048]**

**Have satisfactorily carried out work on the project entitled**

**“ FINPROJ”**

**Towards the fulfillment of mini project during the year 2020-21.**

---

**Project Guide**

**(Prof. Sakina Salmani)**

**PROJECT APPROVAL CERTIFICATE**

**This is to certify that the following students**

**DEVAL NAYAK [2020510048]**

**Have successfully completed the Project report on**

**“ FINPROJ”**

**, which is found to  
be satisfactory and is approved**

**At**

**SARDAR PATEL INSTITUTE OF TECHNOLOGY,  
ANDHERI (W), MUMBAI.**

**INTERNAL EXAMINER**

**EXTERNAL EXAMINER**

**Head of Department Principal (Dr. Pooja Raundale)**

**(Dr. B. N. Chaudhari)**

SerialNo.	Topic	Pageno.
	Abstract .....	i
	Objectives .....	ii
	List of figures .....	iii
	List of tables .....	v
1.	Introduction.....	1
1.1	Problem Definition.....	1
1.2	Objective and Scope... ..	1
1.3	System Requirements .....	2
2.	Tools and Techniques .....	3
2.1	Tool used.....	3
2.2	Technique used .....	3
2.3	Project management plans .....	3
2.3.1	Tasks .....	3
3.	Project Analysis and Design .....	4
3.1	Methodologies Accepted .....	4
3.2	Use case Diagram and Specifications .....	4
3.3	Activity Diagram and Specification.....	7
3.4	Sequence Diagram... ..	11
3.5	Component and Deployment Diagram .....	12
3.6	ER Diagram .....	13
3.7	Gantt Chart.....	14
3.8	PERT Chart.....	15

4. Project Implementation and Testing .....	16
4.1 Snapshot of Application.....	16
4.2 Test Cases and Report.....	32
5. Project Installation .....	34
6. Future Enhancements.....	35
7. Limitations .....	36
8. Conclusion .....	37
9 Bibliography.....	

## **Abstract**

FinProj its main objective is to create a dynamic todo list in order to study the behavior of the user and to make recommendations on that basis. It uses Angular-Storage to store data and Firebase to make authentication. The try was to create an AI based recommendation of tasks in order to bring habitual change in the user.

### **Some Major Incorporations :**

1. Adding tasks.
2. Helping to get more details about the user's interests and routine with the app.
3. Helps in reminding tasks in a timely manner.
4. Helps to maintain the track of behavior user.

### **The major modules are as follows**

#### **1) User**

- 1.1** User has to register before using the app.
- 1.2** User has to give answer of the questions asked.
- 1.3** User can add task with the timings, category and priority of the task.
- 1.4** User can update the existing task's timing, Task's name, task's category and task's priority.

## **Objectives**

1. User has to register before using the app.
2. User can add a task together with its timings.
3. User can set category of the tasks.
4. User can set priority of the tasks.
5. User can update name of the existing task.
6. User can update category of the existing task.
7. User can update priority of the existing task. .
8. User can delete the existing task.

## **List of figures**

### **Problem Definition**

- User needs to be remind to about the task.
- User needs to keep track of the tasks.
- User needs to add task name together with category, priority and timing of the tasks.

### **Objective and Scope**

- Maximum ease of use to the user
- Easy to set remainder
- Easy to track the schedule of the tasks.
- Easy to update the existing tasks.



## System Requirements

### 1.1 Hardware Requirements:

- Mobile devices will be for working with WebApp.

Memory: 2 GB RAM

■ Hard Drive: 4 GB

■ Internet Connection

## Software Requirements

Android with API>26 version

## Tools and Technique

1. Ionic – Angular
2. SQLite Database

# **Project Management Plan**

The major tasks are:

- Planning
  - Questionnaires
  - Requirements Gathering
- Analysis
  - Study of existing project
  - Technology estimations
- Designing
  - UI design
  - Database design

### 3.2 Use case Diagram and Specifications

<b>USECASE_ID</b>	US_1
<b>USECASE_NAME</b>	FinProj
<b>Created By</b>	Deval Nayak
<b>Date of creation</b>	02-Feb-2022
<b>Actors:</b>	Patient
<b>Description of Usecases:</b>	<ul style="list-style-type: none"> <li>• Register in app</li> <li>• Login in app</li> <li>• Answer the questions</li> <li>• Check the areas of interests</li> <li>• Add tasks name</li> <li>• Add task's category</li> <li>• Add task's priority</li> <li>• Add task's timing</li> <li>• Update task's name</li> <li>• Update task's category</li> <li>• Update task's priority</li> <li>• Update task's timing</li> </ul>
<b>Pre-conditions</b>	1. Every user must register in app  2. Tasks must be added to the list before setting remainder.
<b>Post-condition</b>	N.A.

<b>Extends</b>	Login<-extends<-registration  Set schedule<- extends<- task
<b>Includes</b>	tasklist->includes->tasks
<b>Assumptions</b>	

### 3.3. Activity Diagram and Specifications

Step	User	System	Buisne ss Rules
1	Login		Br_1
2		Validat es	Br_2
3	Register		Br_3
4	Adds new tasks		Br_4
5	Add category		Br_5
6	Add priority		Br_6
7	Add timing		Br_7
8	Update Tasks		Br_8
9	Update Category		Br_9
10	Update priority		Br_10

11	Update timing		Br_11
----	---------------	--	-------

**Table 3.2 Activity diagram Business Rules**

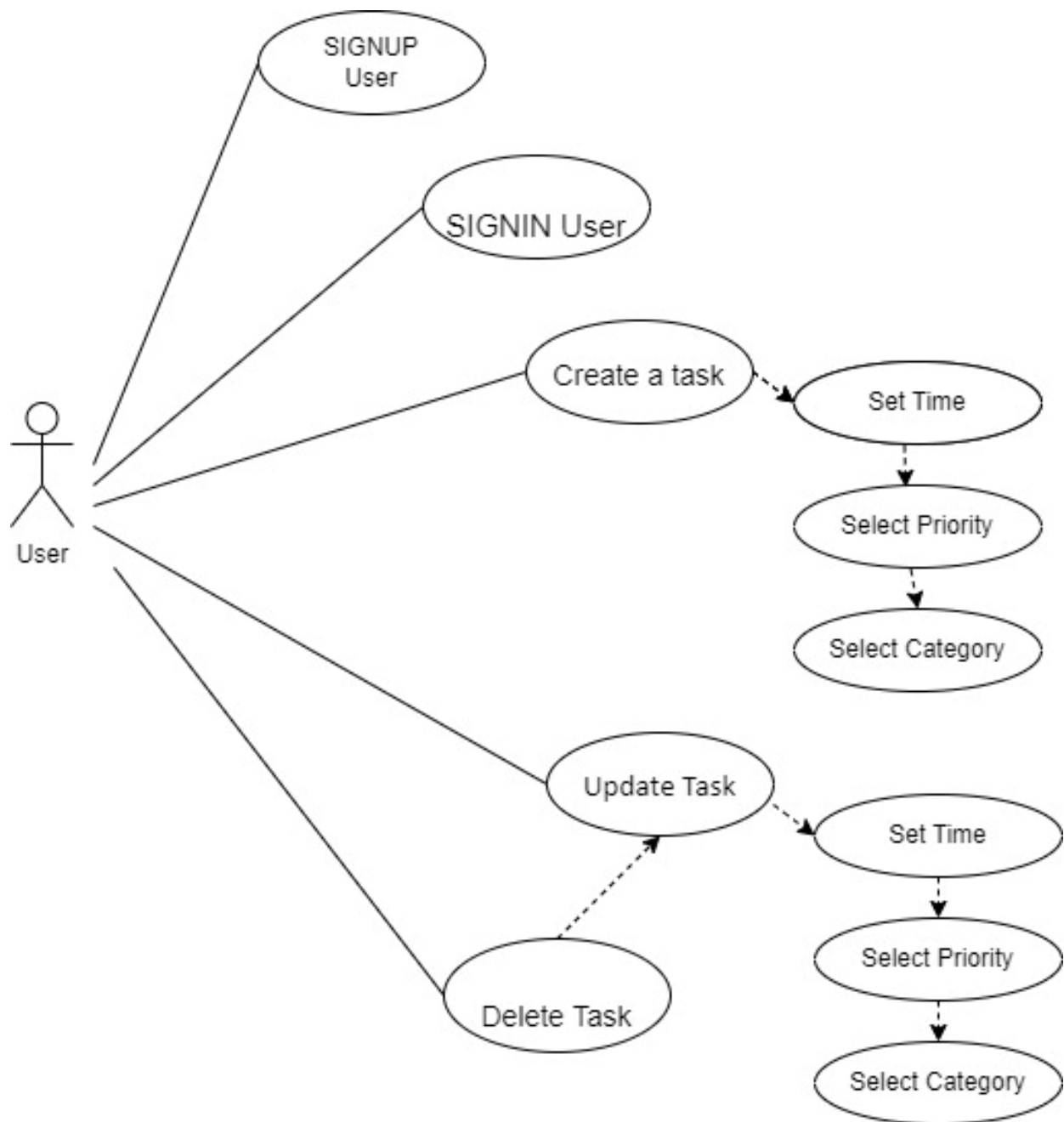
To implement activity diagram specification.

ID	Buisness Rules	Business Rule Description
BR_1	Login	Mandatory
BR_2	Register	Mandatory
BR_3	Add tasks	Optional
BR_4	Add category	Optional
BR_5	Add priority	If BR_4 is complete:  Then set remainder
BR_6	Add timing	If BR_4 is complete:  Then User can see the details of the Task on Page
BR_7	Update Tasks	If BR_3 is done:

**Table 3.3 Detailed Activity Diagram Business Rules.**

### 3.1 Use Case Diagram

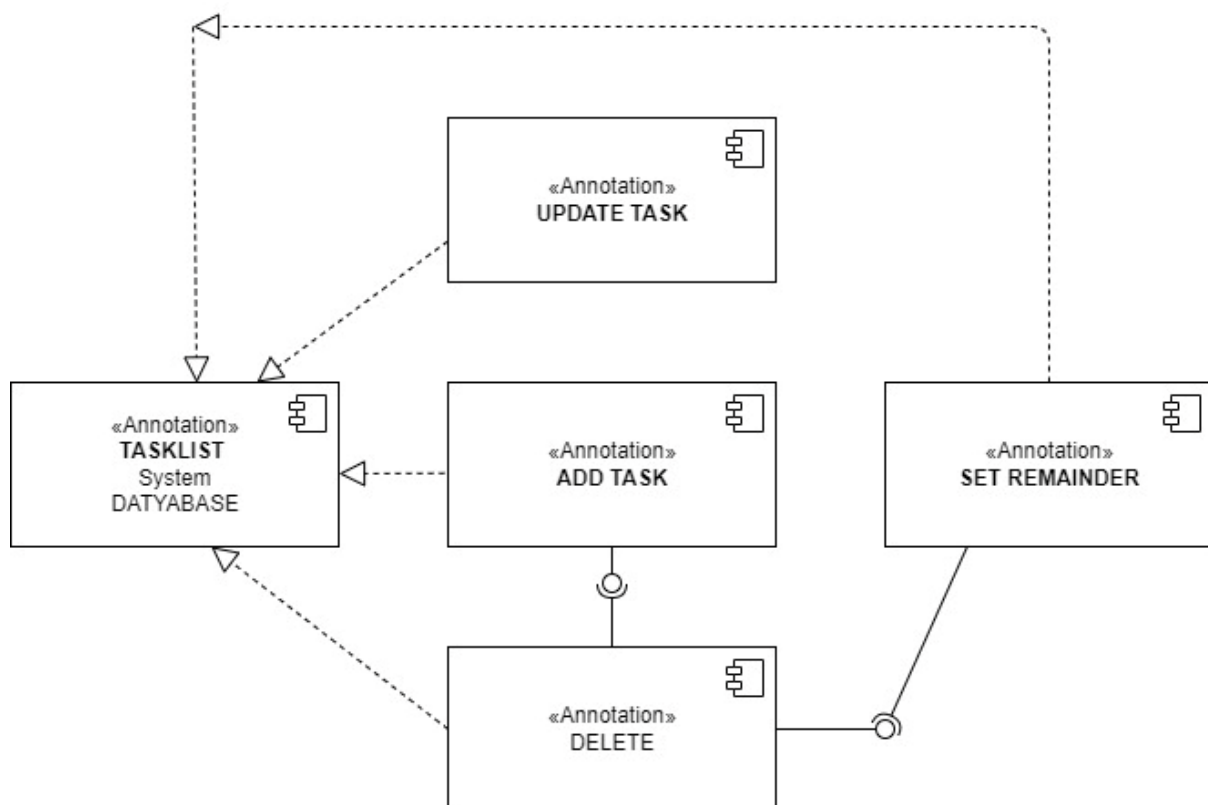
Use case diagrams are a set of use cases, actors and their relationships. They represent the use case view of a system. A use case represents a particular functionality of a system. So a use case diagram is used to describe the relationships among the functionalities and their internal/external controllers. These controllers are known as actors.



## Component Diagram

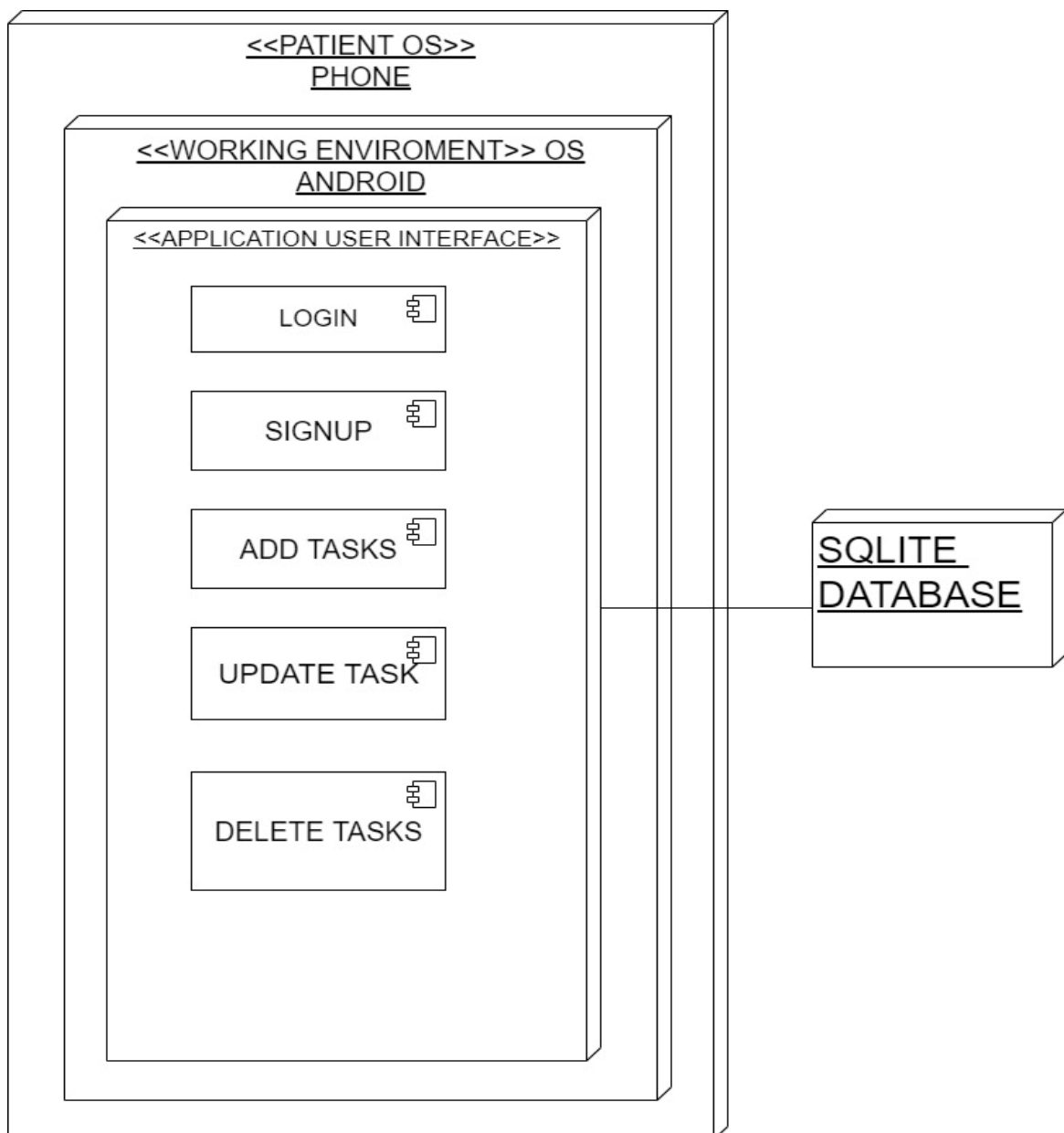
Component diagrams are different in terms of nature and behavior. Component diagrams are used to model physical aspects of a system. Component diagrams are used to visualize the organization and relationships among components in a system. These diagrams are also used to make executable systems. It does not describe the functionality of the system but it describes the components used to make those functionalities. Component diagrams can also be described as a static implementation view of a system. Static implementation represents the organization of the components at a particular moment.

## Component Diagram



## Deployment Diagram

Deployment diagrams are used to visualize the topology of the physical components of a system where the software components are deployed. Deployment diagrams are used for describing the hardware components where software components are deployed. Component diagrams and deployment diagrams are closely related. Component diagrams are used to describe the components and deployment diagrams shows how they are deployed in hardware.

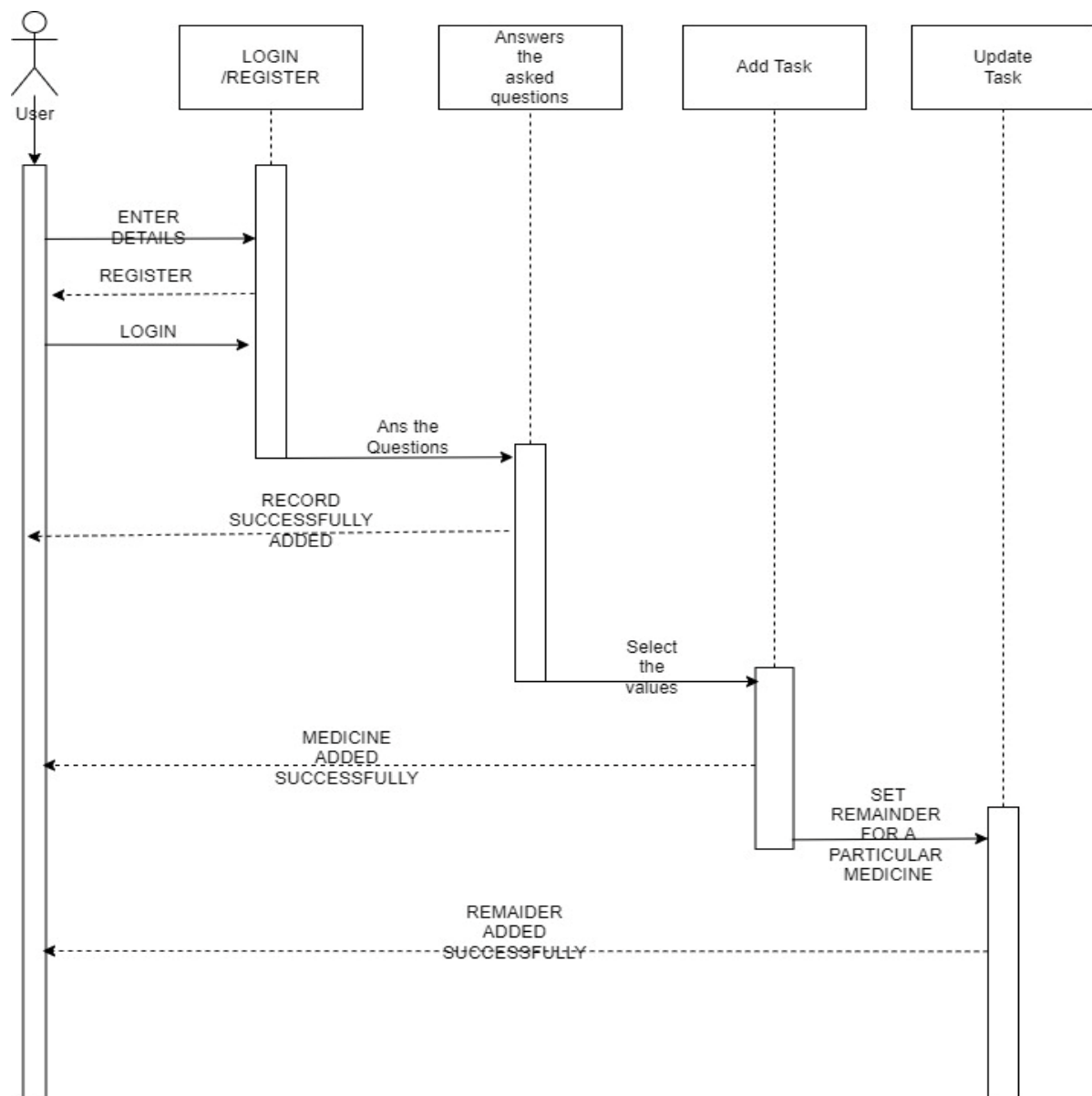


**Deployment Diagram**



## Sequence Diagram

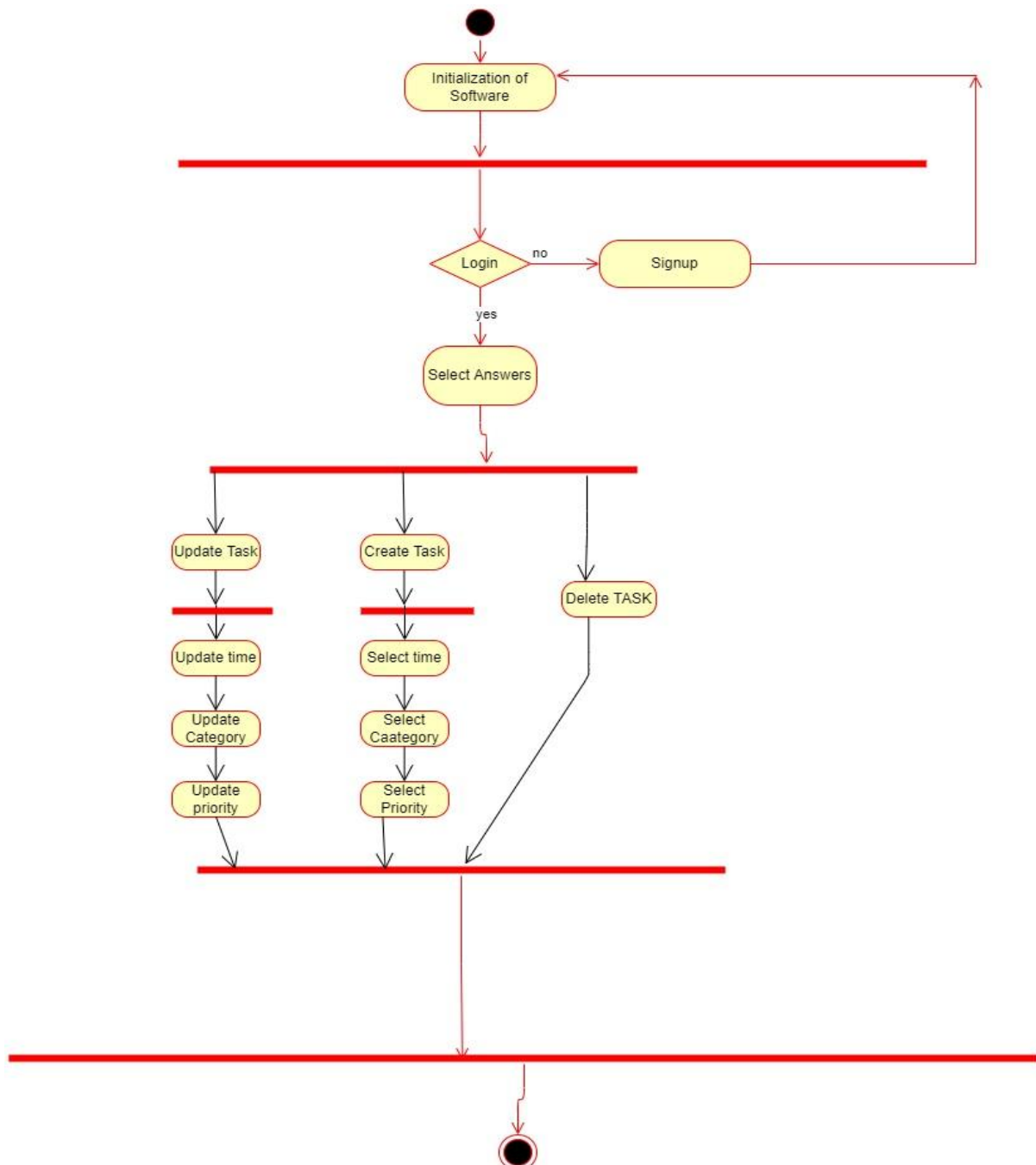
A Sequence diagram is an interaction diagram that shows how objects operate with one another and in what order. It is a construct of a message sequence chart. A sequence diagram shows object interactions arranged in time sequence. A sequence diagram shows, as parallel vertical lines (lifelines), different processes or objects that live simultaneously, and, as horizontal arrows, the messages exchanged between them, in the order in which they occur. This allows the specification of simple runtime scenarios in a graphical manner.



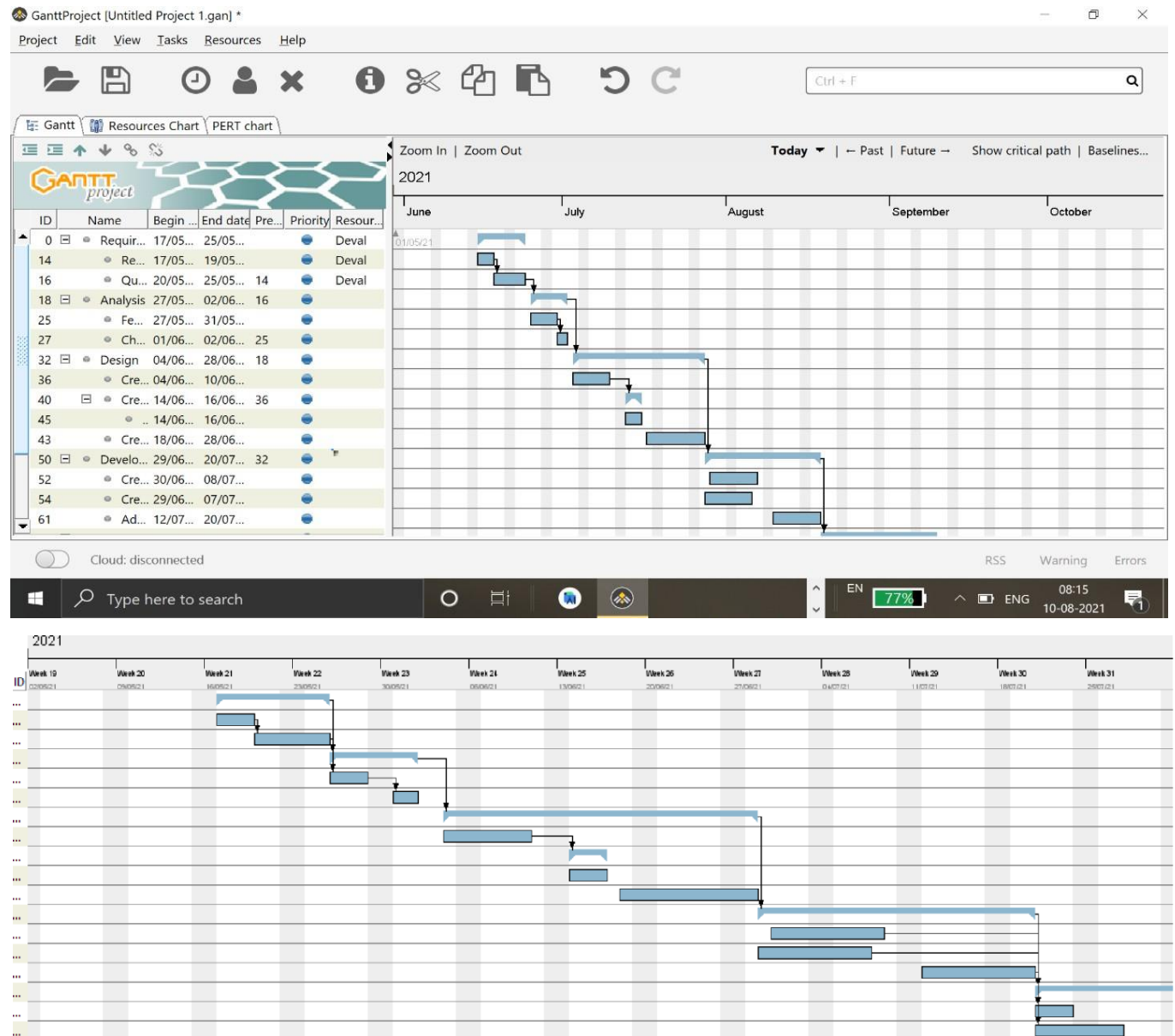
**Sequence Diagram**

### 4.2.3 Activity Diagram

Activity diagram is another important diagram in UML to describe dynamic aspects of the system. Activity is a particular operation of the system. Activity diagrams are not only used for visualizing dynamic nature of a system but they are also used to construct the executable system by using forward and reverse engineering techniques. The only missing thing in activity diagram is the message part. It does not show any message flow from one activity to another. Activity diagram is some time considered as the flow chart. Although the diagrams looks like a flow chart but it is not. It shows different flow like parallel, branched, concurrent and single.



# Gantt Chart



login

Email...

Password...

SIGN IN

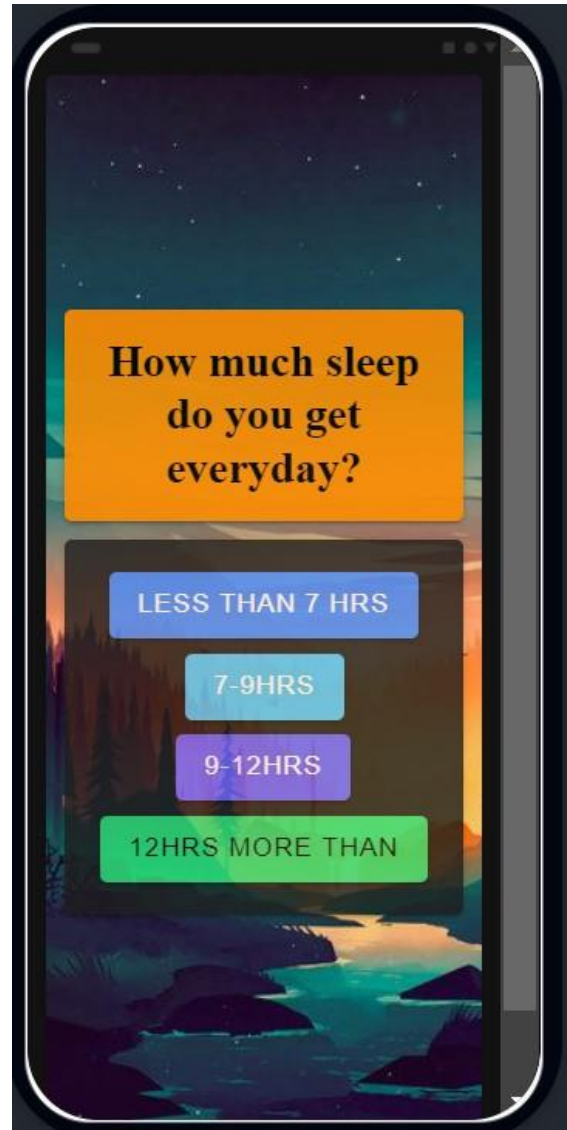
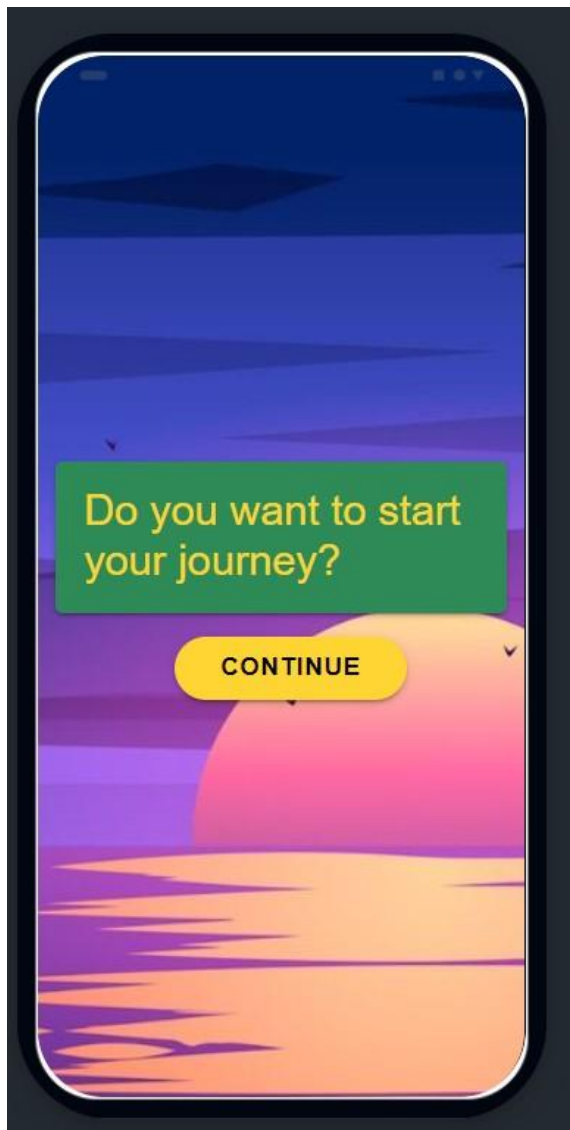
SIGN UP

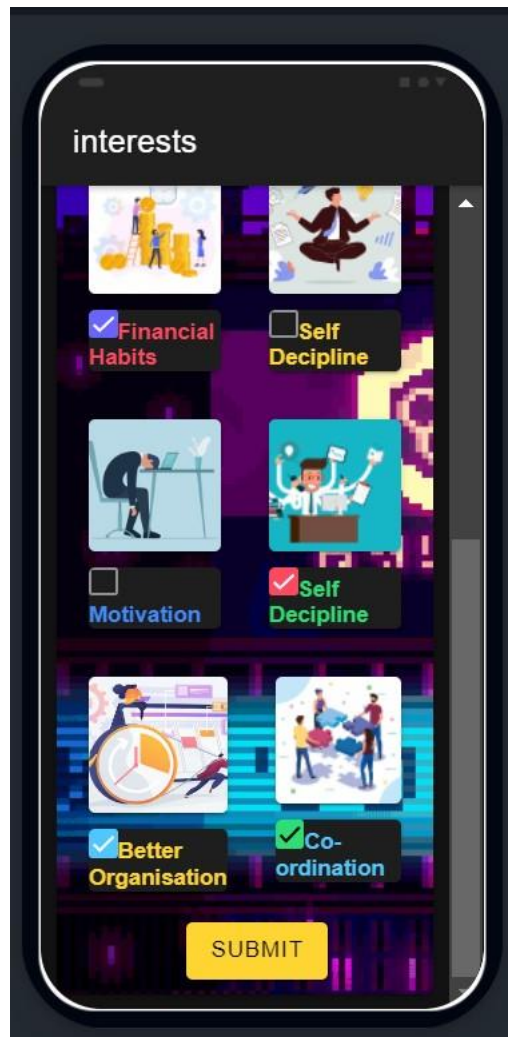
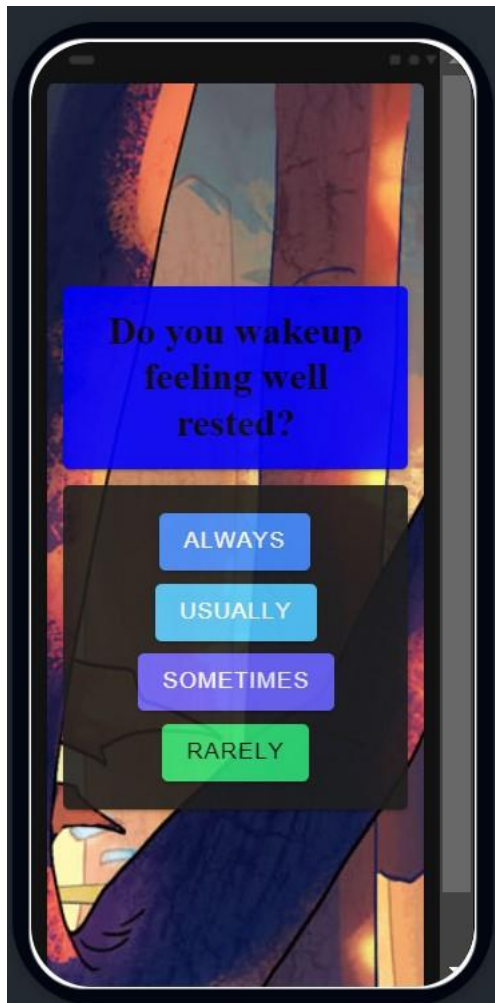
signup

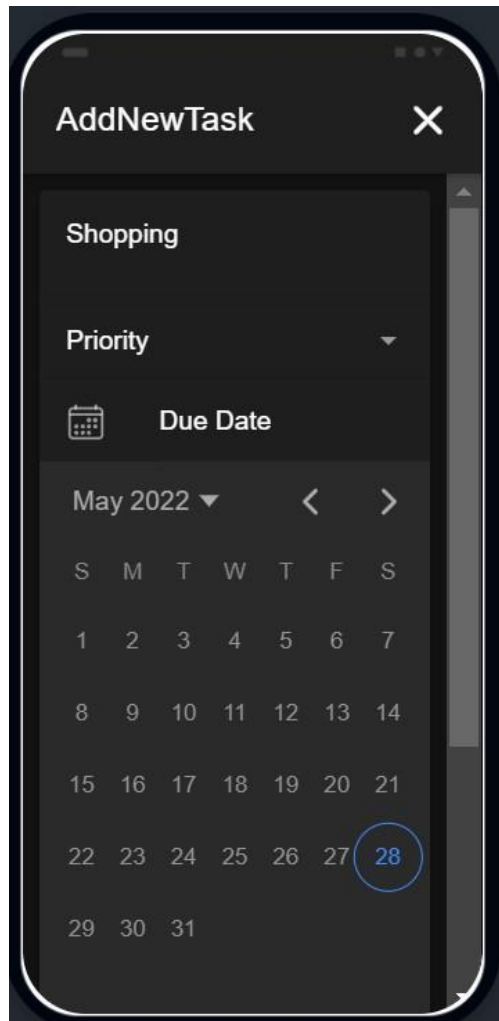
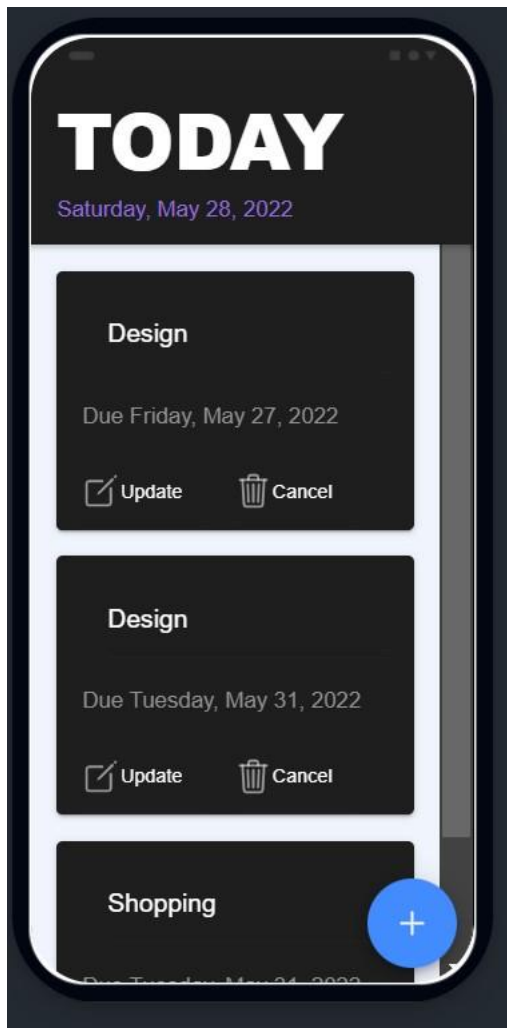
devalnayak39@gmail.com

Password...

SIGN IN








AddNewTask

×

S	M	T	W	T	F	S
1	2	3	4	5	6	7
8	9	10	11	12	13	14
15	16	17	18	19	20	21
22	23	24	25	26	27	28
29	30	31				

Time

8:02 PM



Category

personal

work

personal

home

ADD