

TAP TODO

UCS503 Software Engineering Project Report

Mid-Semester Evaluation

Submitted by:

Harshita Pandey 102003235

Chahat Joneja 102003239

Meharamt Singh 102003241

Anannya Singh 102003253

BE Third Year-COE

Group No: 10

Submitted to:

Ms. Pragya Mishra



**THAPAR INSTITUTE
OF ENGINEERING & TECHNOLOGY
(Deemed to be University)**

Computer Science and Engineering Department TIET, Patiala
September 2022

TABLE OF CONTENTS

S.No.	Assignment	Page No.
1.	Project Selection Phase	4
1.1	i. Software Bid	4
1.2	i. Project Write Up	6
1.3	ii. Feasibility Report	7
1.4	iii. Gantt Chart	10
2.	Analysis Phase	11
2.1	i. Use Cases	11
2.2	ii. Use-Case diagram	12
2.3	iii. Use Case Templates	13
2.4	iv. Swimlane diagrams	22
2.5	v. Data Flow Diagrams –Level 0, Level 1 , Level 2	22
2.6	vi. Software Requirement Specification (SRS) in IEEE Format	24
2.7	vii. User Story Card	31
3.	Design Phase	39
3.1	i. Class Diagram	39
3.2	ii. Sequence Diagram	39
3.3	iii. Collaboration Diagram	40
3.4	iv. Database Design - ER Diagram	40
4.	Implementation	41

UCS503- Software Engineering Lab

4.1	Component Diagram	41
4.2	Deployment Diagram	41
4.3	Screenshot of working project	42
5.	Testing	51
5.1	Test Cases	51

Software Bid

Group : COE10

Dated: 28/8/2022

Team Name: Debuggers

Team ID (will be assigned by Instructor): 2

Please enter the names of your Preferred Team Members.

- You are required to form **a three to four person** teams
- Choose your team members wisely. You will not be allowed to change teams.

Name	Roll Number	Project Experience	Programming Language Used	Signature
Harshita Pandey	102003235	Flight Management System, Fraud Detection	PL/SQL, Flask, Python	
Chahat Joneja	102003239	Airport Management System	PL/SQL	
Meharamt Singh	102003241	Flight Management System, Social Media Data Extraction	PL/SQL, Flask, Python	
Anannya Singh	102003253	Flight Management System, Fraud Detection	PL/SQL, Flask, Python	

Programming Language / Environment Experience

List the languages you are most comfortable developing in, **as a team**, in your order of preference. Many of the projects involve Java or C/C++ programming.

1. C/C++

2. Python
3. Flutter
4. HTML/CSS/JavaScript

Choices of Projects:

Please select **4 projects** your team would like to work on, by order of preference:

[Write at-least one paragraph for each choice (motivation, reason for choice, feasibility analysis, etc.)]

First Choice	Notefy: A todo app focused on modularity, where each module modifies how the app functions.
Second Choice	Blood Bank: Every day 12,000 people in India die due to the sheer lack of donated blood. To overcome this, we think of creating a software to find local blood donors in any geographical area, helping many individuals to get the required blood group within reasonable time.
Third Choice	Food Delivery and Tracker: A software to order food from different restaurants in the nearby location and also a tracker to track your food.
Fourth Choice	Price Compare: A software which helps the users to compare the price of different items from different e-commerce sites.

Planning Phase

Project Write Up

- **Overview**

Todo apps are often the first project one tries to build when they try web or app development, but the end result is not something that seeks to innovate upon or even match popular todo apps. Managing tasks is an intensely personal thing. When it comes to to-do lists, everyone has different criteria.

Therefore we've intended to create a modular app that empowers the end-user to define what a todo planner and todo mean to them instead of following the developer's view of a perfect planner.

- **Objective of Project**

A todo app focused on modularity, where each module modifies how the app functions. Todo is a collection of modules and their contents. Sample modules:

1. Datetime
2. Timespan
3. Alarm
4. Audio
5. Images (including canvas notes)
6. Checklist

- **Global (app modules):**

1. Modify functioning at the application level
2. Priority list
3. Focus Mode
4. Notes hierarchy (tentative)
5. Top-level hierarchy (tentative) - Could be built as a particular case on top of a hierarchy.

- **Need for the Project**

A todo-planner is one of the most essential utilities, but they are generally not as customizable as they should be. The level of customizability this app hopes to achieve is unparalleled among open-source offerings.

- **Scope of the Project**

A cross-platform todo app supporting Linux, Windows, and Android, usable by all.

Feasibility report of TAP TODO

Feasibility Study in Software Engineering is a study to evaluate feasibility of proposed project or system. It is one of the important four stages of the Software Project Management Process. As the name suggests, feasibility study is the feasibility analysis or it is a measure of the software product in terms of how beneficial product development will be for the organisation from a practical point of view. It is carried out based on many purposes to analyse whether a software product will be right in terms of development, implantation, contribution of project to the organisation etc.

- **Technical Feasibility**

Project **TAP TODO** is a complete web-based application. The main technologies and tools associated with it are:

Language: Flutter, Firebase

Tools: Figma

Database: Sqlite3 (for development), PostgreSQL (for production)

- **Economic Feasibility**

The platform will be free to use, to begin with. In the initial stages, the user base will majorly include university and college students. With time, more functionalities will be delivered to push the software towards an

enterprise version. With extended utilisation and greater downloads, the software may be charged in the future.

- **Legal Feasibility**

The software assures that all data inside the system or its part will be protected against malware attacks or unauthorised access. The information inside the database can only be accessed through registered personnel. Each of the technical tools is a freely available open-source program that can be used by any user for their intended purpose but is copy-righted under their respective owners. We have ensured that our project follows all the safety and security protocols and doesn't partake in any copyright infringement.

TAP TODO is licensed under the MIT licence.

Permissions given:

1. Commercial use
2. Modifications
3. Distribution
4. Private use

Limitations:

1. Liability
2. Warranty

- **Social Feasibility**

A System has Social Acceptance if it has favourable reception and is perceived as being an effective system by its users. It can be easy to overlook the users involved in a system which makes the manual job of a person easier. Even if the system is soundproof, users are what make or break the system. Perception is crucial. A TODO application is something that can ease the work for the user to have their list of things present in one place whilst giving them the ability to share them with others as well. Hence the system is well socially feasible.

- **Scheduling Feasibility**

In addition to learning new technologies and stacks, and other overheads and unforeseen challenges, the schedule for the project has been designed keeping college and academic schedules in mind. Despite its strictness, the schedule includes room for overhead expenses, ensuring that the project doesn't drift off track.

Task	Tentative Timeline
Project Initialisation	August, 2022
Documentation and Analysis	September, 2022
UI/UX completion	September, 2022
Prototype completion	October, 2022
Deployment and Testing	November, 2022
Bug Fixing and Performance Testing	November, 2022
Tentative completion	December, 2022

- **Operational Feasibility:**

In Operational Feasibility the degree of providing service to requirements is analysed along with how easy the product will be to operate and maintain after deployment.

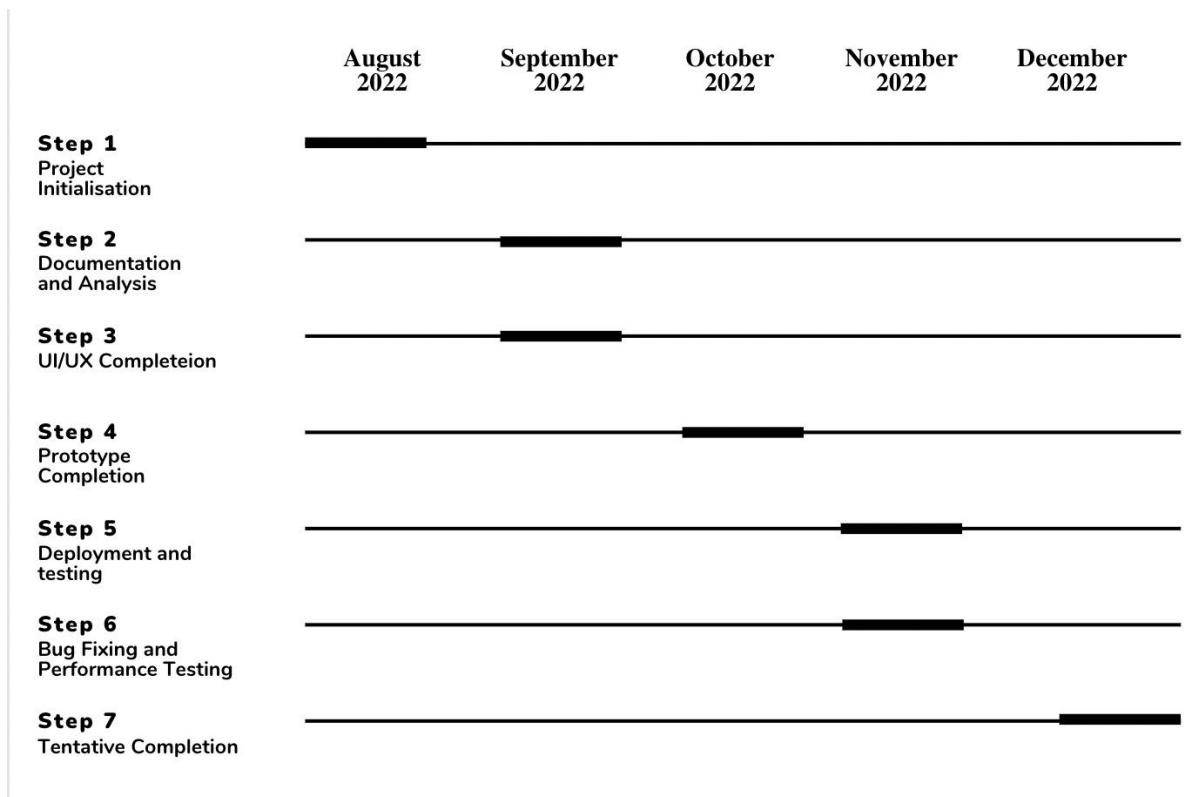
- The software after being deployed can be regularly maintained to fix any existing bugs by employees.
- Effort to maintain the project is negligible as all the work is being done on software and no hardware is used. The requirements of the user will be

satisfied by our project. Our project will be easy to use, and we have also given the pharmacy owners an option to send us feedback which will be considered by our team that will help us to upgrade as per the requirements stated by them.

- **Cultural Feasibility:**

The compatibility of the proposed project with the cultural environment included in the cultural feasibility. Our project is ethical and does not contain anything which has the potential to offend/target any caste, community, or creed. This project was developed and designed by a multicultural team of various religious faiths and beliefs.

- **Gantt Chart**

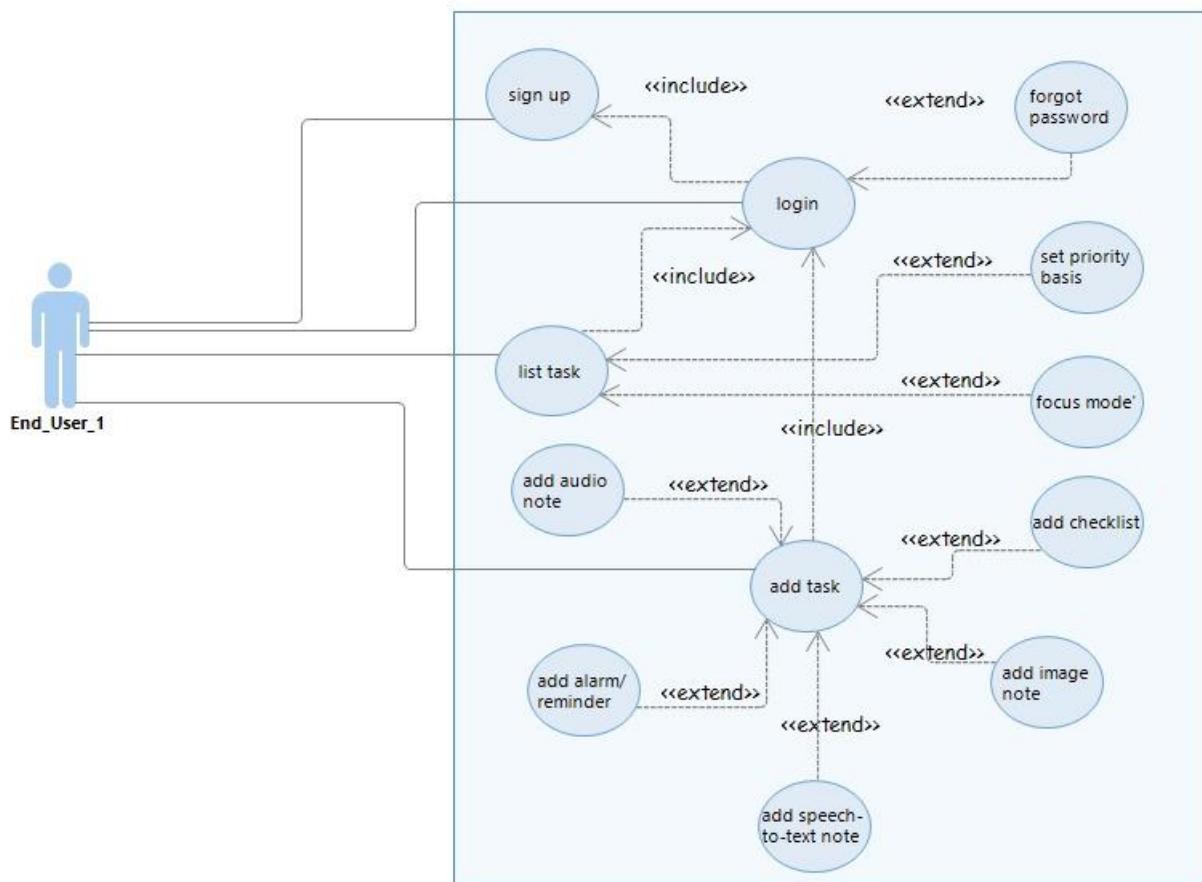


Analysis Phase

Use Cases

Use Case	Description
Login	To authenticate the user and give user access to his/her account.
Signup	To register the user and give user access to his/her account for TAP Todo.
List Task	To list tasks and their time limits, priority order.
Add Task	To add tasks and their time limits, priority order.

● Use Case Diagram



● Use Case Templates

Use Case ID	UC-01
Use Case	Login
Use Case Purpose	To authenticate the user and give user access to his/her account.
Use case Description	The goal of this use case is to authenticate the registered user for the app and retrieve his details (all the listed tasks, scheduled tasks, etc.) and give him/her access to the several features of the app
Assumptions	The user must already be registered/ signed up to the TAP Todo application and must know basic English to understand the visuals.
Variations	The user can login through the website or the phone application as per his/her convenience The user might also use 'keep me logged in' In this case, the user need not have to login again and again
Trigger	The user needs to enter the username or email id and password in the respective fields using keyboard/keypad and then need to click/tap on the Login button.
Primary Actors	The primary actor is the account holder or the end user himself
Secondary Actors	None

Pre-Conditions	The user must already be registered or signed up to TAP Todo with a valid email address or Google Account
Normal Scenario	<ol style="list-style-type: none"> 1. Open the TAP Todo App or Website Login Page. 2. Input the login email id and password in respective fields. 3. Click/Tap on Login button.
Extension points	<p>Forgot Password</p> <p>Forgot Password allows you to change your password using a valid user email id and some security checks.</p> <p>OAuth</p> <p>OAuth using google allows you to directly login using your Google Account connected to the TAP Todo app.</p>
Alternate Scenario	NA
Post Conditions	<p>Success end condition</p> <p>The user will be redirected to the LIST TASK screen. Failure end condition:</p> <p>In case user is unable to login:</p> <ul style="list-style-type: none"> ● check if the emailID used is valid or not, if not, then the user will be asked to re-enter the email ● If emailID is valid but not registered, the user will be redirected to the register/signup screen. ● If the email is registered, the user will be asked to re-enter the password and also shown a forgot password option which can be used in case the user forgot the password.
Special Requirements	NA

UCS503- Software Engineering Lab

Use Case ID	UC-02
Use Case	Signup
Use Case Purpose	To register the user and give user access to his/her account for TAP Todo.
Use case Description	The goal of this use case is to register the user for the app and give him/her access to the several features of the app
Assumptions	The user must have a valid email id OR an existing google account
Variations	The user can register through the website or the phone application as per his/her convenience
Trigger	The user needs to enter the username or email id and password, re-enter the password in the respective fields using keyboard/keypad and then need to click/tap on the Register button.
Primary Actors	The primary actor is the account holder or the end user himself
Secondary Actors	None
Pre-Conditions	The user must have a valid email id.
Normal Scenario	1. Open the TAP Todo App or Website Signup/Register Page.

	<p>2. Input the login email id and password, re-enter password in the respective fields.</p> <p>3. Click/Tap on the register button.</p>
Extension points	<p>OAuth</p> <p>Just click on sign up using google/others n select mail id to use oAuth registration.</p>
Alternate Scenario	NA
Post Conditions	<p><u>Success end condition:</u></p> <p>If the entered email id is valid, a verification email will be sent to the user at the specified email address</p> <p>After email verification the user will be confirmed as REGISTERED and redirected to the LIST TASK screen.</p> <p><u>Failure End Condition:</u></p> <p>In case user is not able to register:</p> <ul style="list-style-type: none"> ● check if the emailID used is valid or not, if not, then the user will be asked to re-enter the emailId . ● check if the password used is strong or not, if not, then the user will be asked to re-enter the password
Special Requirements	NA

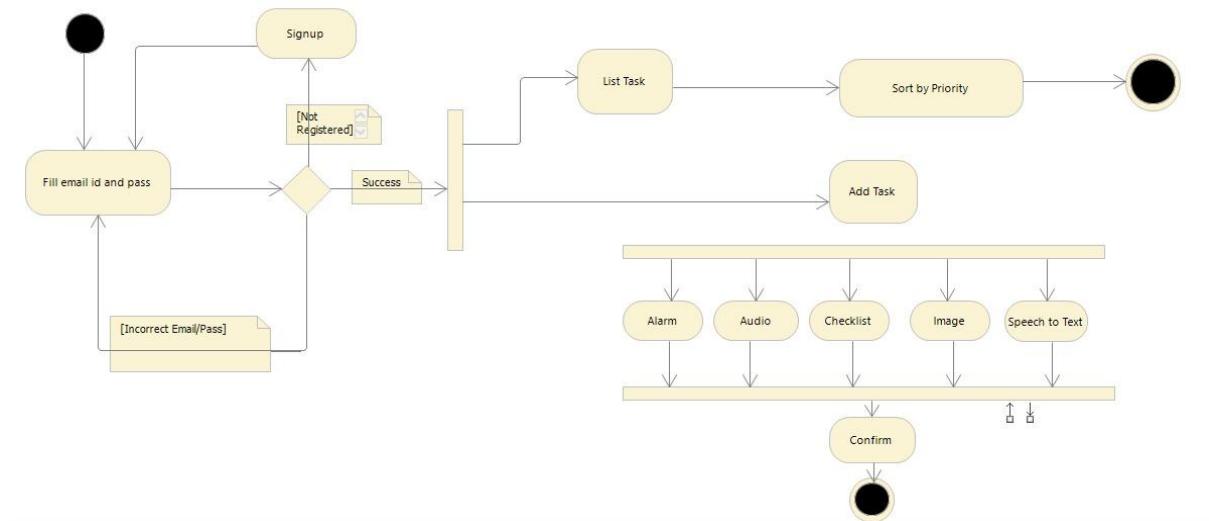
Use Case ID	UC-03
Use Case	List Task
Use Case Purpose	To list tasks and their time limits, priority order
Use case Description	The goal of this use case is to list tasks along with time limits the user had entered previously.
Assumptions	The user must know how to operate the device and applications to go to the desired feature
Variations	The user can be logged in to his/her account in application or website
Trigger	NA
Primary Actors	The primary actor is the account holder or the end user himself
Secondary Actors	None
Pre-Conditions	The user must have been registered and logged in to the application interface
Normal Scenario	1. Open the TAP Todo App And Login.(if not already logged in)

	2. After successful login it should display all the tasks.
Extension points	<p>Time Left</p> <p>In case the user has set a task with a time limit remaining time will be shown to him.</p> <p>Focus Mode</p> <p>In case the user has focus mode enabled he will be able to see tasks in priority order- One task at a time.</p> <p>Priority Task</p> <p>In case the user has Priority tasks, he will be able to see tasks in a priority order specified by him/her.</p>
Alternate Scenario	NA
Post Conditions	<p>Success end condition</p> <p>All tasks will be shown on the screen (according to their priority (if enable)) with time left to complete them (as per the set time limits)</p> <p>Failure end condition:</p> <p>In case user is not able to see all tasks:</p> <ul style="list-style-type: none"> ● The user may not have entered any tasks ● Authentication Failure.
Special Requirements	NA

Use Case ID	UC-04
Use Case	Add Task
Use Case Purpose	To add tasks and their time limits, priority order.
Use case Description	The goal of this use case is to add tasks to the user.
Assumptions	The user must know how to operate the device and applications to go to the desired feature
Variations	The user can be logged in to his/her account in application or website
Trigger	NA
Primary Actors	The primary actor is the account holder or the end user himself
Secondary Actors	None
Pre-Conditions	The user must have been registered and logged in to the application interface
Normal Scenario	<ol style="list-style-type: none"> 1. Open the TAP Todo App and login (if not already logged in) 2. After successful login it should display all the tasks. 3. Click/Tap on the Add Task Button 4. The User Shall Be Shown Various Methods To Add Task

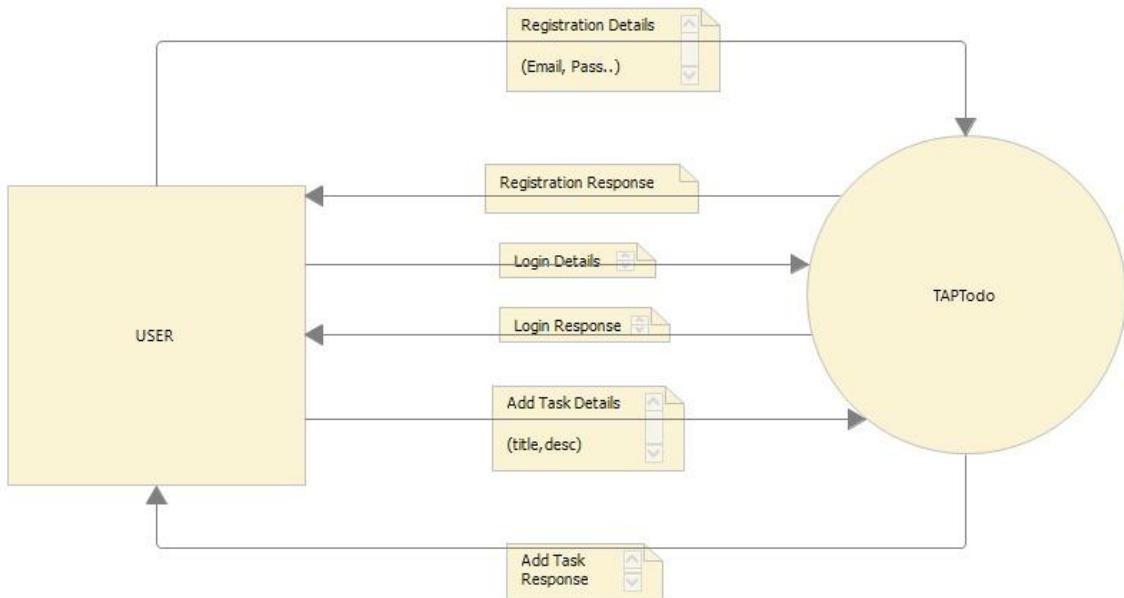
Extension points	<p>Check List</p> <ul style="list-style-type: none"> ● Users can add sub tasks (lower priority) using this checklist feature. <p>Audio Note</p> <ul style="list-style-type: none"> ● Users can add their audio in a task. <p>Image Note</p> <ul style="list-style-type: none"> ● Users can add custom images in a task. <p>Speech to Text</p> <ul style="list-style-type: none"> ● Users can use speech-to-text features to add tasks. <p>Priority</p> <ul style="list-style-type: none"> ● Users can use a number to add priority order for different tasks. <p>Alarm / Reminder</p> <ul style="list-style-type: none"> ● Users can enable alarm / reminder for a specific task. Users can even enter a time limit for ending of a task.
Alternate Scenario	NA
Post Conditions	<p>Success end condition</p> <ul style="list-style-type: none"> ● All tasks will be shown on the screen (according to their priority (if enable)) with time left to complete them (as per the set time limits) on successful login ● On Clicking/Tapping on ADD TASK, the user redirected to the ADD TASK window/screen with the various methods to add a task (as shown in the extension points) ● After the details of task entered (desired input given), click on DONE , success message will be popped “Task Successfully Added” <p>Failure end condition:</p> <p>In case user is not able to add a task:</p> <ul style="list-style-type: none"> ● Pop an error message saying “Task couldn’t be added”.
Special Requirements	NA

Swimlane Diagram

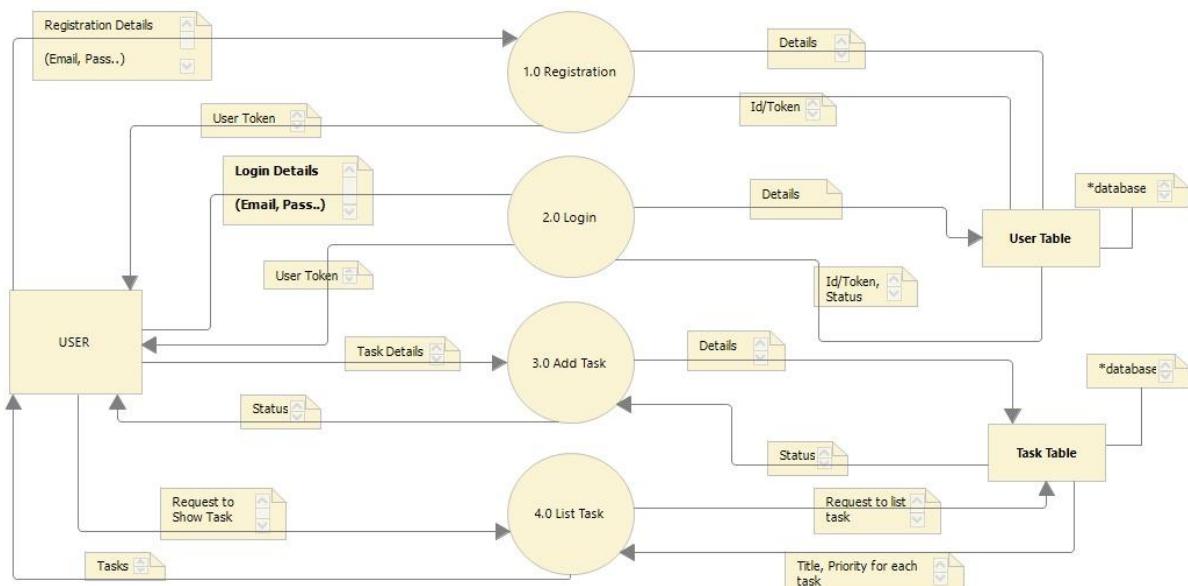


Data Flow Diagrams

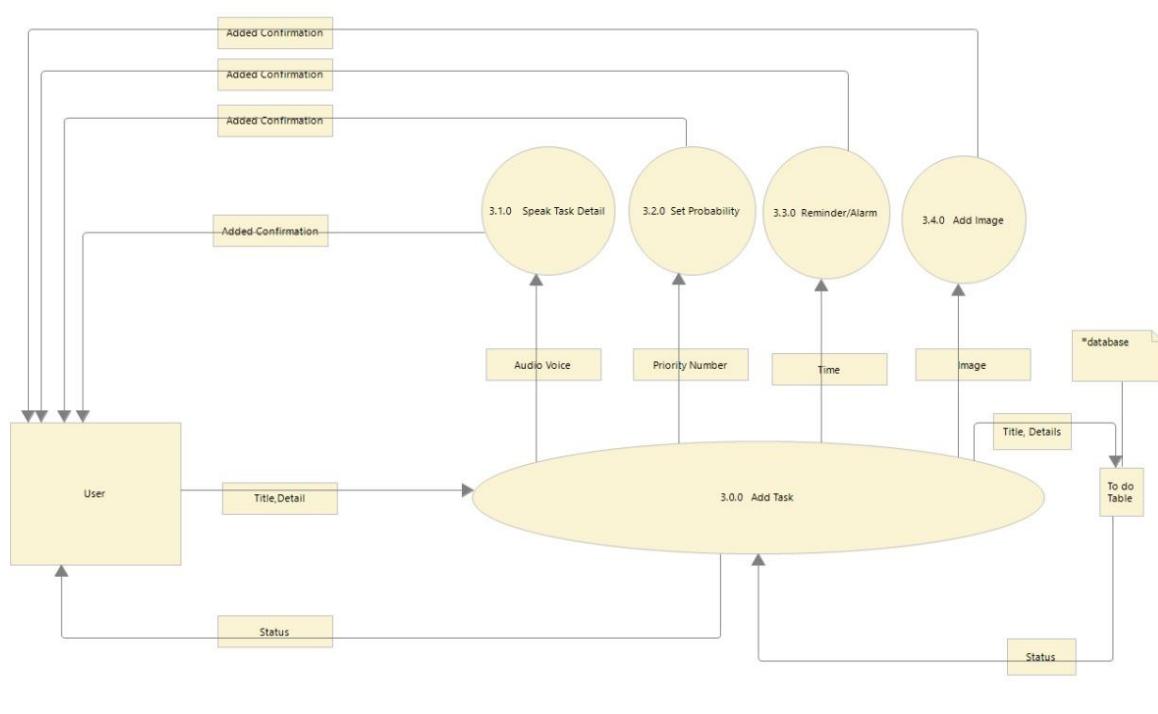
DFD LEVEL 0



DFD LEVEL 1



DFD LEVEL 2



Software Requirements Specification

Introduction

- Purpose

The purpose of this application is to create a modular app which empowers the end user to define what a todo planner and todos mean to them, instead of following the developer's view of a perfect planner.

- Intended Audience and Reading Suggestions

This document is intended for, such as developers who follow the guidelines according to use case diagrams and develop the modules. Project managers use this SRS for checking project milestones and deadlines. Users use SRS to verify the functional requirements and approve the project according to mentioned criteria. Testers use SRS for testing purposes such as alpha testing, integration testing, etc. The project is also used by company employees. So these are the audience for the SRS and in this document product scope, purpose, project

description and all the references mentioned which is useful for all the audience. Sequence of reading the document beginning with purpose and product scope then references from where we take some knowledge then overall description of project then interface requirements then system features and non- functional requirements.

- Project Scope

A todo-planner is one of the most essential utilities but they are generally not as customisable as they should be. The level of customizability this app hopes to achieve unparalleled among open source offerings.

- References

Some of the design Guidelines <https://material.io/design/guidelines-overview>

- Overall Description
- Product Perspective

A todo-planner is one of the most essential utilities but they are generally not as customisable as they should be. The level of customizability this app hopes to achieve unparalleled among open source offerings.

We plan to build a solution that combines all the features of a daily planner to an annual planner making your life simpler.

- Product Features

A todo app focused on modularity, where each module modifies how the app functions. A todo is a collection of modules and their contents.

Sample modules:

1. Date-time – Must be able to add Deadline date and time
2. Timespan – instead of date/time user can add timer / time limit

3. Alarm – on completion of deadline alarm/reminder will be given to user
4. Audio – instead of only text users can add audio, images or checklist in the todo description
5. Images (including canvas notes)
6. Checklist
7. Global (app modules): Modify functioning at the application level
8. Priority list – todos sorted on the basis of priority listing
9. Focus Mode – one single task at a time according to priority
10. Notes hierarchy (tentative)
11. Top level hierarchy (tentative) - Could be built as a special case on top of a hierarchy

c. User Classes and Characteristics

Focusing on security and Privacy, in our system we have only one user class which is:User

Users will be able to login and logout, add tasks, sync tasks they created from their account.

d. Operating Environment

Frontend–Flutter/dart

Backend–Firebase

Users can use it as a mobile app or desktop app.

e. User Documentation

This software provides security. The login form prevents the system from being misused by unauthorised users. Only an authorised operator will be granted rights to modify as per requirements. The user should know the programming language very well that is used to develop software.

f. Assumptions and Dependencies

We assume the user has an android or iOS phone or a desktop/laptop to install the app.

3. System Features

a. Login

a.1 Description and Priority

The login form is used by all the users. This module has the highest priority when compared to all the other modules. This model allows the user to enter his email id and password in order to make use of the software.

a.2 Stimulus/Response Sequences

This screen provides the text boxes for the user to enter credentials or connect with their google account

a.3 Functional Requirements

On valid input of email and password, password will be compared to a hashed(encrypted)password in the online db and a user token will be saved on the client side.

REQ1: User must be registered.

Table 1: Functional Requirements for User Login Screen

Purpose	This screen provides the interface for the user to enter credentials or connect with their google account.
Inputs	Email, password or login with google
Processing	On valid input of email and password, password will be compared to a hashed (encrypted) password in the online db and a user token will be saved on the client side.

Outputs	On successful/valid input of password and email user will be redirected to list task screen.
---------	--

b. Register

b.1 Description and Priority

The register form is used by all the users. This module has the second highest priority when compared to all the other modules. This model allows the user to enter his email id and password in order to make identity for the use of the software.

b.2 Stimulus/Response Sequences

This screen provides the text boxes for the user to enter credentials or connect with their google account.

b.3 Functional Requirements

On valid input of email and password, password will be hashed (encrypted) and then stored in the online db and corresponding user token will be saved on the client side.

REQ1: The user must have an email id.

Table 2: Functional Requirements for User SignUp Screen

Purpose	This screen provides the interface for the user to enter credentials or connect with their google account.
Inputs	Email, password, re-enter password or Signup using google

Processing	On valid input of email and password, password will be hashed (encrypted) and stored in the online db.
Outputs	On valid input of password and email user will be redirected to list task screen.

c. List Task

c.1 Description and Priority

This screen will be used by users to list tasks and their time limits, priority, focus mode (if exists).

c.2 Stimulus/Response Sequences

This screen Provides a button for creating tasks which will be redirected to the task addition screen.

c.3 Functional Requirements

On clicking each task further details of the task will be shown, on click add task button Add task screen will be shown.

REQ1: Users must be logged in and have a task.

Table 3: Functional Requirements for List Task

Purpose	This will list tasks and their time limits, priority, focus mode (if exists)
Inputs	A button for creating tasks which will be redirected to the task addition screen.

Processing	It will contact db and get all tasks, starting time of task (if exists).
Outputs	On clicking each task further details of the task will be shown.

d. Add Task

d.1 Description and Priority

This screen will be used by users to add tasks and their time limits, priority, focus mode (if exists).

d.2 Stimulus/Response Sequences

This screen Provides following things :-

- Input Box - for adding text input
- Check list button - for adding checklist
- Audio Button - to attach audio
- Image Button - for adding an image as an attachment in the task.
- Speech to Text Button - for converting user's speech to text as a task.
- Priority Number - Add priority number to prioritise the task.
- Alarm / Reminder Time input - to enable alarm/reminder
- Done button - Over the task submission

d.3 Functional Requirements

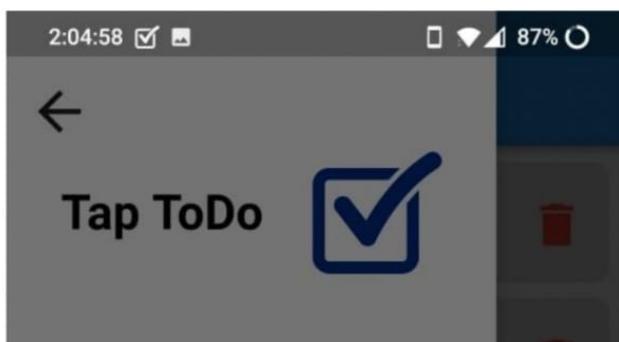
This screen will add the task as per the extensions the user chooses in db in a specific format.

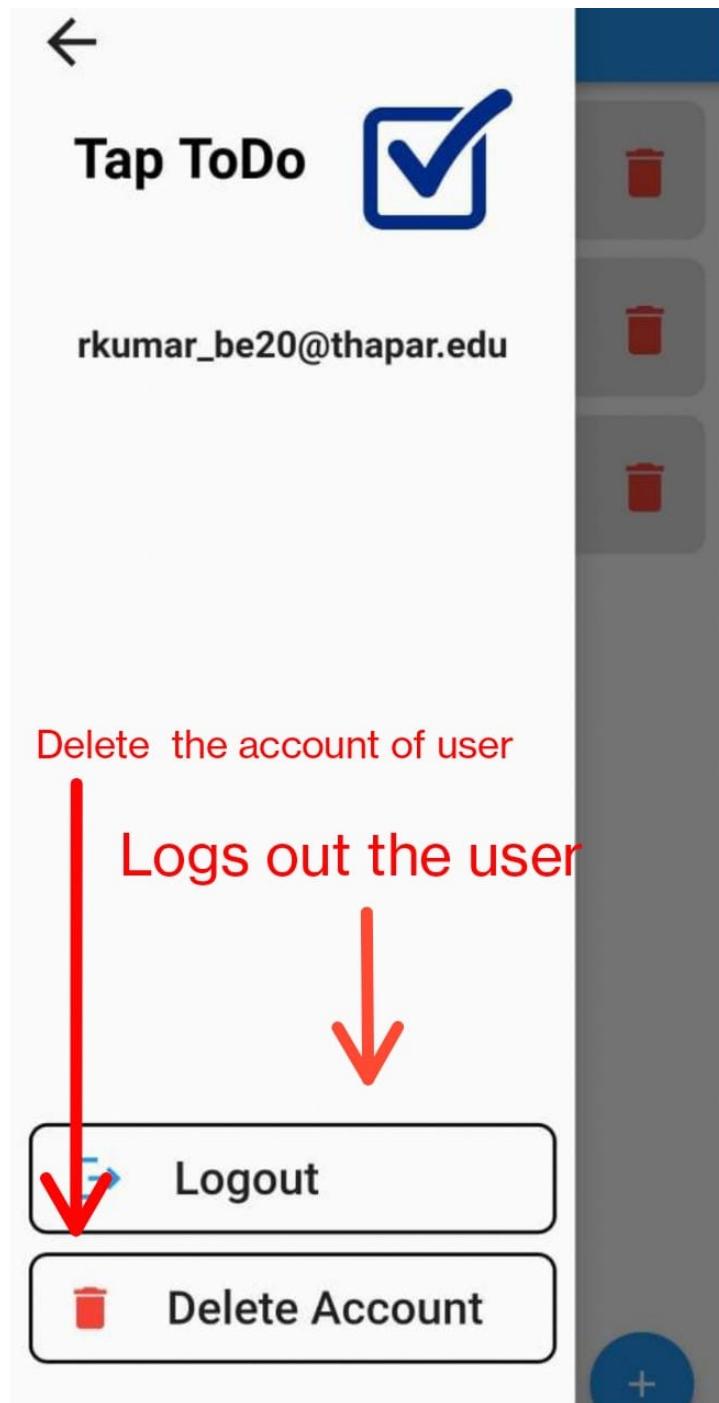
REQ1: User must be logged in.

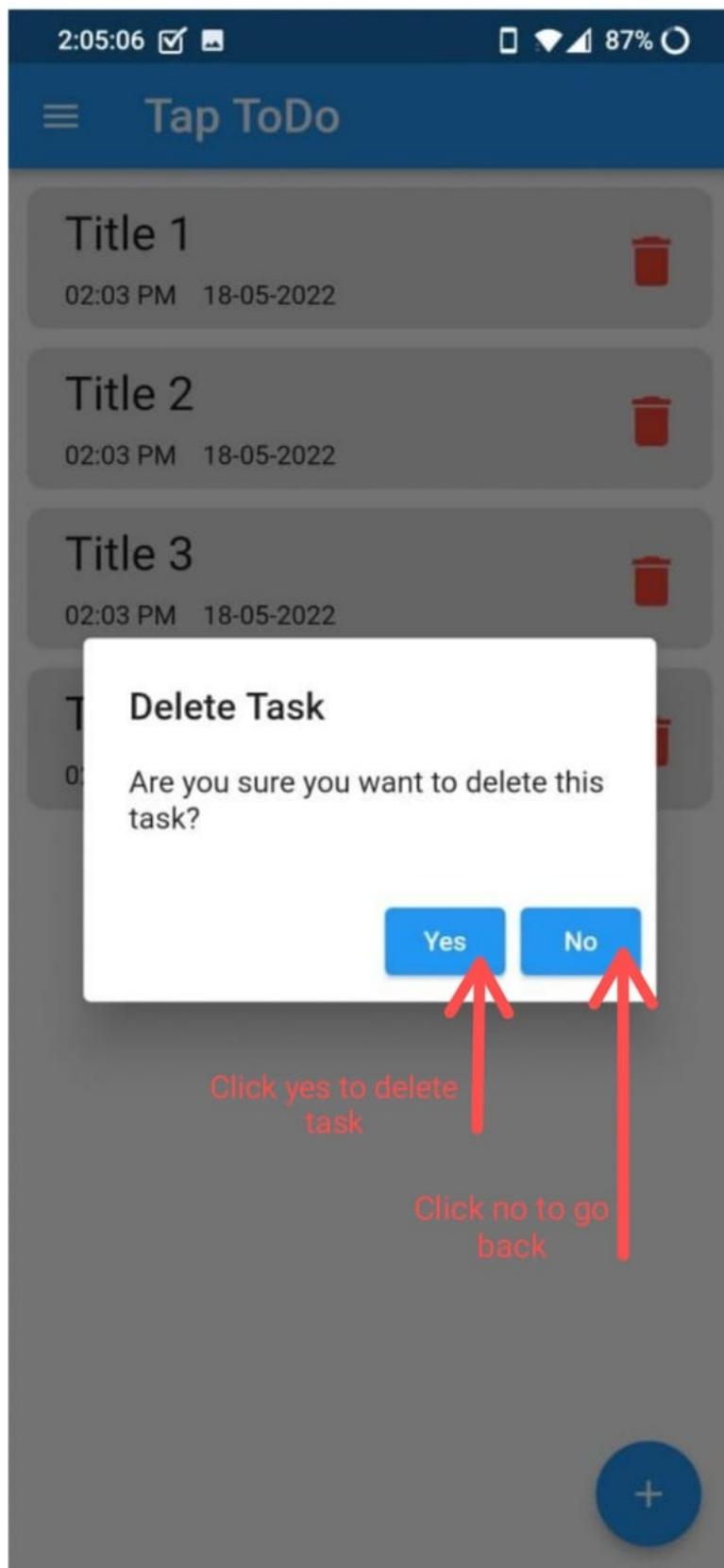
Table 4: Functional Requirements for Add Task

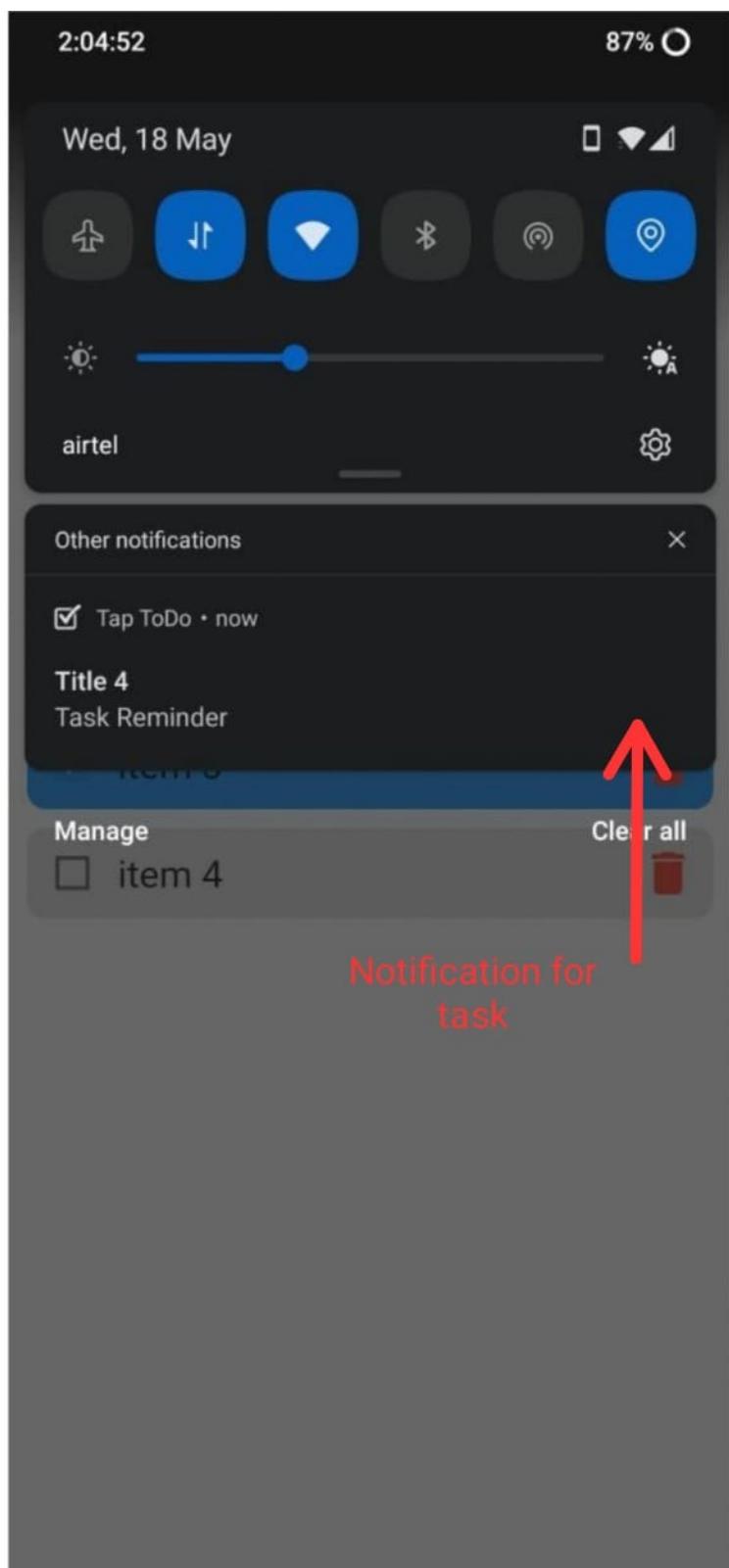
Purpose	This screen will be used to add task.
Inputs	<p>Input Box - for adding text input</p> <p>Check list button - for adding checklist</p> <p>Audio Button - to attach audio</p> <p>Image Button - for adding an image as an attachment in the task.</p> <p>Speech to Text Button - for converting user's speech to text as a task.</p> <p>Priority Number -</p> <p>Add priority number to prioritise the task.</p> <p>Alarm / Reminder Time input - to enable alarm/reminder.</p> <p>Done button - Over the task submission.</p>
Processing	This screen will add the task as per the extensions the user chooses in db in a specific format.
Outputs	On clicking the submit/done button it will redirect to the list task screen.

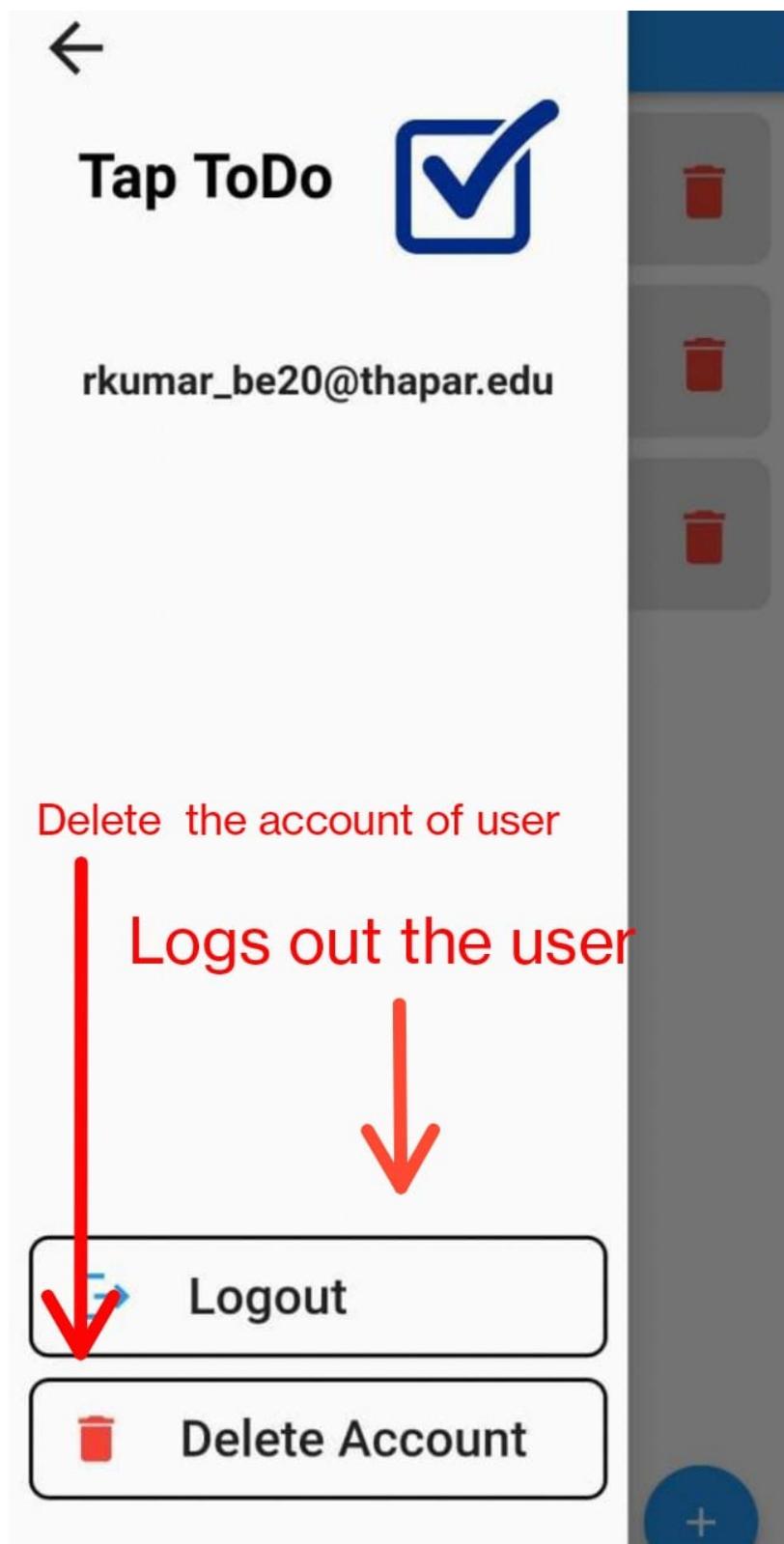
User Story Card

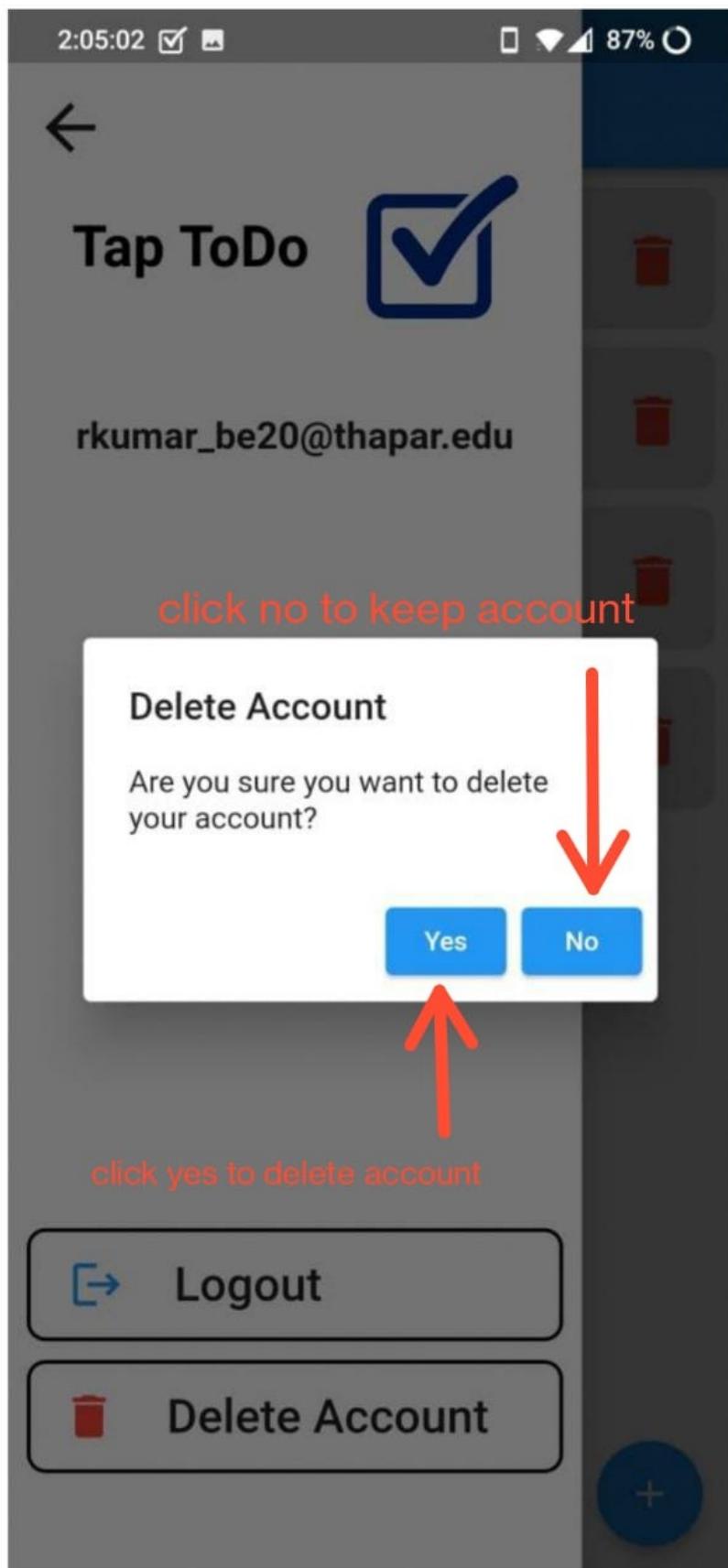


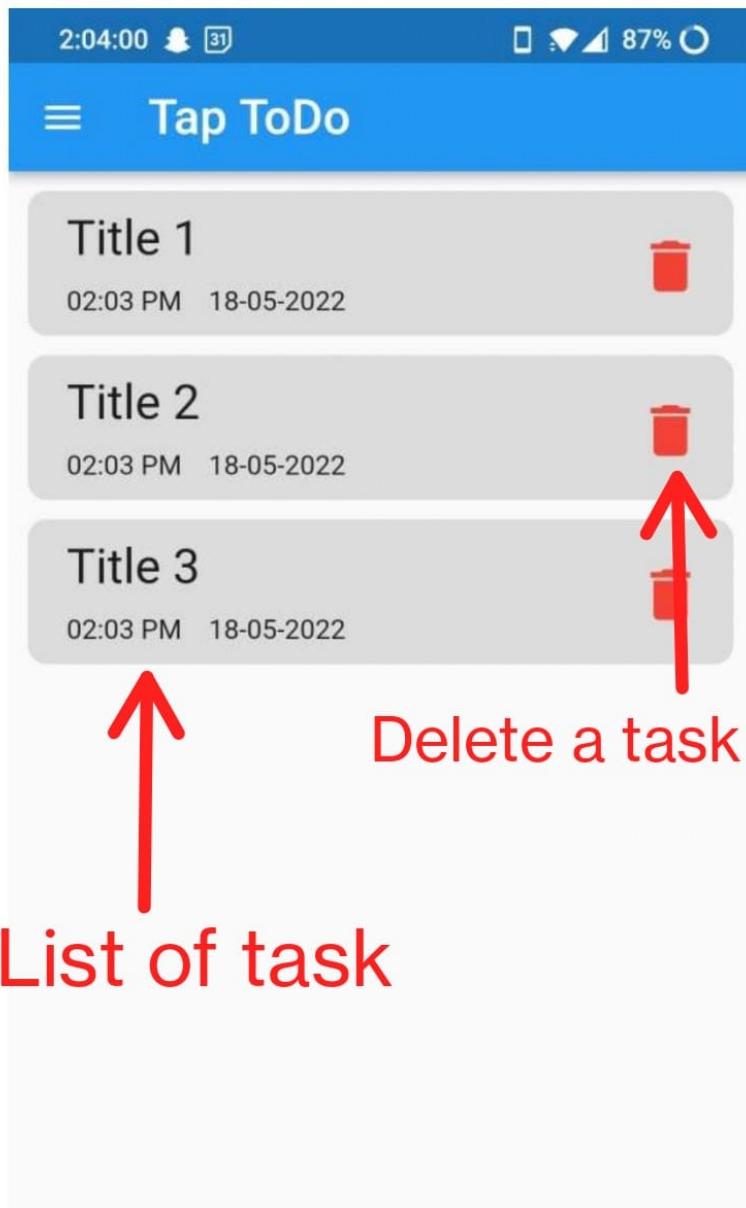


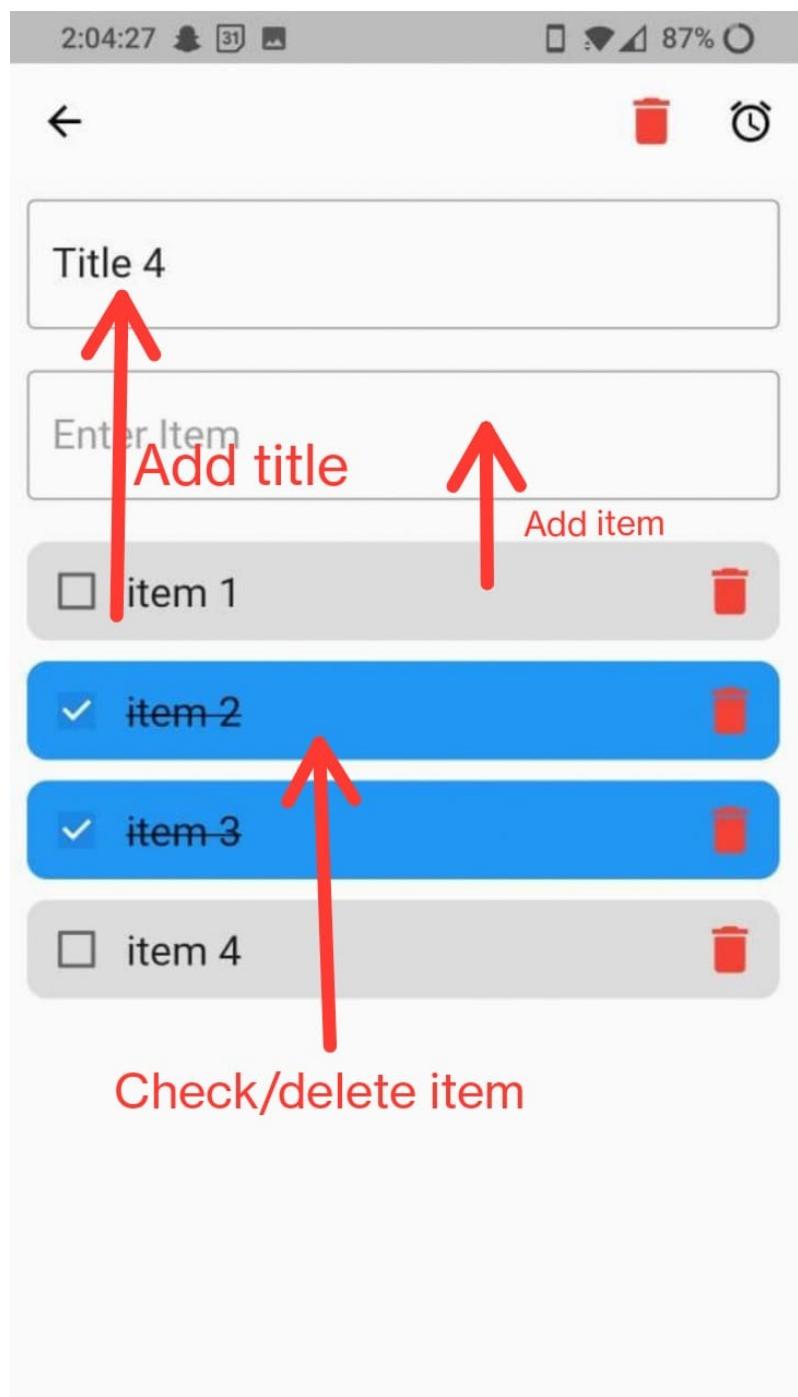






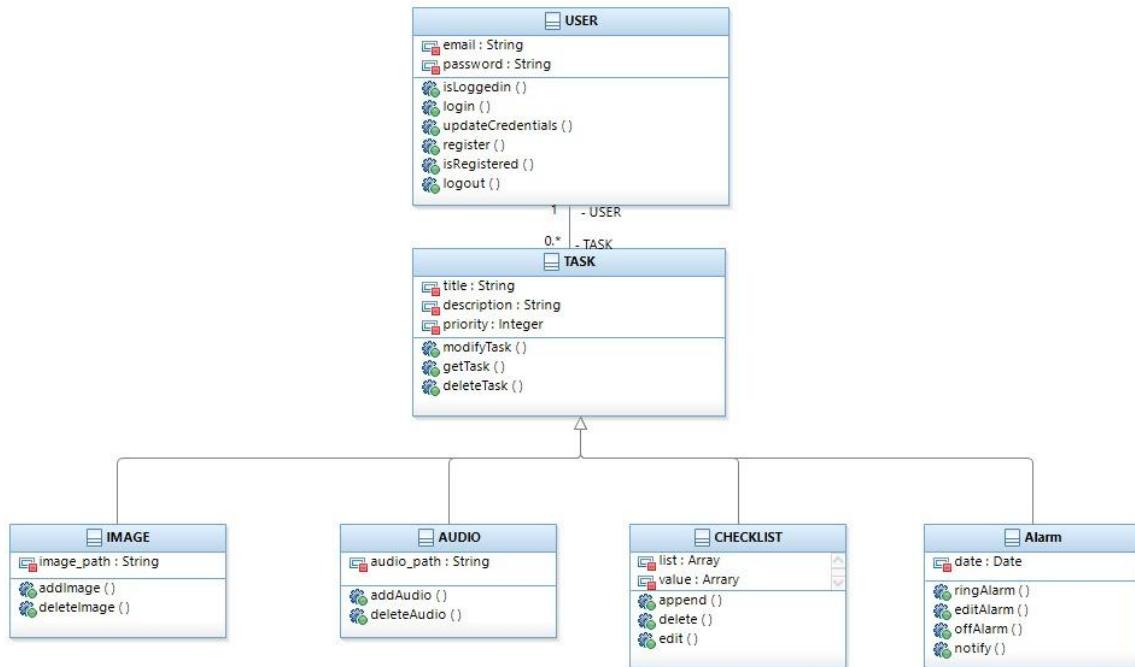




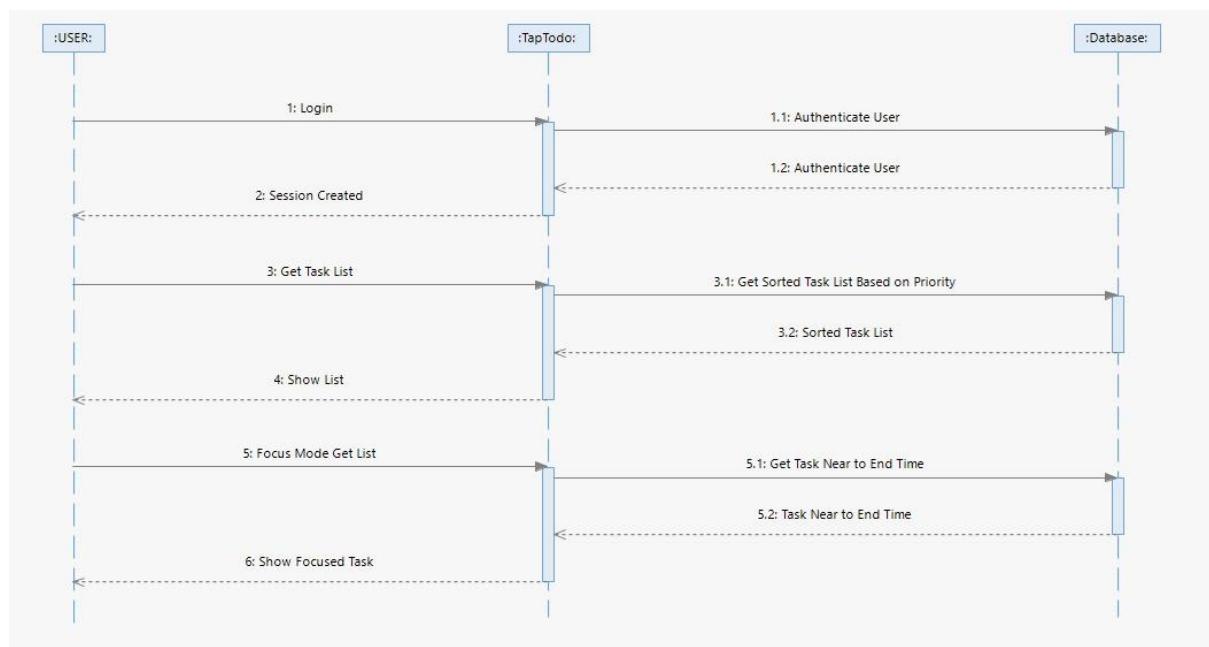


Design Phase

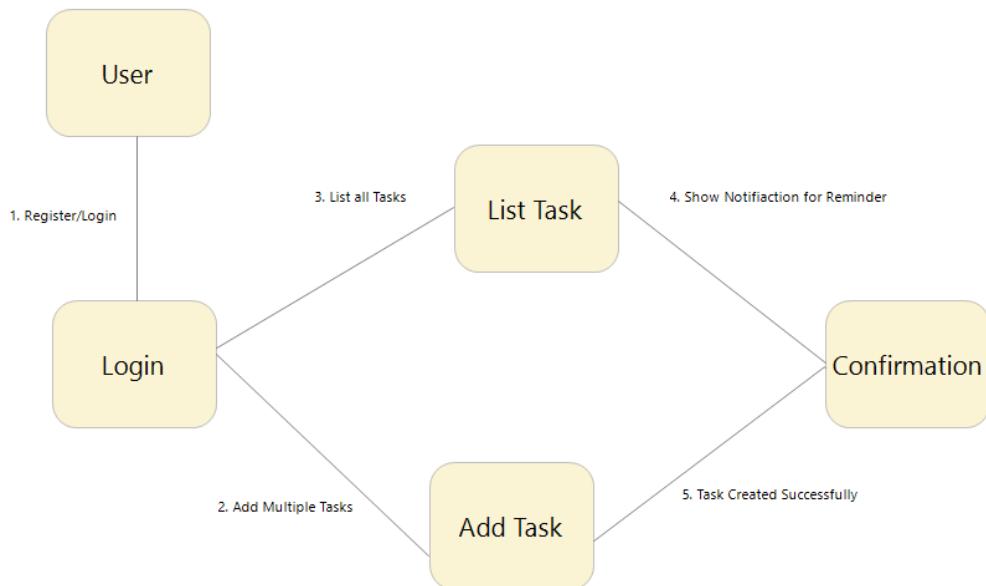
Class Diagram



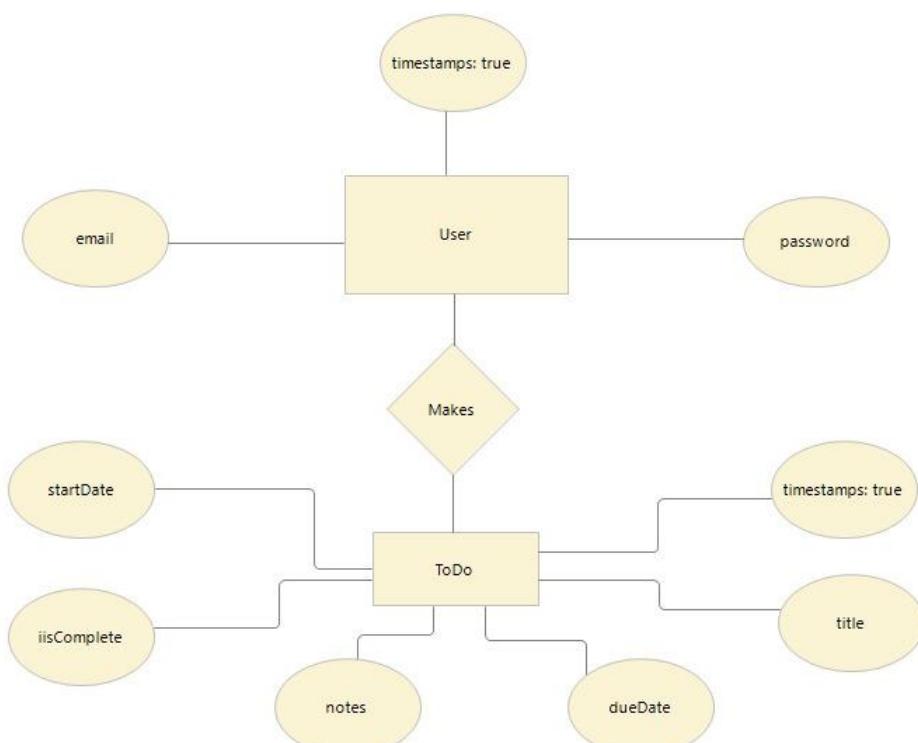
Sequence Diagram



Collaboration Diagram



ER Diagram

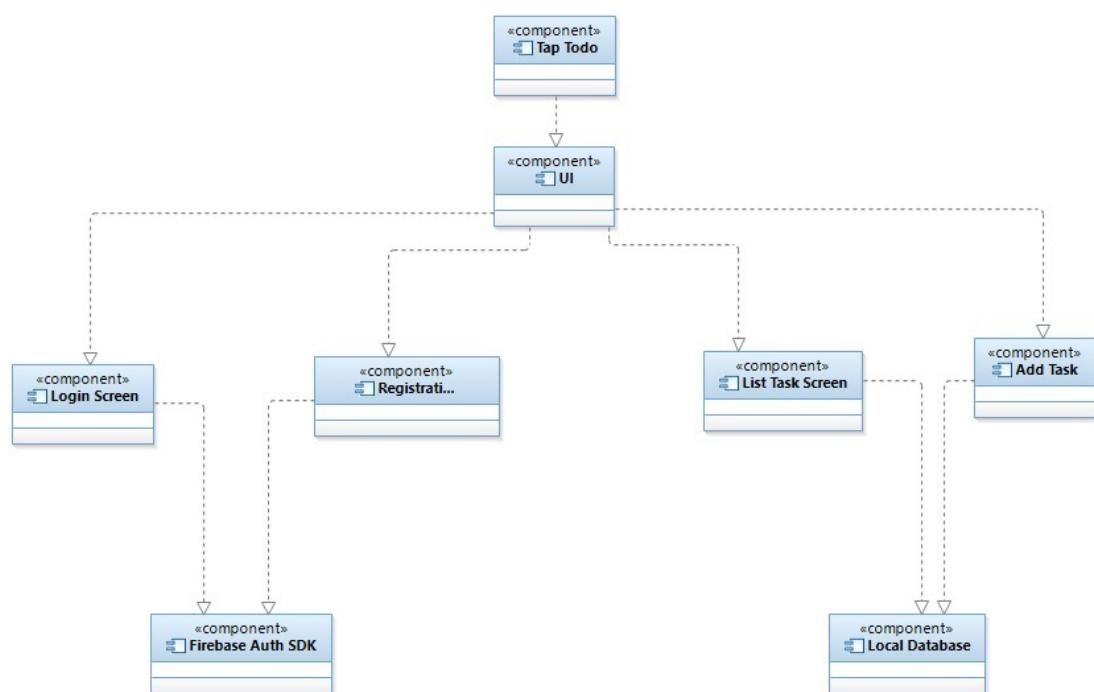


Implementation

Deployment Diagram



Component Diagram



Screenshots of working project:-

Splash screen



Welcome screen



Tap ToDo 

Log In

Register

Login and Register screen



Tap ToDo A large blue checkmark icon with a white interior, positioned next to the app name.

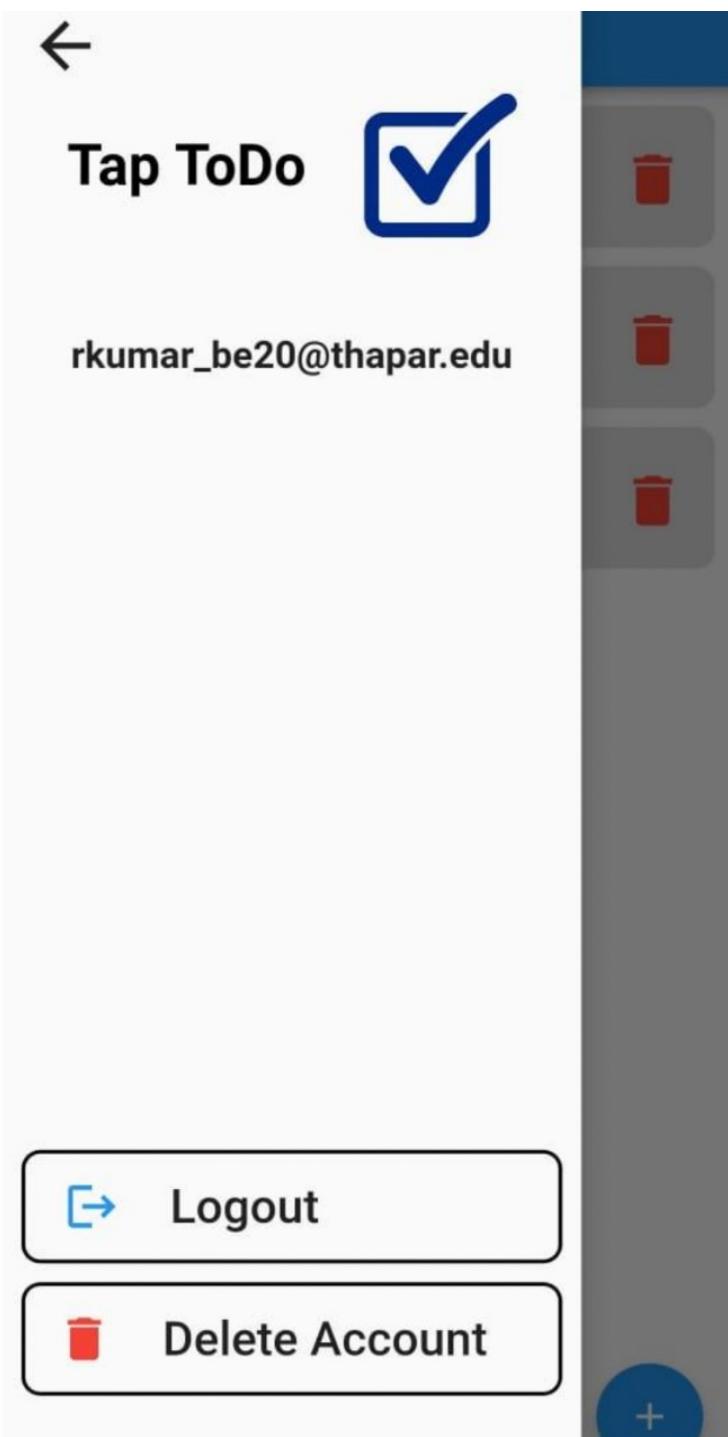
Log In to your account

Enter your email

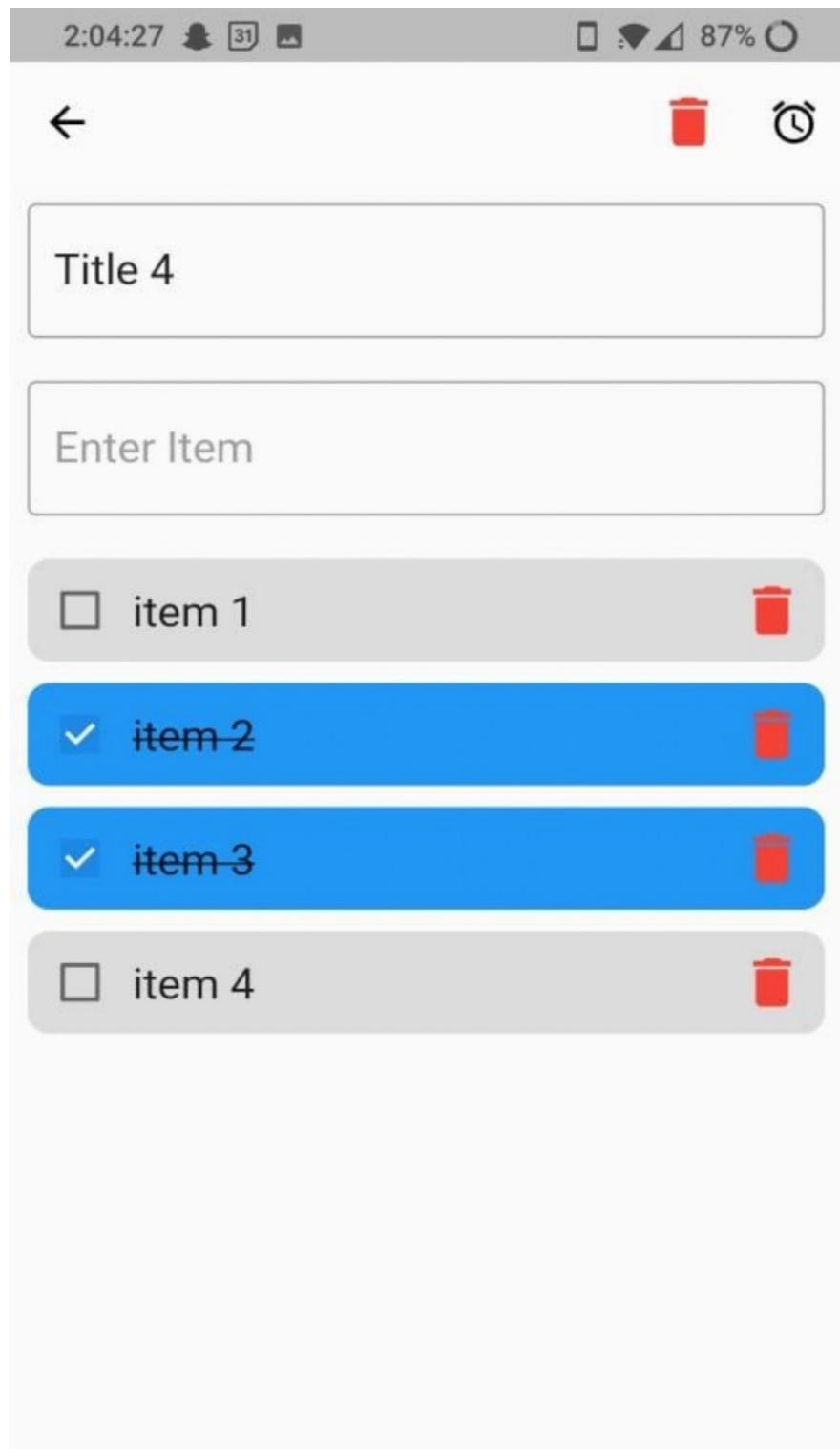
Enter your Password

Log In

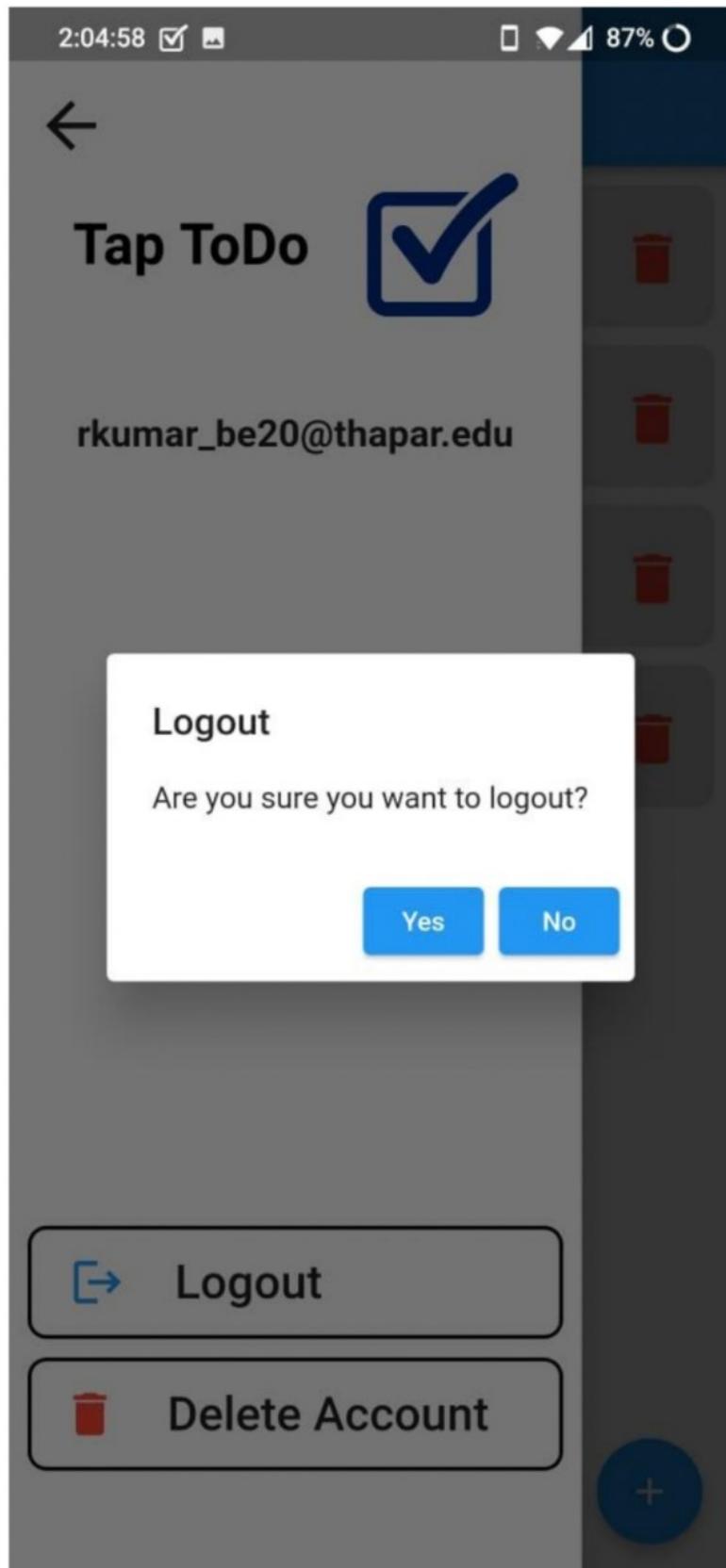
App Drawer



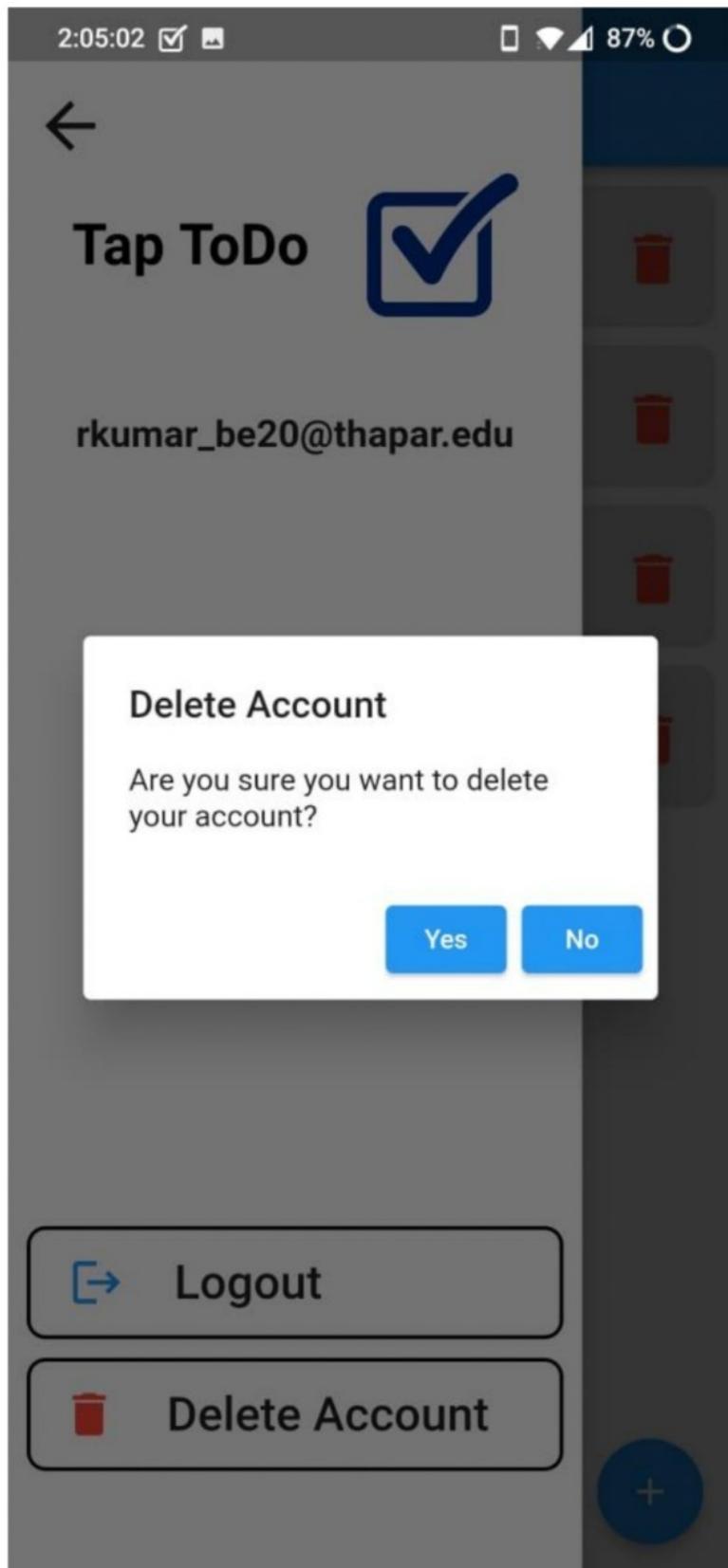
Task Screen



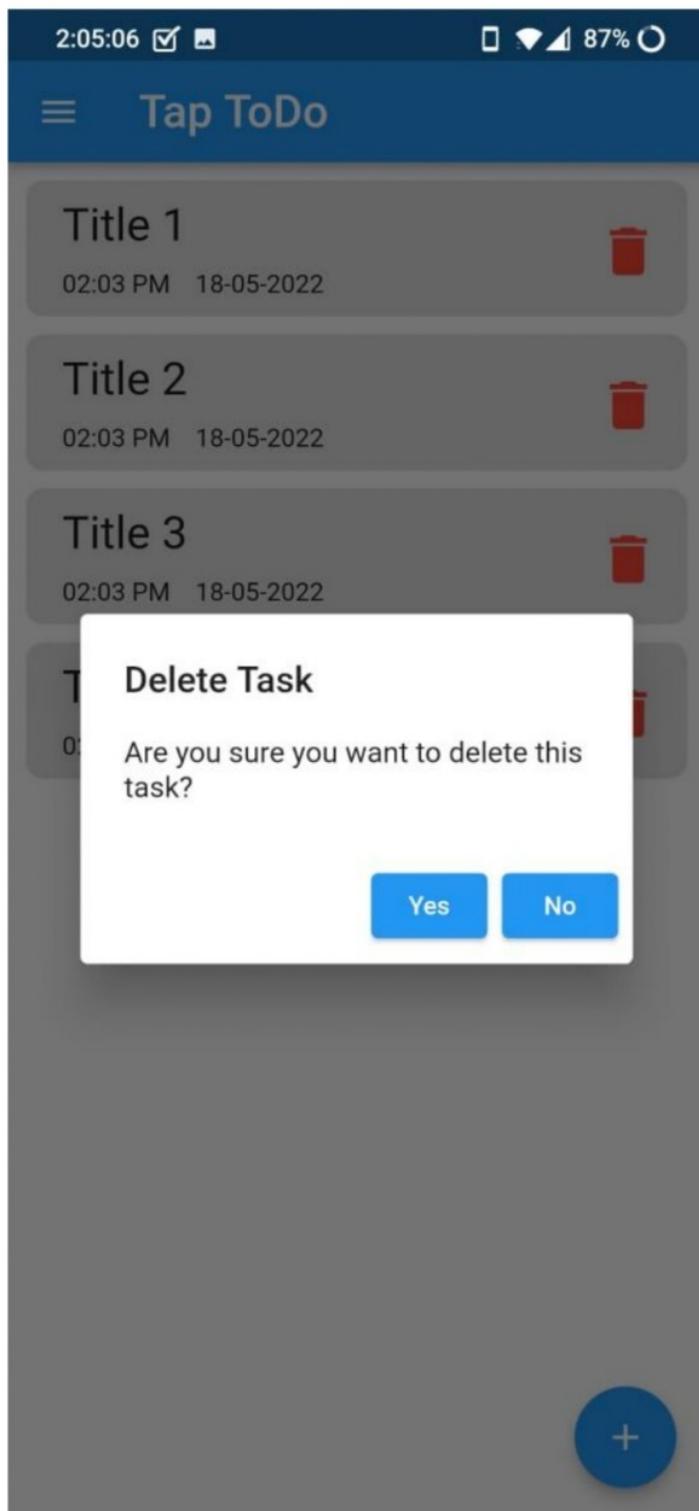
Logout popup



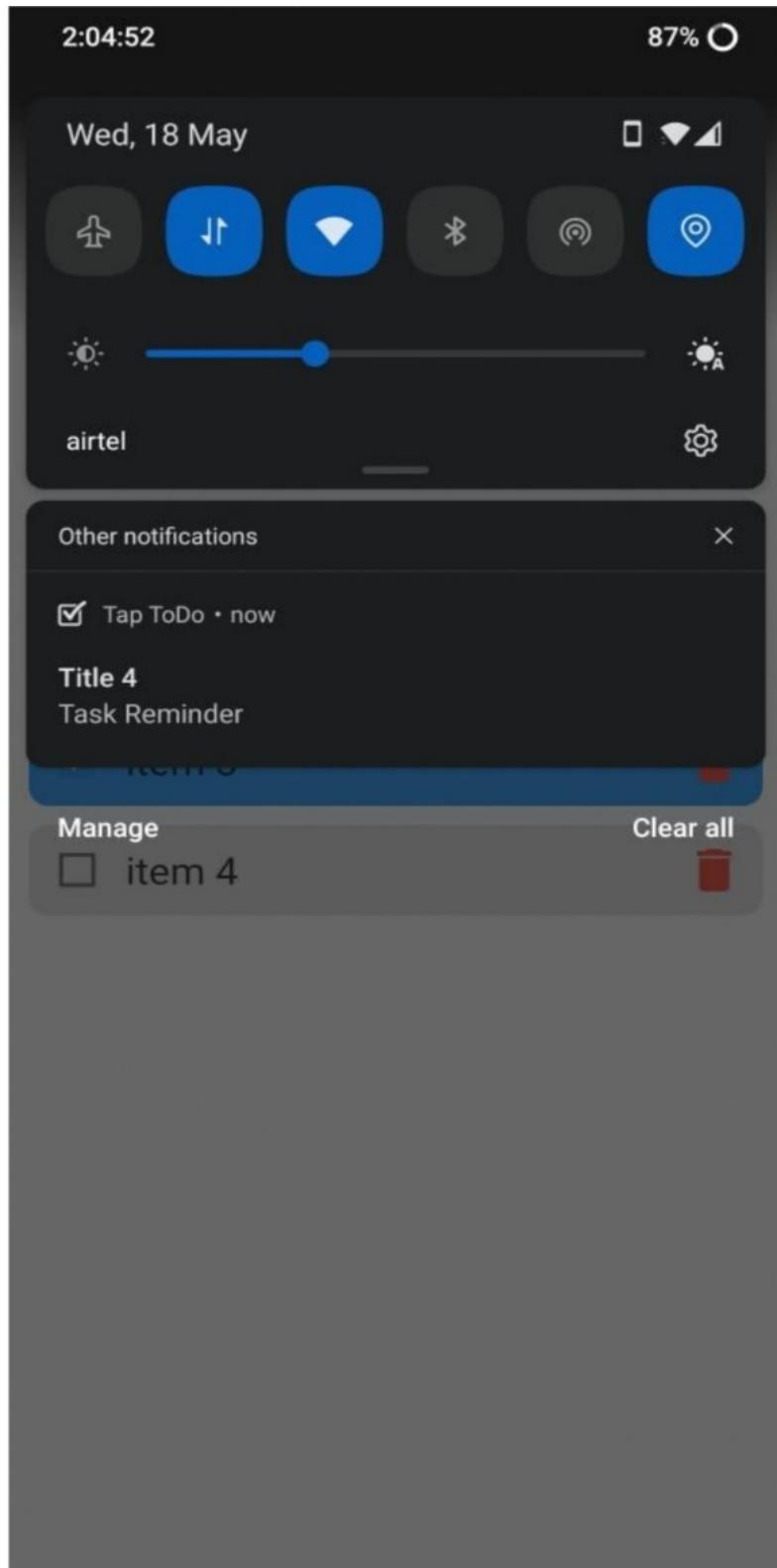
Delete account popup



Delete task popup



Notification message for reminder



Testing:-

Test Cases

Test Case 1

Page: 1 of 1

TEST CASE: - 1

SYSTEM: - **TAP TODO**

TEST CASE NAME: -**User Registration**

Subsystem: - **Registration**

S I

Designed by: - **Meharamt Singh, Harshita Pandey,**

Design Date: -**13/11/2022**

Anannya Singh, Chahat Joneja (Team - 2)

Executed by: - **Team -5**

Execution Date: - **14/11/2022**

Short Description: - Test the register and login service of the user.

Pre-Condition

The user must have a valid email id.

The user must have a secure password.

STEP	ACTION	EXPECTED SYSTEM RESPONSE	PASS/FAIL	COMMENT
1	Click the register button.	The system displays the user registration option asking for the user's details.	PASS	Registration option was displayed, and the user can enter the details..
2	Enter the details and click the register button.	The system displays the user registered successfully.	PASS	Registration successful.
2.1	User has entered an already registered email id	The system shows an error message that the email already exists	PASS	User exists.
2.2	User enters a very short password.	The system shows an error message that the password should be of a minimum of 6 characters for security purposes.	PASS	Password security exists.

Post-Condition

User has been successfully registered and can now log in and use the app for adding tasks.

Test Case 2

Page: 1 of 1

TEST CASE: - 2

SYSTEM: - TAP TODO

TEST CASE NAME: -Add Task S I

Subsystem: - Addition of tasks

Designed by: - Meharamt Singh, Harshita Pandey,

Design Date: - 13/11/2022

Anannya Singh, Chahat Joneja (Team - 2)

Executed by: - Team -5

Execution Date: - 14/11/2022

Short Description: - Test the addition and deletion of tasks for the user.

Pre-Condition

No Pre-Condition.

STEP	ACTION	EXPECTED SYSTEM RESPONSE	PASS/FAIL	COMMENT
1	Click on add task icon.	Text box displayed for adding tasks.	PASS	Text box displayed.
2	Enter the task requirements.	No response.	PASS	Details entered successfully.
3	Click enter task button.	The request shows up on the dashboard.	PASS	Task created successfully.

Post-Condition

Task has been successfully created and shows up on dashboard.

Test Case 3

Page: 1 of 1

TEST CASE: - 2

SYSTEM: - TAP TODO

TEST CASE NAME: -Login S 1

Subsystem: - Login in the app

Designed by: - Meharamt Singh, Harshita Pandey,

Design Date: -13/11/2022

Anannya Singh, Chahat Joneja (Team - 2)

Executed by: - Team -5

Execution Date: - 14/11/2022

Short Description: - Test the login service of the user.

Pre-Condition

Users must register themselves first.

The user must remember the registration password.

STEP	ACTION	EXPECTED SYSTEM RESPONSE	PASS/FAIL	COMMENT
1	Click on the login button.	The system displays the user login option asking for details.	PASS	Login option was displayed, and the user can enter login details.
2	Enter the details and click the login button.	The system displays user logged in successfully.	PASS	Login successful.
2.1	Users enter the wrong password/email.	The system shows an error message that the login password/email is wrong.	PASS	Error displayed.

Post-Condition

User has logged in successfully and can now use the app.

Test Case 4

Page: 1 of 1

TEST CASE: - 4

SYSTEM: - TAP TODO

Designed by: - Meharamt Singh, Harshita Pandey,

Anannya Singh, Chahat Joneja (Team - 2)

Executed by: - Team -5

TEST CASE NAME: -Delete Task

S I

Subsystem: - Delete task from the list

Design Date: -13/11/2022

Execution Date: - 14/11/2022

Short Description: - Test the login service of the user.

Pre-Condition

Users must log in to the app.

STEP	ACTION	EXPECTED SYSTEM RESPONSE	PASS/FAIL	COMMENT
1	Click on the task you wish to delete.	The system displays the tasks in that topic.	PASS	Tasks displayed.
2	Click on the delete task button.	The system shows up a prompt to confirm if the user wants to delete the task.	PASS	Prompt shown.
3	Click on the yes/no button.	The system successfully deletes the task if the user clicks yes, and returns to the previous screen if the user clicks no.	PASS	Task can be deleted.

Post-Condition

Users can delete the tasks if they wish to do so.

Test Case 5

Page: **1 of 1**

TEST CASE: - 5

SYSTEM: - TAP TODO

TEST CASE NAME: - Logout

S I

Subsystem: - Logout from the app

Designed by: - Meharamt Singh, Harshita Pandey,

Design Date: - 13/11/2022

Anannya Singh, Chahat Joneja (Team - 2)

Executed by: - Team -5

Execution Date: - 14/11/2022

Short Description: - Test the login service of the user.

Pre-Condition

Users must log in to the app.

STEP	ACTION	EXPECTED SYSTEM RESPONSE	PASS/FAIL	COMMENT
1	Click on the menu bar button.	The system displays two options asking to either logout or delete the account.	PASS	Menu bar displayed.
2	Click on the logout button.	The system shows a prompt to confirm if the user wants to logout.	PASS	Prompt is shown.
3	Click on the yes/no button.	The system successfully logsouts the user if the user clicks yes, and returns to the previous screen if the user clicks no.	PASS	Users can logout from the app.

Post-Condition

Users can logout from the app.

Test Case 6

Page: 1 of 1

TEST CASE: - 6

SYSTEM: - TAP TODO

TEST CASE NAME: -Delete Account S I

Subsystem: - Delete account from the app

Designed by: - Meharamt Singh, Harshita Pandey,

Design Date: - 13/11/2022

Anannya Singh, Chahat Joneja (Team - 2)

Executed by: - Team -5

Execution Date: - 14/11/2022

Short Description: - Test the login service of the user.

Pre-Condition

Users must register themselves first / already have an account.

The user must log in to the app to delete the account.

STEP	ACTION	EXPECTED SYSTEM RESPONSE	PASS/FAIL	COMMENT
1	Click on the menu bar button.	The system displays 2 options asking to logout/delete the account.	PASS	Menu bar displayed.
2	Click on the delete account button.	The system shows up a prompt to confirm if the user wants to delete the account.	PASS	Prompt shown.
3	Click on the yes/no button.	The system successfully deletes the account if the user clicks yes, and goes back to the previous screen if the user clicks no.	PASS	Account can be deleted.

Post-Condition

Users can delete their accounts if they wish to do so.