



**BSc (Hons) in Information Technology Specializing in Software Engineering
Year 2 - 2022**

**IT2010 – Mobile Application Development
Group Project (Phase 02): Technical Report**

Group Name: Empress Code

Group Member Details

Student ID number	Name	SLIIT Email
IT21060694	Weerasooriya K.A	it21060694@my.sliit.lk
IT21037610	Wanniarachchi W.D..M	it21037610@my.sliit.lk
IT21073564	Priyabandu H.K.S.N	it21073564@my.sliit.lk
IT21073496	Nadeera A.A.D.P	it21073496@my.sliit.lk

Git Repository: <https://github.com/anusaraweerasooriya/MAD-Project.git>

Social Media video:

Facebook: <https://www.facebook.com/100007353428256/videos/366068215681145/>

LinkedIn: https://www.linkedin.com/posts/kavindu-anusara-weerasooriya-7973b9246_androiddevelopment-android-appdevelopment-activity-6992502145056919552-sCyF?utm_source=share&utm_medium=member_desktop

(When sharing the social media video, make sure that it is publicly available)

Group Presentation Video:

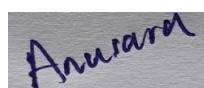
https://mysliit.sharepoint.com/sites/MADProjectGroup-Y2S2/_layouts/15/stream.aspx?id=%2Fsites%2FMADProjectGroup%2DY2S2%2FShared%20Documents%2FGeneral%2FRecordings%2FIT2110%20%2D%20MAD%20Group%20Project%20%28EmpressCode%29%2D20221030%5F164801%2DMeeting%20Recording%2Emp4&referrer=Teams%2ETEAMS%2DELECTRON&referrerScenario=teams%5Fns%2Dmw&ga=1

Project Start Date: 12th of September 2022

Project End Date: 30th of October 2022

Project Declaration:

I the undersigned solemnly declare that the project technical based on our group members own work carried out during the course of our study under the supervision of the lecturer.

A handwritten signature in blue ink, which appears to read "Anusara".

Group Leader

Marking Rubric

Date		Student 1	Student 2	Student 3	Student 4
Student ID		IT21060694	IT21037610	IT21073564	IT21073496
Description	Marks				
Integrated system using a repository	5				
Marketing the app using a video clip published in a social media account (e.g.: LinkedIn)	10				
CRUD Function completeness	25				
Database connectivity	5				
Usability	10				
Good coding practice	5				
Out of box features	10				
Test cases	10				
Report	20				
Total	100				

Report Template.

1. Component Description. Describe the usage of crud operations (Member wise)
2. Screenshots of the Mobile Application (Member wise)
3. When you write the report, create topics as given in the report template and subtopics from the member's SLIIT Student ID number. Please refer to the marking rubric as well.

1. Introduction

Most people pay less attention on writing down important things such as diaries, lists, due tasks and special notes which are most likely to be misplaced or forgotten easily. "Empress Notes" is a mobile application which can be introduced as a great solution for such occasions. Main focus of this application is to allow users to store all kind of notes in one place which can be manipulated easily by just one click. In our mobile application there are many useful functionalities and different note taking options. Those options are,

1. My Diary - IT21060694

In terms of flexibility and convenience, this type of digital diary is frequently regarded as superior to paper calendars or daily planner books.

2. My To-Do List – IT21037610

To-do list app make it simple to create a new task that needs to be completed. You do not have to enter the tasks in any particular order. To-do list app allow you to keep track of daily tasks and easily sort them according to your preferences.

3. My Sticky Notes - IT21073564

A digital sticky note app is useful when a user needs to get a note of something urgently. User can store all important notes in a flexible manner.

4. My Lists - IT21073496

A list of items such as grocery lists, shopping lists and book lists stored in the mobile is a flexible and convenient option for the busy lifestyle of people. Most importantly digital lists can be updated easily whenever necessary.

User friendliness and user experience is highly valued in designing the app interfaces and navigation activities throughout the entire mobile application. When the app is launched user will be able to view a flash screen containing the app logo and immediately be navigated to the home page of the application. In here the users can easily select the required type of notes from the available card options.

2. Database Connectivity

I. IT21060694 – Weerasooriya K.A

```
// my_diary table columns
private static final String COLUMN_DIARY_ID = "id";
private static final String COLUMN_DIARY_TITLE = "title";
private static final String COLUMN_DIARY_DATETIME = "dateTime";
private static final String COLUMN_DIARY_BODY = "diaryBody";
private static final String COLUMN_DIARY_IMAGE_PATH = "imagePath";

// My Diary-----
String my_diary_query = " CREATE TABLE " + TABLE_NAME1 +
    "(" + COLUMN_DIARY_ID + " INTEGER PRIMARY KEY AUTOINCREMENT, " +
    COLUMN_DIARY_TITLE + " TEXT, " +
    COLUMN_DIARY_DATETIME + " TEXT, " +
    COLUMN_DIARY_BODY + " TEXT, " +
    COLUMN_DIARY_IMAGE_PATH + " TEXT);";

sqLiteDatabase.execSQL(my_diary_query);

// My Diary
sqLiteDatabase.execSQL(" DROP TABLE IF EXISTS " + TABLE_NAME1);
onCreate(sqLiteDatabase);
```

II. IT21037610 – Wanniarachchi W.D.D.M

```
// my_todo table columns
private static final String COLUMN_TASK_ID = "id";
private static final String COLUMN_TASK_TITLE = "title";
private static final String COLUMN_TASK_DESCRIPTION = "description";
private static final String COLUMN_TASK_DATE = "date";
private static final String COLUMN_TASK_TIME = "time";
private static final String COLUMN_TASK_URL = "url";
private static final String COLUMN_TASK_STATUS = "status";

// My To-Do-----
String todo_query = " CREATE TABLE " + TABLE_NAME2 +
    "(" + COLUMN_TASK_ID + " INTEGER PRIMARY KEY AUTOINCREMENT, " +
    COLUMN_TASK_TITLE + " TEXT, " +
    COLUMN_TASK_DESCRIPTION + " TEXT, " +
    COLUMN_TASK_DATE + " TEXT, " +
    COLUMN_TASK_TIME + " TEXT, " +
    COLUMN_TASK_URL + " TEXT, " +
    COLUMN_TASK_STATUS + " TEXT DEFAULT 'UPCOMING');";

sqLiteDatabase.execSQL(todo_query);

// My To-Do
sqLiteDatabase.execSQL(" DROP TABLE IF EXISTS " + TABLE_NAME2);
onCreate(sqLiteDatabase);
```

III. IT21073564 - Priyabandu H.K.S.N

```
// my_notes table columns
private static final String COLUMN_NOTES_ID = "id";
private static final String COLUMN_NOTES_TITLE = "noteTitle";
private static final String COLUMN_NOTES_BODY = "note";
private static final String COLUMN_NOTES_DATETIME = "datetime";
private static final String COLUMN_NOTES_COLOR = "color";
```

```
// My Notes-----
String notes_query = "CREATE TABLE " + TABLE_NAME3 +
    "(" + COLUMN_NOTES_ID + " INTEGER PRIMARY KEY AUTOINCREMENT, " +
    COLUMN_NOTES_TITLE + " TEXT, " +
    COLUMN_NOTES_BODY + " TEXT, " +
    COLUMN_NOTES_DATETIME + " TEXT, " +
    COLUMN_NOTES_COLOR + " TEXT);";

sqLiteDatabase.execSQL(notes_query);
```

```
// My Notes
sqLiteDatabase.execSQL("DROP TABLE IF EXISTS " + TABLE_NAME3);
onCreate(sqLiteDatabase);
```

IV. IT21073496 - Nadeera A.A.D.P

```
// my_lists table columns
private static final String COLUMN_LIST_ID = "id";
private static final String COLUMN_LIST_Title = "title";
private static final String COLUMN_LIST_Description = "description";
private static final String COLUMN_LIST_Quantity = "quantity";
```

```
// my_sub_list table columns
private static final String COLUMN_my_sub_list_ID = "id";
private static final String COLUMN_my_sub_list_mainID = "m_id";
private static final String COLUMN_my_sub_list_Title = "title";
private static final String COLUMN_my_sub_list_Description = "description";
private static final String COLUMN_my_sub_list_Quantity = "quantity";
```

```

// My Lists-----
String my_list_query = " CREATE TABLE " + TABLE_NAME4 +
    "(" + COLUMN_List_ID + " INTEGER PRIMARY KEY AUTOINCREMENT, " +
    COLUMN_List_Title + " TEXT, " +
    COLUMN_List_Description + " TEXT, " +
    COLUMN_List_Quantity + " INTEGER);";

sqLiteDatabase.execSQL(my_list_query);

// Sub List-----
String my_sub_list_query = " CREATE TABLE " + TABLE_NAME5 +
    "(" + COLUMN_my_sub_list_ID + " INTEGER PRIMARY KEY AUTOINCREMENT, " +
    COLUMN_my_sub_list_mainID + " TEXT, " +
    COLUMN_my_sub_list_Title + " TEXT, " +
    COLUMN_my_sub_list_Description + " TEXT, " +
    COLUMN_my_sub_list_Quantity + " INTEGER);";

sqLiteDatabase.execSQL(my_sub_list_query);

```

```

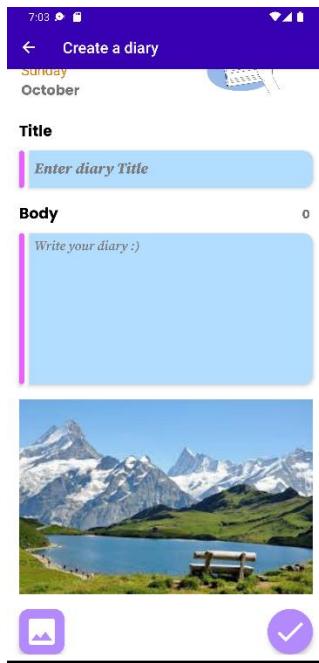
// My Lists
sqLiteDatabase.execSQL(" DROP TABLE IF EXISTS " + TABLE_NAME4);
onCreate(sqLiteDatabase);
// Sub List
sqLiteDatabase.execSQL(" DROP TABLE IF EXISTS " + TABLE_NAME5);
onCreate(sqLiteDatabase);

```

3. CRUD Functions

I. IT21060694 – Weerasooriya K.A

CREATE : The user can add a diary page in “Create a diary” page where he can enter all diary details. Current date will be generated and displayed automatically for user convenience.



```

//===== MY DIARY =====
//Add a new diary-----
public void addDiary(String diaryTitle, String diaryDateTime, String diaryBody, String imagePath){

    SQLiteDatabase dbRef = this.getWritableDatabase();
    ContentValues cv = new ContentValues();

    cv.put(COLUMN_DIARY_TITLE, diaryTitle);
    cv.put(COLUMN_DIARY_BODY, diaryBody);
    cv.put(COLUMN_DIARY_DATETIME, diaryDateTime);
    cv.put(COLUMN_DIARY_IMAGE_PATH, imagePath);

    long result = dbRef.insert(TABLE_NAME1, nullColumnHack: null, cv);
    if (result == -1) {
        Toast.makeText(context, text: "Failed To insert Diary", Toast.LENGTH_SHORT).show();
    } else {
        Toast.makeText(context, text: "Diary Added Successfully", Toast.LENGTH_SHORT).show();
    }
}

```

```

buttonDiarySave.setOnClickListener(new View.OnClickListener() {

    @Override
    public void onClick(View view) {
        if (inputDiaryTitle.getText().toString().trim().isEmpty()) {
            Toast.makeText(context: CreateADiary.this, text: "Note title cannot be empty!", Toast.LENGTH_SHORT).show();
            return;
        } else if (inputDiaryBody.getText().toString().trim().isEmpty()) {
            Toast.makeText(context: CreateADiary.this, text: "Diary Body cannot be empty", Toast.LENGTH_SHORT).show();
            return;
        } else {

            DatabaseHelper myDB = new DatabaseHelper(context: CreateADiary.this);
            myDB.addDiary(inputDiaryTitle.getText().toString().trim(),
                diaryDateTime.getText().toString(),
                inputDiaryBody.getText().toString().trim(),
                selectedDiaryImagePath
            );

            Intent intent = new Intent(packageContext: CreateADiary.this, MyDiary.class);
            startActivity(intent);
        }
    }
});

```

RETRIEVE : Each diary page will be displayed in separate cards where the user will be able to scroll through easily. If there are no saved diaries available at the moment the page will display NO DATA.



```

// Read all the diaries-----
public Cursor readDiaryData() {
    String diary_query = "SELECT * FROM " + TABLE_NAME1;
    SQLiteDatabase dbRef = this.getReadableDatabase();

    Cursor cursor = null;
    if (dbRef != null) {
        cursor = dbRef.rawQuery(diary_query, selectionArgs: null);
    }
    return cursor;
}

void storeDiaryDataInArrays() {
    textDiaryID = findViewById(R.id.textDiaryID);

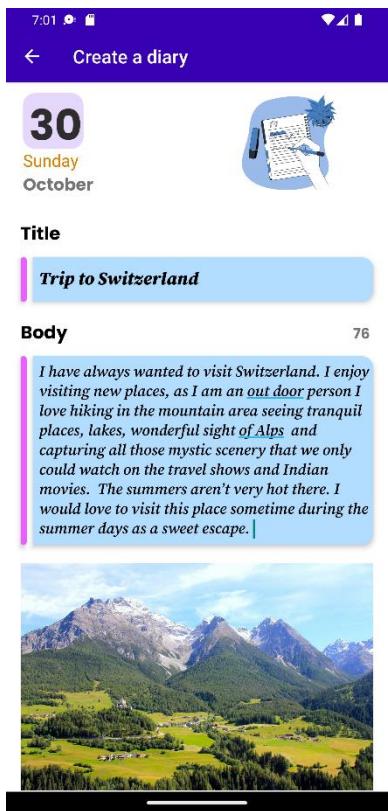
    Cursor cursor = diary_db.readDiaryData();
    if (cursor.getCount() == 0) {

        empty_diaryImageView.setVisibility(View.VISIBLE);
        noDiaryData.setVisibility(View.VISIBLE);

    } else {
        while (cursor.moveToNext()) {
            diary_id.add(cursor.getString(i: 0));
            diary_title.add(cursor.getString(i: 1));
            diary_date.add(cursor.getString(i: 2));
            diary_body.add(cursor.getString(i: 3));
            diary_image.add(cursor.getString(i: 4));
        }
        empty_diaryImageView.setVisibility(View.GONE);
        noDiaryData.setVisibility(View.GONE);
    }
}

```

UPDATE: User will be able to view and update necessary diary details by clicking on the diary page and simply updating necessary fields. he can also upload a new image if needed.



```
//Update diary-----
public void updateDiary(String row_id, String title, String body, String date, String image) {

    SQLiteDatabase dbRef = this.getWritableDatabase();
    ContentValues cv = new ContentValues();
    cv.put(COLUMN_DIARY_TITLE, title);
    cv.put(COLUMN_DIARY_BODY, body);
    cv.put(COLUMN_DIARY_DATETIME, date);
    cv.put(COLUMN_DIARY_IMAGE_PATH, image);

    long result = dbRef.update(TABLE_NAME1, cv, whereClause: "id=?", new String[]{row_id});

    if (result == -1) {
        Toast.makeText(context, text: "Failed to update diary !", Toast.LENGTH_SHORT).show();
    } else {
        Toast.makeText(context, text: "Diary updated successfully!", Toast.LENGTH_SHORT).show();
    }
}
```

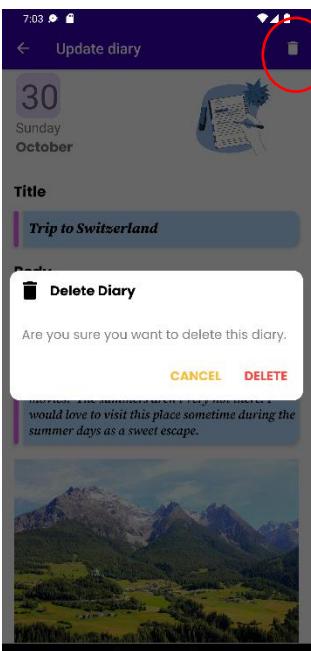
```
updateButton.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View view) {
        DatabaseHelper diaryDB =new DatabaseHelper( context: UpdateDiary.this);
        title = inputDiaryTitle.getText().toString().trim();
        body = inputDiaryBody.getText().toString().trim();
        date = diaryDateTime.getText().toString();
        image = selectedDiaryImagePath;

        diaryDB.updateDiary(id, title, body, date, image);

        Intent intent = new Intent( packageContext: UpdateDiary.this, MyDiary.class);
        startActivity(intent);
    }
});

getAndSetInsertDiaryData();
```

DELETE: User can easily delete a single diary page by selecting the bin icon on the view diary action bar. If he wants to delete all the diary pages in the My Diary, he can select the “Delete All” option from the Diary home page. In either instances user needs to confirm delete action from the alert dialog.



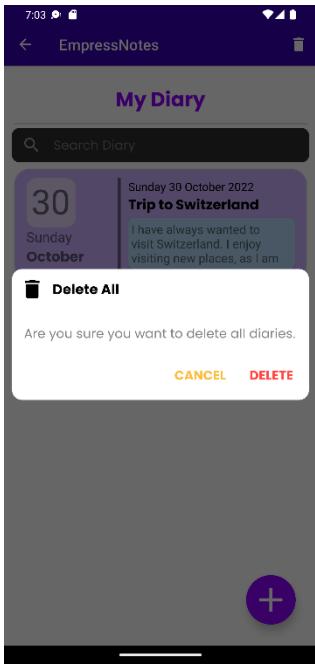
```
//Delete a row of diary-----
public void deleteDiaryRow(String row_id) {

    SQLiteDatabase dbRef = this.getWritableDatabase();
    long result = dbRef.delete(TABLE_NAME1, whereClause: "id=?", new String[]{row_id});

    if (result == -1) {
        Toast.makeText(context, text: "Failed to delete diary!", Toast.LENGTH_SHORT).show();
    } else {
        Toast.makeText(context, text: "Successfully deleted the diary!", Toast.LENGTH_SHORT).show();
    }
}

//Delete all diaries-----
public void deleteAllDiaries() {

    SQLiteDatabase dbRef = this.getWritableDatabase();
    dbRef.execSQL("DELETE FROM " + TABLE_NAME1);
}
```



```

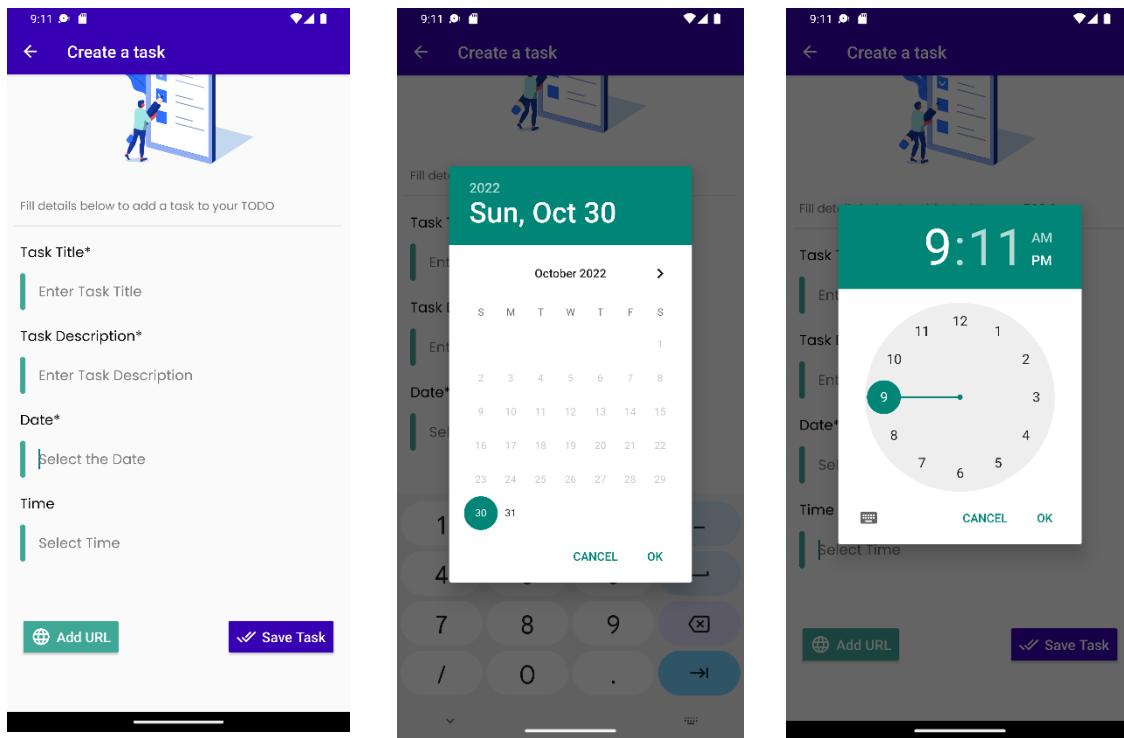
void confirmDeleteDialog() {
    AlertDialog.Builder builder = new AlertDialog.Builder( context: MyDiary.this );
    View view = LayoutInflater.from(this).inflate(
        R.layout.diary_delete_all_layout,
        (ViewGroup) findViewById(R.id.layoutDiaryDeleteAll)
    );
    builder.setView(view);
    AlertDialog deleteAllDialog = builder.create();
    if (deleteAllDialog.getWindow() != null) {
        deleteAllDialog.getWindow().setBackgroundDrawable(new ColorDrawable(0));
    }

    // If user confirm delete action
    view.findViewById(R.id.textDeleteAllDiaryBtn).setOnClickListener(new View.OnClickListener() {
        @Override
        public void onClick(View view) {
            DatabaseHelper diaryDB = new DatabaseHelper( context: MyDiary.this );
            diaryDB.deleteAllDiaries();
            // Refresh and return back to to-do home page
            Intent intent = new Intent( packageContext: MyDiary.this, MyDiary.class );
            startActivity(intent);
        }
    });
    // If user cancel delete action
    view.findViewById(R.id.textCancelDeleteAllDiaryBtn).setOnClickListener(new View.OnClickListener() {
        @Override
        public void onClick(View view) { deleteAllDialog.dismiss(); }
    });
    deleteAllDialog.show();
}

```

II. IT21037610 – Wanniarachchi W.D.D.M

CREATE : The user can add a new task by touching the + floating action button. Then the user will be directed to the “Add Task” page where he can enter all details such as Task Title, Description and select a date and time from Date picker and Time picker respectively. Upon completion user will be navigated back to the View tasks page.



```

//=====MY TO-DO=====
// CREATE A TO-DO TASK
public void createToDoTask(String title, String description, String date, String time, String url) {
    SQLiteDatabase db = this.getWritableDatabase();
    ContentValues cv = new ContentValues();

    cv.put(COLUMN_TASK_TITLE, title);
    cv.put(COLUMN_TASK_DESCRIPTION, description);
    cv.put(COLUMN_TASK_DATE, date);
    cv.put(COLUMN_TASK_TIME, time);
    cv.put(COLUMN_TASK_URL, url);

    long result = db.insert(TABLE_NAME2, nullColumnHack: null, cv);
    if (result == -1) {
        Toast.makeText(context, text: "Failed to create the task", Toast.LENGTH_SHORT).show();
    } else {
        Toast.makeText(context, text: "Task is created successfully!", Toast.LENGTH_SHORT).show();
    }
}

```

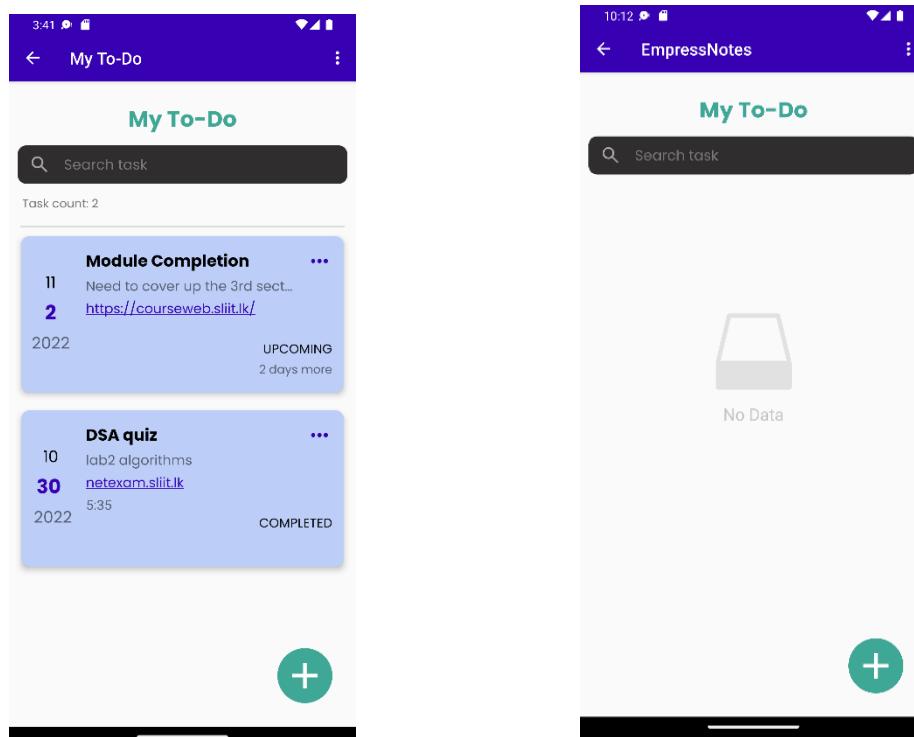
```

save_btn.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View view) {
        if (title_input.getText().toString().trim().isEmpty()) {
            Toast.makeText(context: ToDoCreateTask.this, text: "Task title can not be empty!", Toast.LENGTH_SHORT).show();
            return;
        } else if (description_input.getText().toString().trim().isEmpty()) {
            Toast.makeText(context: ToDoCreateTask.this, text: "Task description can not be empty!", Toast.LENGTH_SHORT).show();
            return;
        } else if (date_input.getText().toString().trim().isEmpty()) {
            Toast.makeText(context: ToDoCreateTask.this, text: "Task date can not be empty", Toast.LENGTH_SHORT).show();
            return;
        } else {
            DatabaseHelper myDB = new DatabaseHelper(context: ToDoCreateTask.this);
            myDB.createToDoTask(title_input.getText().toString().trim(),
                description_input.getText().toString().trim(),
                date_input.getText().toString().trim(),
                time_input.getText().toString().trim(),
                text_url.getText().toString().trim()
            );
        }

        Intent intent = new Intent(packageContext: ToDoCreateTask.this, ToDoMain.class);
        startActivity(intent);
    }
});

```

RETRIEVE : Each task will appear in separate cards where the user will be able to scroll through easily. If there are no saved tasks available at the moment the page will display NO DATA.

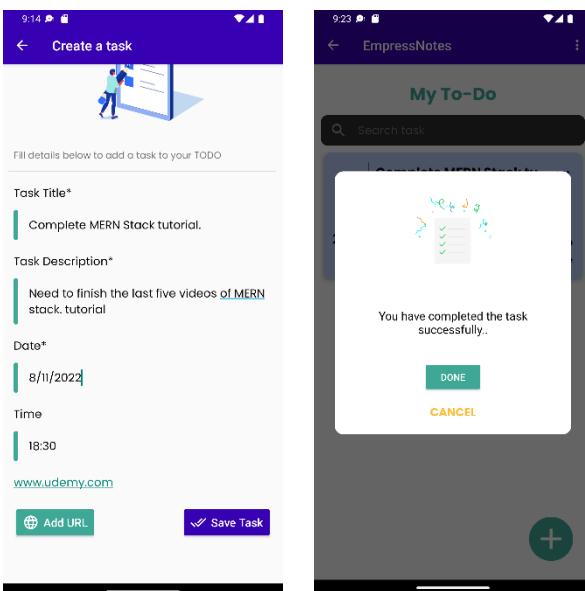


```
// READ ALL TO-DO TASKS
public Cursor readTaskData() {
    String query = "SELECT * FROM " + TABLE_NAME2;
    SQLiteDatabase db = this.getReadableDatabase();

    Cursor cursor = null;
    if (db != null) {
        cursor = db.rawQuery(query, selectionArgs: null);
    }
    return cursor;
}
```

```
void storeTaskDataInArrays() {
    Cursor cursor = myDB.readTaskData();
    if(cursor.getCount() == 0) {
        empty_img.setVisibility(View.VISIBLE);
        empty_txt.setVisibility(View.VISIBLE);
    }else{
        while (cursor.moveToNext()) {
            task_id.add(cursor.getString( i 0));
            task_title.add(cursor.getString( i 1));
            task_description.add(cursor.getString( i 2));
            task_date.add(cursor.getString( i 3));
            task_time.add(cursor.getString( i 4));
            task_url.add(cursor.getString( i 5));
            task_status.add(cursor.getString( i 6));
        }
        empty_img.setVisibility(View.GONE);
        empty_txt.setVisibility(View.GONE);
    }
}
```

UPDATE : User will be able to view and update necessary task details by clicking on the task card and simply updating necessary fields. He can also add new web-link or update the web-link if needed. Whenever user completes a certain task he can update the task status into “COMPLETED” by choosing from more options.



```
// UPDATE A TO-DO TASK
public void updateTaskData(String task_id, String title, String description, String date, String time, String url) {
    SQLiteDatabase db = this.getWritableDatabase();
    ContentValues cv = new ContentValues();

    cv.put(COLUMN_TASK_TITLE, title);
    cv.put(COLUMN_TASK_DESCRIPTION, description);
    cv.put(COLUMN_TASK_DATE, date);
    cv.put(COLUMN_TASK_TIME, time);
    cv.put(COLUMN_TASK_URL, url);

    long result = db.update(TABLE_NAME2, cv, whereClause: "id=?", new String[] {task_id});
    if(result == -1) {
        Toast.makeText(context, text: "Failed to update task", Toast.LENGTH_SHORT).show();
    }else {
        Toast.makeText(context, text: "Task updated successfully", Toast.LENGTH_SHORT).show();
    }
}

// UPDATE A TASK AS COMPLETED
public void taskCompleted(String task_id) {
    SQLiteDatabase db = this.getWritableDatabase();
    ContentValues cv = new ContentValues();

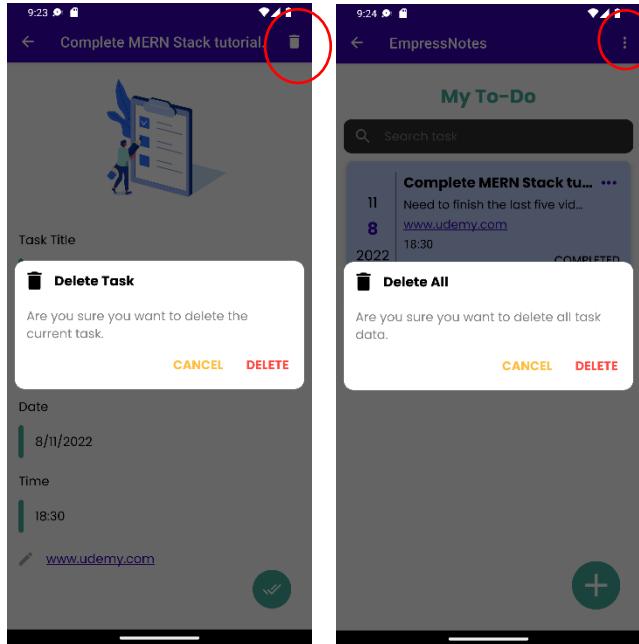
    cv.put(COLUMN_TASK_STATUS, "COMPLETED");

    long result = db.update(TABLE_NAME2, cv, whereClause: "id=?", new String[] {task_id});
    if(result == -1) {
        Toast.makeText(context, text: "Failed to complete task", Toast.LENGTH_SHORT).show();
    }else {
        Toast.makeText(context, text: "Task completion successful", Toast.LENGTH_SHORT).show();
    }
}
```

```
update_btn.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View view) {
        //then only we call update method
        DatabaseHelper myDB = new DatabaseHelper( context: ToDoUpdateTask.this);
        title = title_input.getText().toString().trim();
        description = description_input.getText().toString().trim();
        date = date_input.getText().toString().trim();
        time = time_input.getText().toString().trim();
        url = url_input.getText().toString().trim();
        myDB.updateTaskData(id, title, description, date, time, url);

        Intent intent = new Intent( packageContext: ToDoUpdateTask.this, ToDoMain.class);
        startActivity(intent);
    }
});
```

DELETE: User can easily delete a single task by selecting the bin icon on the view task action bar. If he wants to delete all the tasks in the My To-Do, he can select the “Delete All” option from the To-Do home page. In either instances user needs to confirm delete action from the alert dialog.



```
// DELETE ONE TO-DO TASK
public void deleteTask(String task_id) {
    SQLiteDatabase db = this.getWritableDatabase();
    long result = db.delete(TABLE_NAME2, whereClause: "id=?", new String[] {task_id});
    if(result == -1) {
        Toast.makeText(context, text: "Failed to delete task", Toast.LENGTH_SHORT).show();
    } else {
        Toast.makeText(context, text: "Task deleted successfully", Toast.LENGTH_SHORT).show();
    }
}

// DELETE ALL TO-DO TASKS
public void deleteAllTasks() {
    SQLiteDatabase db = this.getWritableDatabase();
    db.execSQL("DELETE FROM " + TABLE_NAME2);
}

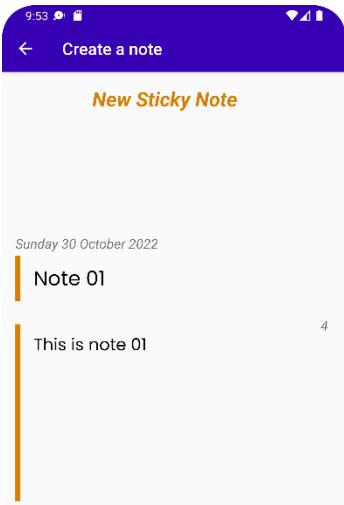
public void confirmDeleteDialog(String id) {
    AlertDialog.Builder builder = new AlertDialog.Builder(context: ToDoUpdateTask.this);
    View view = LayoutInflater.from(this).inflate(
        R.layout.todo_delete_task_layout,
        (ViewGroup) findViewById(R.id.layoutTaskDelete)
    );
    builder.setView(view);
    AlertDialog deleteTaskDialog = builder.create();
    if (deleteTaskDialog.getWindow() != null) {
        deleteTaskDialog.getWindow().setBackgroundDrawable(new ColorDrawable(0));
    }

    // If user confirm delete action
    view.findViewById(R.id.textDeleteTaskBtn).setOnClickListener(new View.OnClickListener() {
        @Override
        public void onClick(View view) {
            DatabaseHelper taskDB = new DatabaseHelper(context: ToDoUpdateTask.this);
            taskDB.deleteTask(id);
            // Return back to to-do home page
            Intent intent = new Intent(packageContext: ToDoUpdateTask.this, ToDoMain.class);
            startActivity(intent);
        }
    });

    // If user cancel delete action
    view.findViewById(R.id.textCancelDeleteTaskBtn).setOnClickListener(new View.OnClickListener() {
        @Override
        public void onClick(View view) {
            deleteTaskDialog.dismiss();
        }
    });
    deleteTaskDialog.show();
}
```

III. IT 21073564 - Priyabandu H.K.S.N

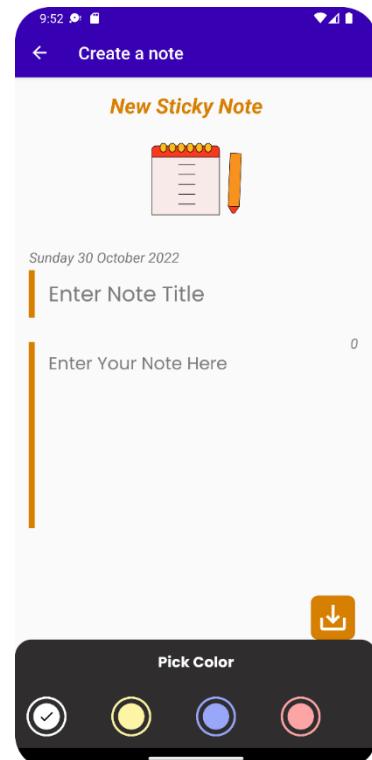
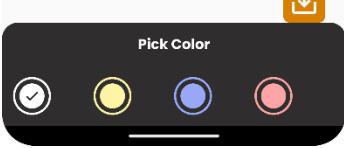
CREATE: User can add sticky notes from this interface. User can direct to this interface through the plus icon of the MY STICKY NOTES home page. Sticky note color can be changed by picking color according to user's choice. There are validations when entering the note as you can't enter a note without giving note title and note body. When entering note body, the word count is automatically calculated each and every time when a new word is inserted. After entering relevant data user can click on save icon and after that title, note, created date and the picked color is saved in database.



```
//Add new sticky Note
public void addNote(String title, String note, String datetime, String color) {
    SQLiteDatabase db = this.getWritableDatabase();
    ContentValues cv = new ContentValues();

    cv.put(COLUMN_NOTES_TITLE, title);
    cv.put(COLUMN_NOTES_BODY, note);
    cv.put(COLUMN_NOTES_DATETIME, datetime);
    cv.put(COLUMN_NOTES_COLOR, color);

    long result = db.insert(TABLE_NAME3, nullColumnHack: null, cv);
    if(result == -1) {
        Toast.makeText(context, text: "Failed to add note", Toast.LENGTH_SHORT).show();
    }else {
        Toast.makeText(context, text: "Note added successfully!", Toast.LENGTH_SHORT).show();
    }
}
```



```
noteAdd_button.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View view) {
        if (noteTitle_input.getText().toString().trim().isEmpty()) {
            Toast.makeText( context: CreateNote.this, text: "Title can not be empty!", Toast.LENGTH_SHORT).show();
            return;
        } else if(note_input.getText().toString().trim().isEmpty()){
            Toast.makeText( context: CreateNote.this, text: "Note can not be empty!", Toast.LENGTH_SHORT).show();
            return;
        } else {
            DatabaseHelper myDB = new DatabaseHelper( context: CreateNote.this);
            myDB.addNote(noteTitle_input.getText().toString().trim(),
                         note_input.getText().toString().trim(),
                         noteDateTime.getText().toString().trim(),
                         selectedNoteColor);

            Intent intent = new Intent( packageContext: CreateNote.this, MyNotes.class);
            startActivity(intent);
        }
    }
});
```

RETRIEVE: This is MY STICKY NOTES home page. All the sticky notes that has been created are displayed as follows with mentioning the created date and chosen color.



```
//Read all sticky notes
public Cursor readAllNotes() {
    String query = "SELECT * FROM " + TABLE_NAME3;
    SQLiteDatabase db = this.getReadableDatabase();

    Cursor cursor = null;
    if(db != null) {
        cursor = db.rawQuery(query, null);
    }
    return cursor;
}
```

```
//call function to display data as recycler view
storeNotesInArrays();

notesAdapter = new NotesAdapter(activity: MyNotes.this, context: MyNotes.this, note_id, note_title,
                                note_body, note_datetime, note_color);
notes_recyclerView.setAdapter(notesAdapter);
notes_recyclerView.setLayoutManager(new LinearLayoutManager(context: MyNotes.this));
}

void storeNotesInArrays() {
    Cursor cursor = myDB.readAllNotes();
    if(cursor.getCount() == 0) {
        empty_imageview.setVisibility(View.VISIBLE);
        no_data.setVisibility(View.VISIBLE);
    }else{
        while (cursor.moveToNext()) {
            note_id.add(cursor.getString(0));
            note_title.add(cursor.getString(1));
            note_body.add(cursor.getString(2));
            note_datetime.add(cursor.getString(3));
            note_color.add(cursor.getString(4));
        }
        empty_imageview.setVisibility(View.GONE);
        no_data.setVisibility(View.GONE);
    }
}
```

UPDATE: By touching particular note from the list in MY STICKY NOTE home page this page with the details of selected note is displayed. In here user can view data which is entered earlier. And also, user can do any modifications to note and existing data will be automatically updated with edited data after touching the edit icon at the bottom of this page.



```
//Update sticky notes
public void editNote(String row_id, String title, String note){
    SQLiteDatabase db = this.getWritableDatabase();
    ContentValues cv = new ContentValues();

    cv.put(COLUMN_NOTES_TITLE, title);
    cv.put(COLUMN_NOTES_BODY, note);

    long result = db.update(TABLE_NAME3, cv, whereClause: "id=?", new String[] {row_id});
    if(result == -1) {
        Toast.makeText(context, text: "Failed to edit note", Toast.LENGTH_SHORT).show();
    }else {
        Toast.makeText(context, text: "Note edited successfully!", Toast.LENGTH_SHORT).show();
    }
}
```

```

void getAndSetIntentData(){
    if(getIntent().hasExtra( name: "id" ) && getIntent().hasExtra( name: "title" ) && getIntent().hasExtra( name: "note")){
        //getting data from intent
        id = getIntent().getStringExtra( name: "id");
        title = getIntent().getStringExtra( name: "title");
        note = getIntent().getStringExtra( name: "note");

        //setting intent data
        noteTitle_input.setText(title);
        note_input.setText(note);
    }else{
        Toast.makeText( context: this, text: "No Data", Toast.LENGTH_LONG).show();
    }
}

```

```

//get earlier inserted data that to be updated through intent
getAndSetIntentData();

ActionBar ab = getSupportActionBar();
if(ab != null){
    ab.setTitle(title);
}

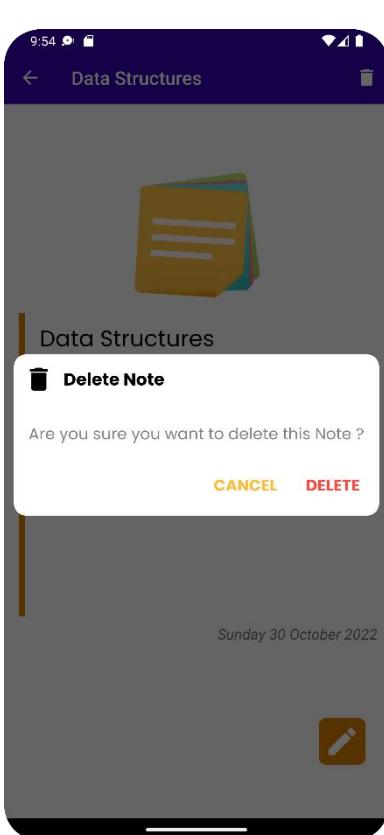
noteEdit_button.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View view) {
        DatabaseHelper myDB = new DatabaseHelper( context: UpdateNote.this);
        title = noteTitle_input.getText().toString().trim();
        note = note_input.getText().toString().trim();

        myDB.editNote(id, title, note);

        Intent intent = new Intent( packageName: UpdateNote.this, MyNotes.class);
        startActivity(intent);
    }
});

```

DELETE: By touching the delete icon on the top of view note page it will display a dialog box like this. By touching the delete option user can remove particular note from the database.



```

//delete particular sticky note
public void deleteNote(String row_id){
    SQLiteOpenHelper db = this.getReadableDatabase();
    long result = db.delete(TABLE_NAME3, whereClause: "id=?", new String[] {row_id});
    if(result == -1) {
        Toast.makeText(context, text: "Failed to delete note", Toast.LENGTH_SHORT).show();
    }else {
        Toast.makeText(context, text: "Note deleted successfully!", Toast.LENGTH_SHORT).show();
    }
}

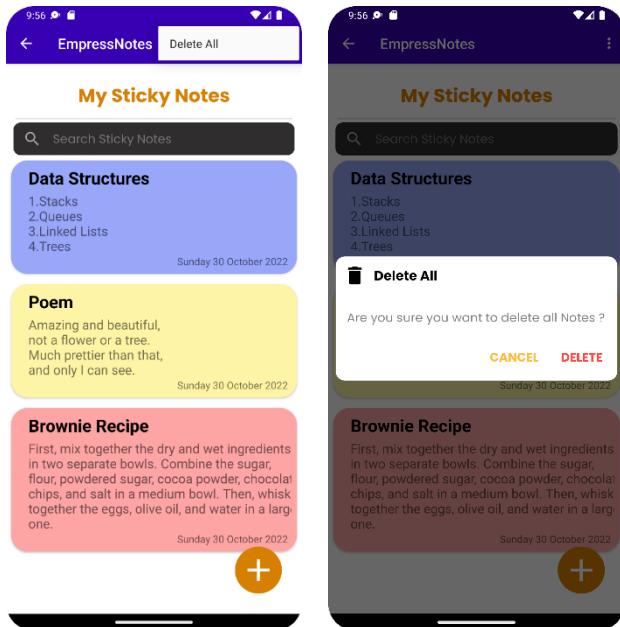
public void deleteAllNotes(){
    SQLiteOpenHelper db = this.getReadableDatabase();
    db.execSQL("DELETE FROM " + TABLE_NAME3);
}

//delete selected sticky note using dialog box
void confirmDialog() {
    AlertDialog.Builder builder = new AlertDialog.Builder( context: UpdateNote.this);
    View view = LayoutInflater.from(this).inflate(
        R.layout.note_delete_note_layout,
        (ViewGroup) findViewById(R.id.layoutNoteDelete)
    );
    builder.setView(view);
    AlertDialog deleteNoteDialog = builder.create();
    if (deleteNoteDialog.getWindow() != null) {
        deleteNoteDialog.getWindow().setBackgroundDrawable(new ColorDrawable(0));
    }

    // If user confirm delete action
    view.findViewById(R.id.textDeleteNoteBtn).setOnClickListener(new View.OnClickListener() {
        @Override
        public void onClick(View view) {
            DatabaseHelper diaryDB = new DatabaseHelper( context: UpdateNote.this);
            diaryDB.deleteNote(id);
            // Refresh and return back to to-do home page
            Intent intent = new Intent( packageName: UpdateNote.this, MyNotes.class);
            startActivity(intent);
        }
    });
    // If user cancel delete action
    view.findViewById(R.id.textCancelDeleteNoteBtn).setOnClickListener(new View.OnClickListener() {
        @Override
        public void onClick(View view) { deleteNoteDialog.dismiss(); }
    });
    deleteNoteDialog.show();
}

```

And also, our app provides facility for users to delete all the notes in the app. It can be done by touching delete all option in the MY STICKY NOTES home page and it will display a pop-up dialog box asking to confirm. By confirming all the record in the database will be deleted.



```
//Delete all notes using dialog box
void confirmDialog() {
    AlertDialog.Builder builder = new AlertDialog.Builder(context: MyNotes.this);
    View view = LayoutInflater.from(this).inflate(
        R.layout.note_delete_all_layout,
        (ViewGroup) findViewById(R.id.layoutNoteDeleteAll)
    );
    builder.setView(view);
    AlertDialog deleteAllDialog = builder.create();
    if (deleteAllDialog.getWindow() != null) {
        deleteAllDialog.getWindow().setBackgroundDrawable(new ColorDrawable(0));
    }
}

// If user confirm delete action
view.findViewById(R.id.textDeleteAllNotesBtn).setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View view) {
        DatabaseHelper diaryDB = new DatabaseHelper(context: MyNotes.this);
        diaryDB.deleteAllNotes();
        // Refresh and return back to to-do home page
        Intent intent = new Intent(packageContext: MyNotes.this, MyNotes.class);
        startActivity(intent);
    }
});
// If user cancel delete action
view.findViewById(R.id.textCancelDeleteAllNotesBtn).setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View view) { deleteAllDialog.dismiss(); }
});
deleteAllDialog.show();
```

```
public void deleteAllNotes(){
    SQLiteDatabase db = this.getWritableDatabase();
    db.execSQL("DELETE FROM " + TABLE_NAME3);
}
```

IV. IT21073496 - Nadeera A.A.D.P

This function allows the user to create a list and add items to each list.

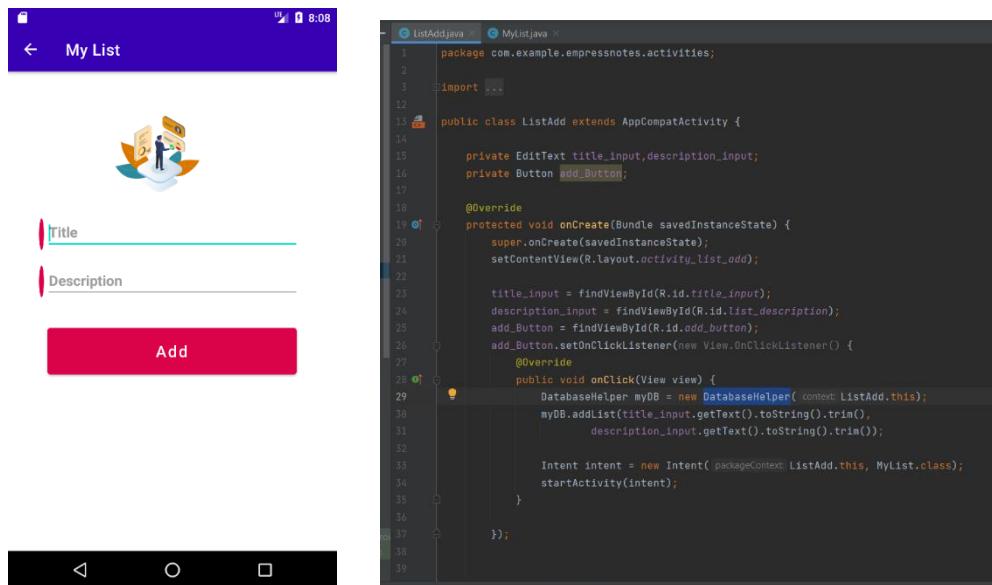
User can navigate to this function by selecting the “My Lists” option in the home page.

Users can take and make shopping list, grocery list, household list, equipment list or add items easily and quickly with the simple note pad.

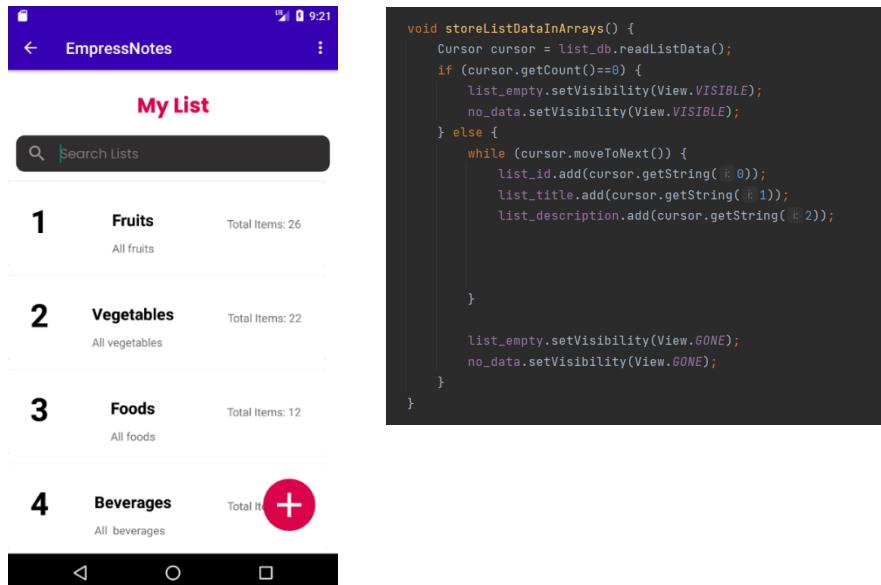
Moreover, users can manage list efficiently and easily. It can be adding new items and can be update or delete details.

Users can view all details which have been done earlier. As well as can be delete all items at once.

CREATE - When the user clicks on the floating button then the user can create list.

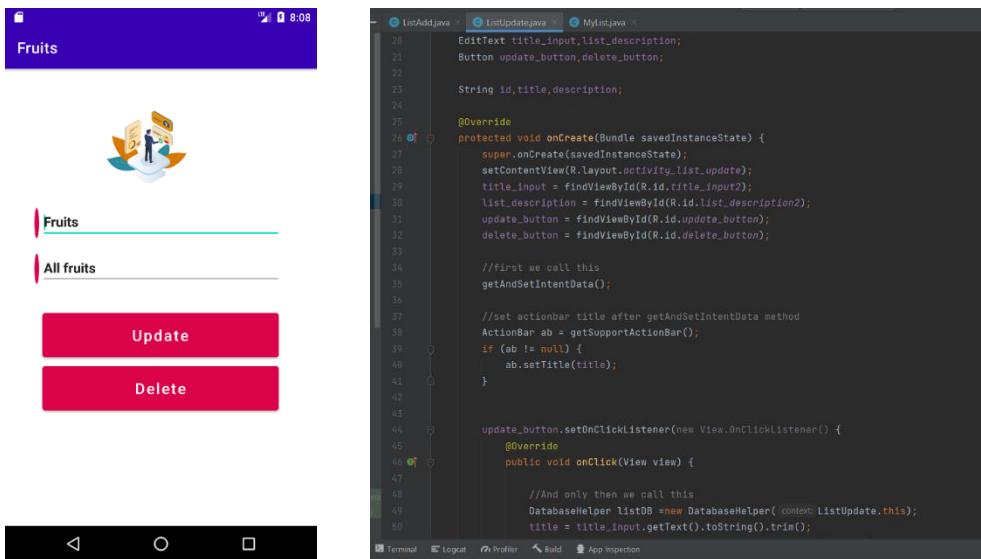


RETRIEVE - When the user clicks on the floating button then the user can create list.

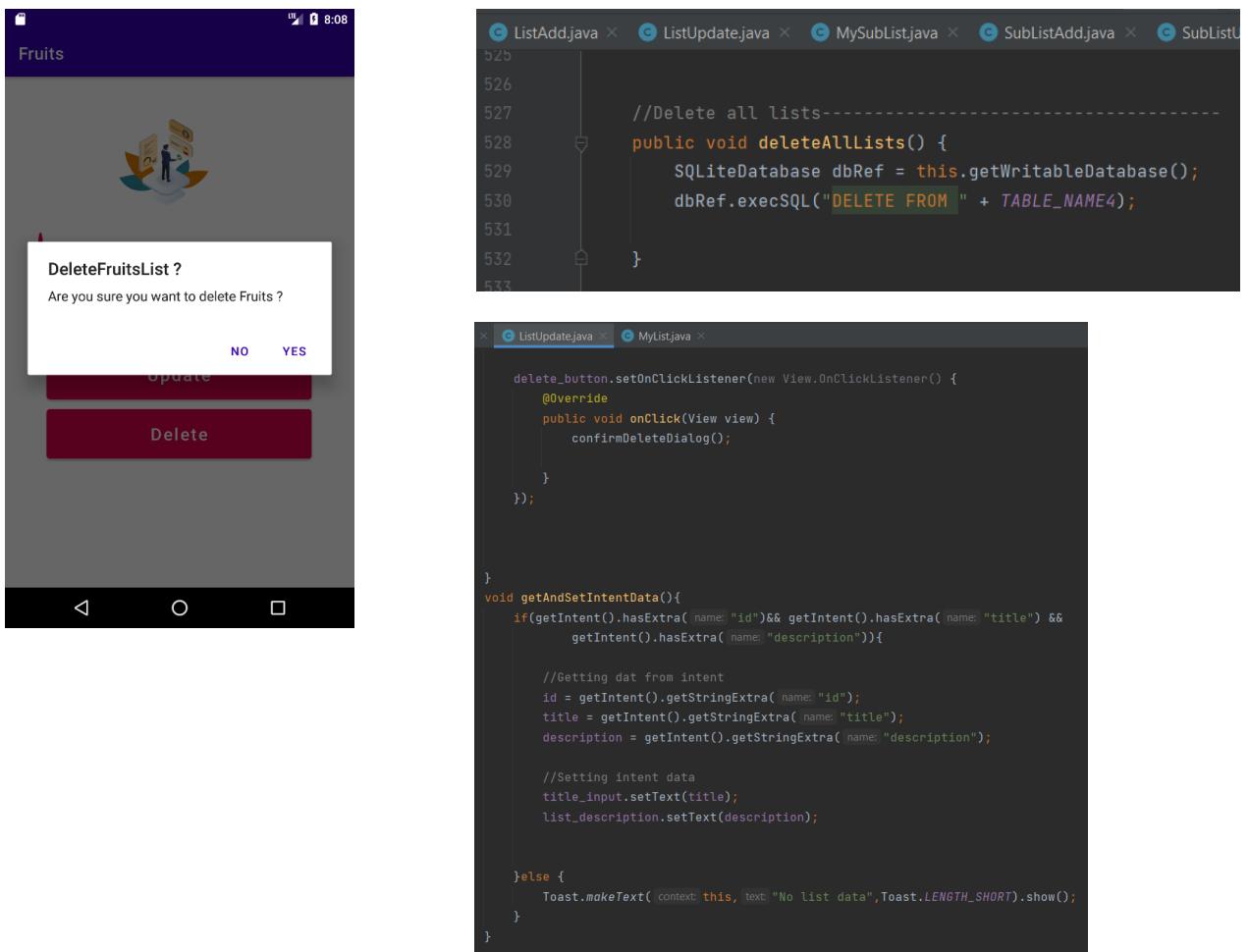


UPDATE / DELETE - all the updates and the deletions are also handled here.

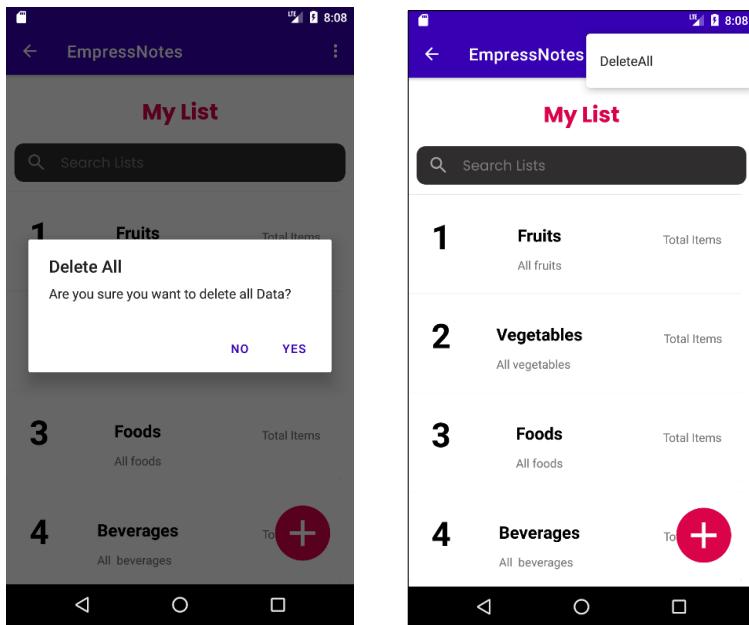
Update



Delete



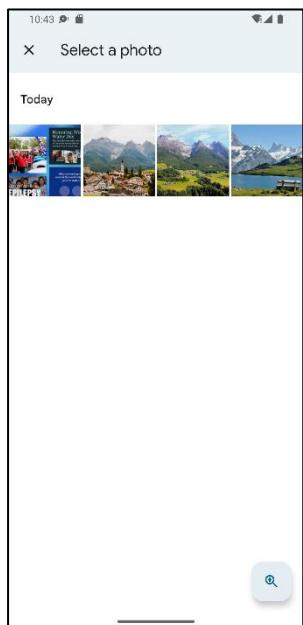
Delete All Lists



4. Out of box features

I. IT21060694 – Weerasooriya K.A

Adding an image



```
public void uploadDiaryImage(){
    buttonImageSave = findViewById(R.id.diaryAddImage);
    buttonImageSave.setOnClickListener(new View.OnClickListener() {
        @Override
        public void onClick(View view) {
            Intent intent = new Intent(Intent.ACTION_PICK, MediaStore.Images.Media.EXTERNAL_CONTENT_URI);
            startActivityForResult(intent, REQUEST_CODE_SELECT_DIARY_IMAGE);
        }
    });

    @Override
    protected void onActivityResult(int requestCode, int resultCode, @Nullable Intent data) {
        super.onActivityResult(requestCode, resultCode, data);
        if (requestCode == REQUEST_CODE_SELECT_DIARY_IMAGE & resultCode == RESULT_OK) {
            if (data != null) {
                Uri selectedImageUri = data.getData();
                if (selectedImageUri != null) {
                    try {
                        InputStream inputStream = getContentResolver().openInputStream(selectedImageUri);
                        Bitmap bitmap = BitmapFactory.decodeStream(inputStream);
                        imageDiary.setImageBitmap(bitmap);
                        imageDiary.setVisibility(View.VISIBLE);

                        selectedDiaryImagePath = getPathFromUri(selectedImageUri);

                    } catch (Exception e) {
                        Toast.makeText(context, e.getMessage(), Toast.LENGTH_SHORT).show();
                    }
                }
            }
        }
    }
}
```

Calculation: Calculating diary body word count



```
//Calculating word count method implementation
public void getWordCount() {

    wordCount = findViewById(R.id.wordCount);
    inputDiaryBody = findViewById(R.id.inputDiaryBody);

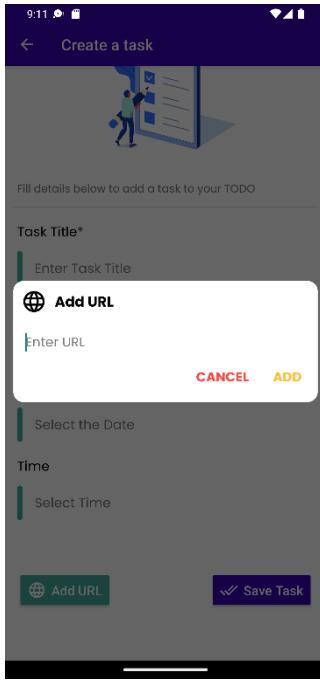
    inputDiaryBody.addTextChangedListener(new TextWatcher() {
        @Override
        public void beforeTextChanged(CharSequence charSequence, int i, int i1, int i2) {
        }

        @Override
        public void onTextChanged(CharSequence charSequence, int i, int i1, int i2) {
            String text = inputDiaryBody.getText().toString();
            text = text.replace("\n", " ");
            String[] textArray = text.split(" ");
            wordCount.setText(" "+textArray.length);
        }

        @Override
        public void afterTextChanged(Editable editable) {
        }
    });
}
```

II. IT21037610 – Wanniarachchi W.D.D.M

Adding a Web-Link

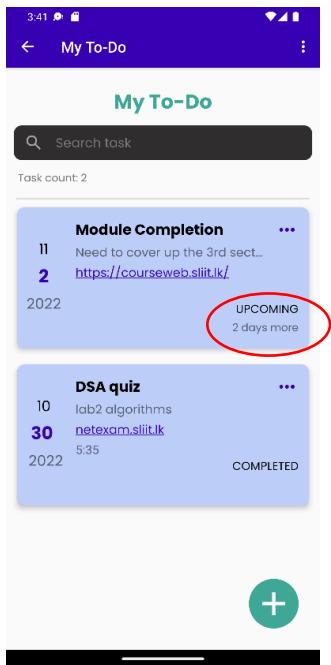


```
private void showAddTaskUrlDialog() {
    if(dialog_add_url == null) {
        AlertDialog.Builder builder = new AlertDialog.Builder(context).setPositiveButton("ADD");
        View view = LayoutInflater.from(this).inflate(R.layout.todo_add_url_layout, (ViewGroup) findViewById(R.id.layoutAddTaskUrl));
        builder.setView(view);
        dialog_add_url = builder.create();
        if(dialog_add_url.getWindow() != null) {
            dialog_add_url.getWindow().setBackgroundDrawable(new ColorDrawable(0));
        }

        final EditText inputUrl = view.findViewById(R.id.editTextTaskUrl);
        inputUrl.requestFocus();

        // Confirm add URL
        view.findViewById(R.id.textAddTaskUrlBtn).setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View v) {
                if(inputUrl.getText().toString().trim().isEmpty()) {
                    // Notify user that URL is empty
                    Toast.makeText(context, "Enter URL", Toast.LENGTH_SHORT).show();
                } else if(!Patterns.WEB_URL.matcher(inputUrl.getText().toString()).matches()) {
                    // Display error if user enter invalid URL
                    Toast.makeText(context, "Please Enter Valid URL", Toast.LENGTH_SHORT).show();
                } else {
                    // Display the submitted valid url in the text view
                    text_url.setText(inputUrl.getText().toString());
                    layout_url.setVisibility(View.VISIBLE);
                    dialog_add_url.dismiss();
                }
            }
        });
    }
}
```

Calculation: Calculating number of days to a task



```
try {
    String dateInput = String.valueOf(task_date.get(position));

    // Split date components
    String[] s = dateInput.split( regex: "/" );
    String day = s[0];
    String month = s[1];
    String year = s[2];

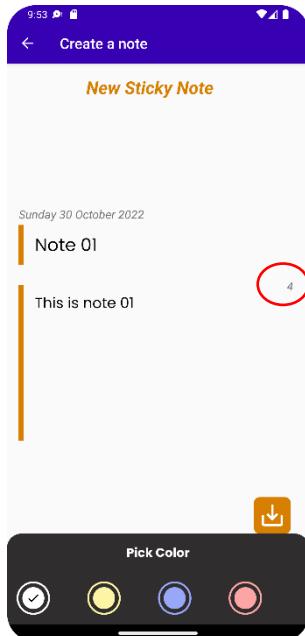
    holder.txt_task_day.setText(day);
    holder.txt_task_month.setText(month);
    holder.txt_task_year.setText(year);

    // CALCULATE NUMBER OF DAYS
    SimpleDateFormat sdf = new SimpleDateFormat( pattern: "dd/MM/yyyy", Locale.getDefault() );
    long diff = sdf.parse(dateInput).getTime() - new Date().getTime();
    long seconds = diff / 1000;
    long minutes = seconds / 60;
    long hours = minutes / 60;
    int days = ((int) (long) hours / 24);
    holder.txt_days_calc.setText(String.valueOf(days)+" days more");

} catch (Exception e) {
    e.printStackTrace();
}
```

III. IT21073564 - Priyabandu H.K.S.N

Calculation: Calculating note word count



```
public void getWordCount() {

    wordCount = findViewById(R.id.lbl_word_count);
    note_input = findViewById(R.id.inputNote);

    note_input.addTextChangedListener(new TextWatcher() {
        @Override
        public void beforeTextChanged(CharSequence charSequence, int i, int i1, int i2) {

        }

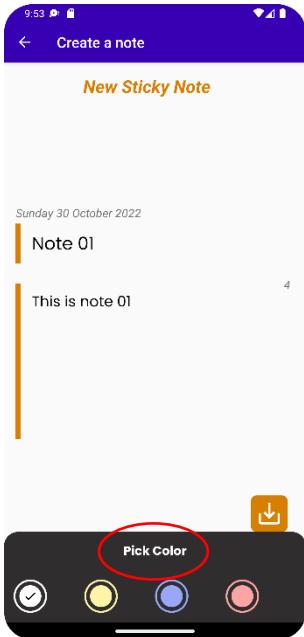
        @Override
        public void onTextChanged(CharSequence charSequence, int i, int i1, int i2) {
            String text = note_input.getText().toString();
            text = text.replace( target: "\n", replacement: " " );
            String[] textArray = text.split( regex: " " );
            wordCount.setText(" "+textArray.length);
        }

        @Override
        public void afterTextChanged(Editable editable) {

        }
    });
}
```

Adding sticky note colour :

Sticky note color can be changed by picking color according to user's choice.



```
private void initMiscellaneous(){
    final LinearLayout layoutMiscellaneous = findViewById(R.id.layoutMiscellaneous);
    final BottomSheetBehavior<LinearLayout> bottomSheetBehavior = BottomSheetBehavior.from(layoutMiscellaneous);
    layoutMiscellaneous.findViewById(R.id.textMiscellaneous).setOnClickListener(new View.OnClickListener() {
        @Override
        public void onClick(View view) {
            if(bottomSheetBehavior.getState() != BottomSheetBehavior.STATE_EXPANDED){
                bottomSheetBehavior.setState(BottomSheetBehavior.STATE_EXPANDED);
            }else{
                bottomSheetBehavior.setState(BottomSheetBehavior.STATE_COLLAPSED);
            }
        }
    });

    final ImageView imageViewColor1 = layoutMiscellaneous.findViewById(R.id.imageViewColor1);
    final ImageView imageViewColor2 = layoutMiscellaneous.findViewById(R.id.imageViewColor2);
    final ImageView imageViewColor3 = layoutMiscellaneous.findViewById(R.id.imageViewColor3);
    final ImageView imageViewColor4 = layoutMiscellaneous.findViewById(R.id.imageViewColor4);

    layoutMiscellaneous.findViewById(R.id.viewColor1).setOnClickListener(new View.OnClickListener() {
        @Override
        public void onClick(View view) {
            selectedNoteColor = "#FFFFFF";
            imageViewColor1.setImageResource(R.drawable.ic_done);
            imageViewColor2.setImageResource(0);
            imageViewColor3.setImageResource(0);
            imageViewColor4.setImageResource(0);
        }
    });
}
```

```
private void initMiscellaneous(){
    final LinearLayout layoutMiscellaneous = findViewById(R.id.layoutMiscellaneous);
    final BottomSheetBehavior<LinearLayout> bottomSheetBehavior = BottomSheetBehavior.from(layoutMiscellaneous);
    layoutMiscellaneous.findViewById(R.id.textMiscellaneous).setOnClickListener(new View.OnClickListener() {
        @Override
        public void onClick(View view) {
            if(bottomSheetBehavior.getState() != BottomSheetBehavior.STATE_EXPANDED){
                bottomSheetBehavior.setState(BottomSheetBehavior.STATE_EXPANDED);
            }else{
                bottomSheetBehavior.setState(BottomSheetBehavior.STATE_COLLAPSED);
            }
        }
    });

    final ImageView imageViewColor1 = layoutMiscellaneous.findViewById(R.id.imageViewColor1);
    final ImageView imageViewColor2 = layoutMiscellaneous.findViewById(R.id.imageViewColor2);
    final ImageView imageViewColor3 = layoutMiscellaneous.findViewById(R.id.imageViewColor3);
    final ImageView imageViewColor4 = layoutMiscellaneous.findViewById(R.id.imageViewColor4);

    layoutMiscellaneous.findViewById(R.id.viewColor1).setOnClickListener(new View.OnClickListener() {
        @Override
        public void onClick(View view) {
            selectedNoteColor = "#FFFFFF";
            imageViewColor1.setImageResource(R.drawable.ic_done);
            imageViewColor2.setImageResource(0);
            imageViewColor3.setImageResource(0);
            imageViewColor4.setImageResource(0);
        }
    });
}
```

IV. IT21073496 - Nadeera A.A.D.P

Calculation: Sum of all quantity of sub-items , will be shown in main category.

4	Fruits	Total Items: 15
	All fruits	
5	Vegetables	Total Items: 22
	All vegetables	

4	Apple	10
	Orange	5

```
//get total item count
public Cursor getTotalItems(String id) {

    String total_quantity = "SELECT SUM(" + COLUMN_my_sub_list_Quantity + ") FROM " + TABLE_NAME5+" WHERE m_id= "+id ;
    SQLiteDatabase dbRef = this.getReadableDatabase();
    Cursor cursor = dbRef.rawQuery(total_quantity, selectionArgs: null);

    return cursor;
}
```

5. Test Cases

I. IT21060694 – Weerasooriya K.A

Test No	Test Case	Actual Result	Expected Result	PASS / FAIL
1	In very first access	Display empty diary Image View.	Display empty diary Image View.	PASS
2	After click add icon	Display <u>create</u> a diary activity.	Display <u>create</u> a diary activity.	PASS
3	If the user selects image icon	Open the device storage	Open the device storage	PASS
4	When the user selects an image	Display it in the create diary view.	Display it in the create diary view.	
5	When user save new diary	Display successful toast message and inserted to database	Display successful toast message and diary is displayed in My Diary page.	PASS
6	If data is not inserted to database	Display unsuccessful toast message	Display unsuccessful toast message	PASS
7	When user select a particular Diary in My Diary home page	The diary is displayed on update diary page	The diary is displayed on update diary page	PASS
8	When user edit selected diary and click save button	Replace old diary and display successful message	Successfully updated and display old diary replaced with updated Diary.	PASS
9	If failed to update diary with edited details	Display unsuccessful toast message	Display unsuccessful toast message	PASS
9	When user click delete icon on menu bar in update diary page	Display dialog box asking are u sure to delete diary	Display dialog box asking are u sure to delete diary	PASS
10	When user click cancel option in delete dialog box	Display update diary page without any change	Display update diary page without any change	PASS
11	When user click delete option in delete dialog box	Display diary deleted successful toast message and delete the diary from data base	Display diary deleted successful toast message and diary is not displayed in My Diary page.	PASS
12	When user click the delete all option on top of My Diary home page	Display dialog box asking are u sure to delete all diaries	Display dialog box asking are u sure to delete all diaries	PASS
13	When user click cancel option in delete all dialog box	Display My Diary page without any change	Display My Diary page without any change	PASS
14	When user click delete option in delete all dialog box	Delete all diaries from data base and display deleted successful toast message	Display empty diary page and display deleted successful toast message	PASS

II. IT21037610 – Wanniarachchi W.D.D.M

Test No	Test Case	Actual Result	Expected Result	PASS / FAIL
1	In very first access	Display empty task Image View.	Display empty task Image View.	PASS
2	After click add icon	Display todo_create task activity.	Display todo_create task activity.	PASS
3	If User select date	Display the calendar with the current date.	Display the calendar with the current date.	PASS
4	If the user selects add URL button	Display URL input dialog box	Display URL input dialog box	PASS
5	When user save new task	Display successful toast message and inserted to database	Display successful toast message and task is displayed in To-do task homepage	PASS
6	If data is not inserted to database	Display unsuccessful toast message	Display unsuccessful toast message	PASS
7	When user select a particular To-do in sticky To-do task homepage	The task is displayed on update to-do page	The task is displayed on update to-do page	PASS
8	When user edit selected task and click save task	Replace old task and display successful message	Successfully updated and display old task replaced with updated To-do task.	PASS
9	If failed to update task with edited details	Display unsuccessful toast message	Display unsuccessful toast message	PASS
9	When user click delete icon on menu bar in update To-do page	Display dialog box asking are u sure to delete To-do task	Display dialog box asking are u sure to delete To-do task	PASS
10	When user click cancel option in delete dialog box	Display update to-do page without any change	Display update to-do page without any change	PASS
11	When user click delete option in delete dialog box	Display task deleted successful toast message and delete the task from data base	Display task deleted successful toast message and task is not displayed in My To-do page.	PASS
12	When user click the delete all option on top of My To-do home page	Display dialog box asking are u sure to delete all tasks	Display dialog box asking are u sure to delete all tasks	PASS
13	When user click cancel option in delete all dialog box	Display My To-do page without any change	Display My To-do page without any change	PASS
14	When user click delete option in delete all dialog box	Delete all to-do tasks from data base and display deleted successful toast message	Display empty to-do page and display deleted successful toast message	PASS

III. IT21073564 - Priyabandu H.K.S.N

Test No	Test Case	Actual Result	Expected Result	PASS / FAIL
1	In very first access	Display empty data toast message	Display empty data toast message	PASS
2	After click plus icon	Display Sticky note typing activity	Display Sticky note typing activity	PASS
3	If User select a color to sticky note	Display Sticky note in selected color	Display Sticky note in selected color	PASS
4	When user entering note description	Display the word count when entering each and every word	Display the word count when entering each and every word	PASS
5	When user save new sticky note	Display successful toast message and inserted to database	Display successful message and note is displayed in sticky note home page	PASS
6	If data is not inserted to database	Display unsuccessful toast message	Display unsuccessful toast message	PASS
7	When user select a particular note in sticky note home page	The note is displayed on view page	The note is displayed on view page	PASS
8	When user edit note and click edit icon	Replace old note and display successful message	Successfully updated and display old note replaced with updated note.	PASS
9	If failed to update note with edited details	Display unsuccessful toast message	Display unsuccessful toast message	PASS
9	When user click delete icon on view note page	Display dialog box asking are u sure to delete note	Display dialog box asking are u sure to delete note	PASS
10	When user click cancel option in delete dialog box	Display view note page without any change	Display view note page without any change	PASS
11	When user click delete option in delete dialog box	Display note deleted successful toast message and delete the note from data base	Display note deleted successful toast message and note is not displayed in my sticky note home page	PASS
12	When user click the delete all option on top of my sticky note home page	Display dialog box asking are u sure to delete all notes	Display dialog box asking are u sure to delete all notes	PASS
13	When user click cancel option in delete all dialog box	Display my sticky notes home page without any change	Display my sticky notes home page without any change	PASS
14	When user click delete option in delete all dialog box	Delete all sticky notes from data base and display deleted successful toast message	Display empty home page and display deleted successful toast message	PASS

IV. IT21073496 - Nadeera A.A.D.P

Test No	Test Case	Actual Result	Expected Result	Pass/Fail
No.1	Click + Button	Redirect to list add page	Redirect to list add page	Pass
No.2	Click + Item Add Button	Redirect to list view page	Redirect to list view page	Pass
No.3	Click + Button	Redirect to item add page	Redirect to item add page	Pass
No.4	Click + Back Button	Redirect to Item View page	Redirect to Item View page	Pass
No.5	Click + item	Redirect to Item update/delete page	Redirect to Item update/delete page	Pass
No.6	Click + delete button	Display Dialog box	Display Dialog box	Pass
No.7	Click + update button	Redirect item View page	Redirect item View page	Pass

