Create Performance Task Template 2018

2a.

Write your 2a response here (<150 words):

Hello, welcome to my app showcase. My app is a task app that will help people become more organized in their lives. I will display the features of my app right now. Here is the way someone will add a task to their list. The app will then send them to the editing/info page. This info page will allow them to change different things about the task they have just entered. They will be able to change the task name, change the due date, and change the notes on the task. They will then be able to either delete the task or just close the info page to go back to the main screen. Everything entered by the user will be saved so when the user goes back into the app their tasks will still be present.

Video: https://www.youtube.com/watch?v=b8_ZFp5DZA4&feature=youtu.be

2b.

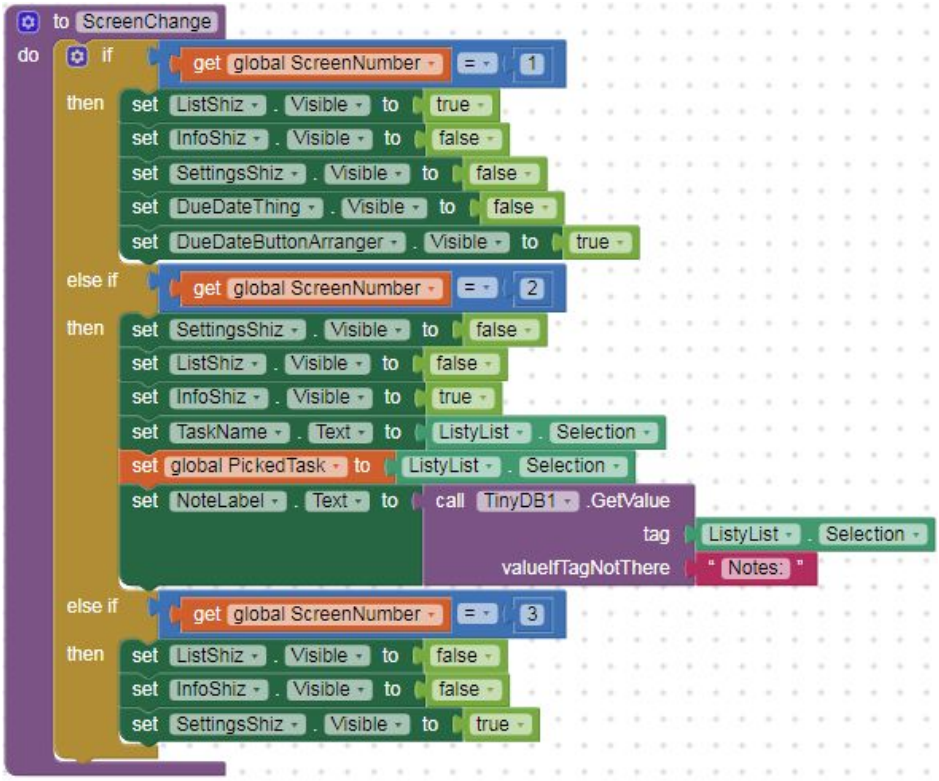
Write your 2b response here (< 200 words):

I started my app off by creating my main UI where the user will be able to add tasks. After getting the UI done I moved to the blocks section and started to add my code to create a popup for the user to enter their tasks and view all the tasks that they have entered. I had difficulty with adding tasks because my list view would not show everything I added but I just forget to switch on/off my variables to the right value. Next, I moved to the Info page that would allow me to change the name of the task, the notes associated with the app and the due date. The hardest part about adding these features was saving the notes to the correct task and keeping them saved in a TinyDB. I ran into some problems with changing pages and sending variables so I developed a procedure that dynamically sets the visibility of vertical arrangements. In each of these vertical arrangements was a different tab of UI. This allowed me to keep all my code on one screen and not have to switch back and forth across different screens.

2c.

Write your 2c response here (< 200 words)

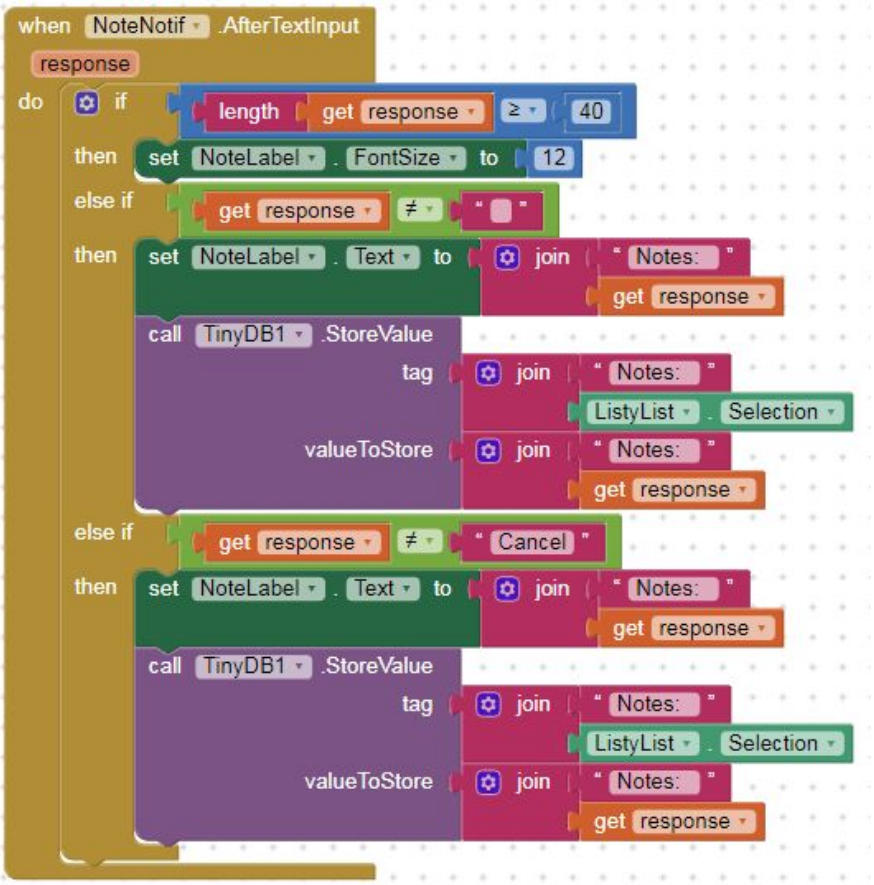
The procedure below is the procedure I created to dynamically switch what vertical arrangement was visible. The first algorithm checks if the variable screen number is equal to 1. If it is then it will set the first vertical arrangement to visible and set every other arrangement set to false. Next, my procedure will check if ScreenNumber is equal to 2. If it is it will make vertical arrangement 2 visible and make any other arrangements invisible. The same algorithm is repeated for vertical arrangement 3. All together this procedure allows me to dynamically change the visible arrangement without massive amounts of code needed every time I want to stitch visible UI. This also allows me to keep all my UI and code on one screen and not have to worry about transferring variables across screens.

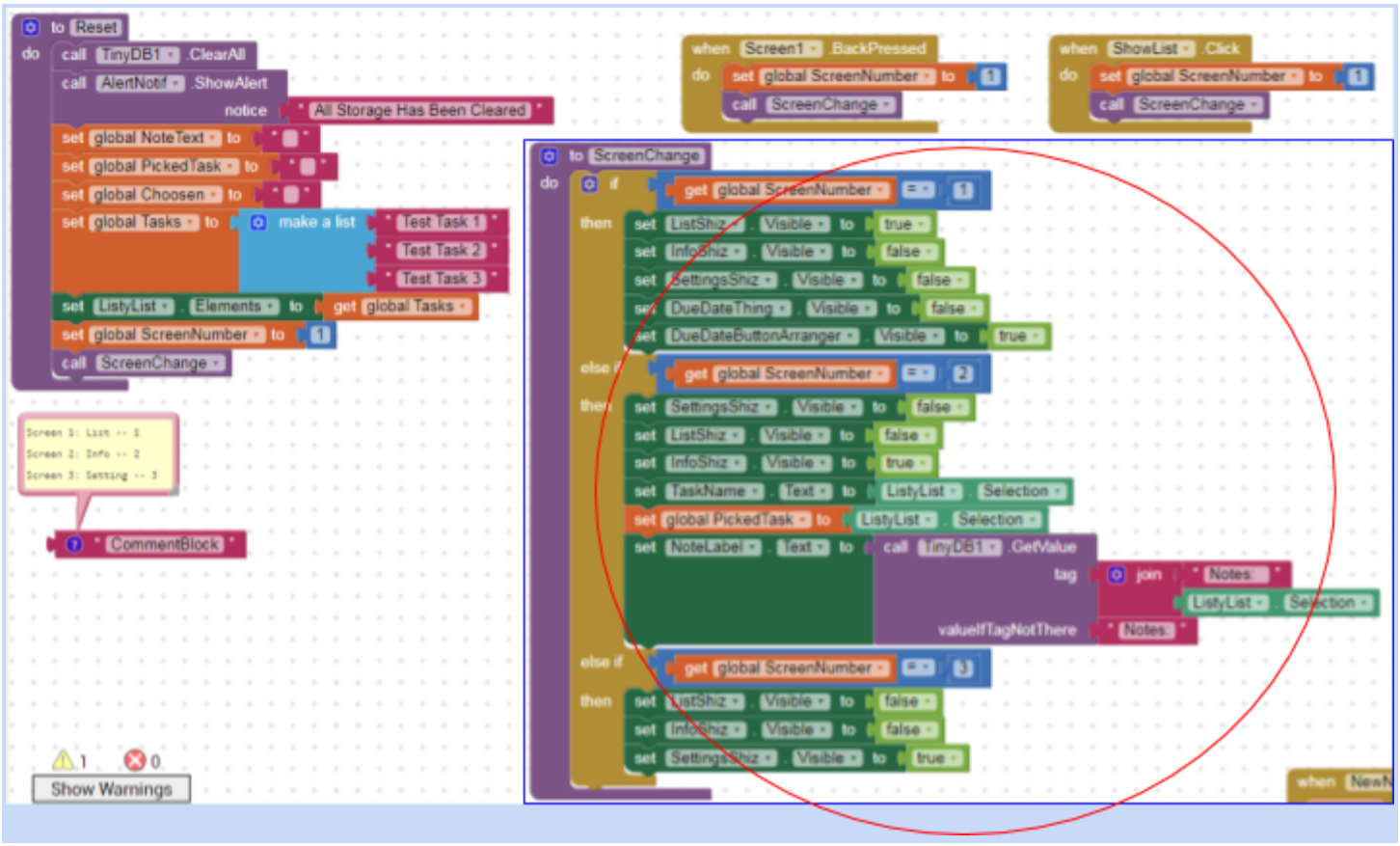


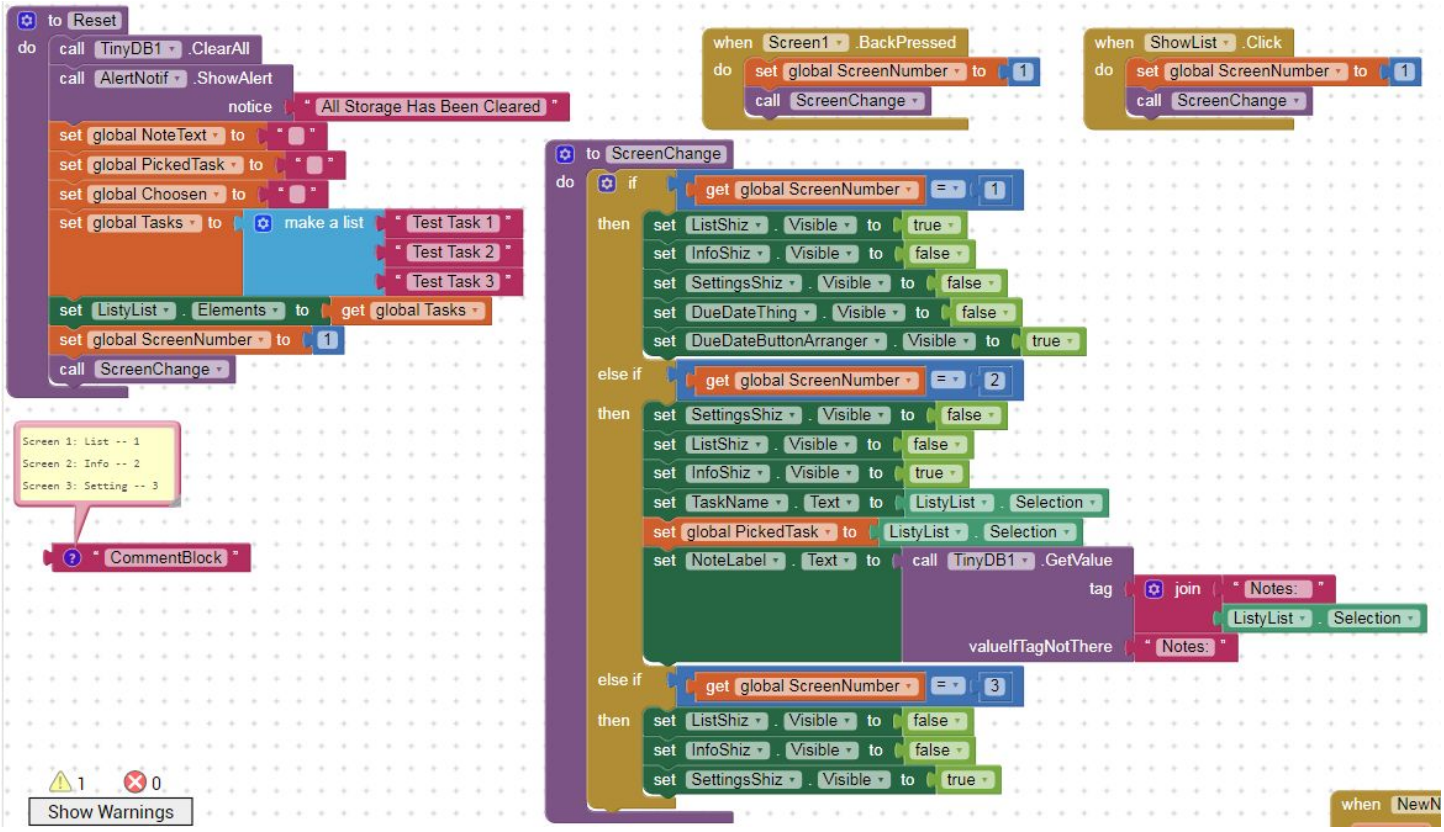
2d.

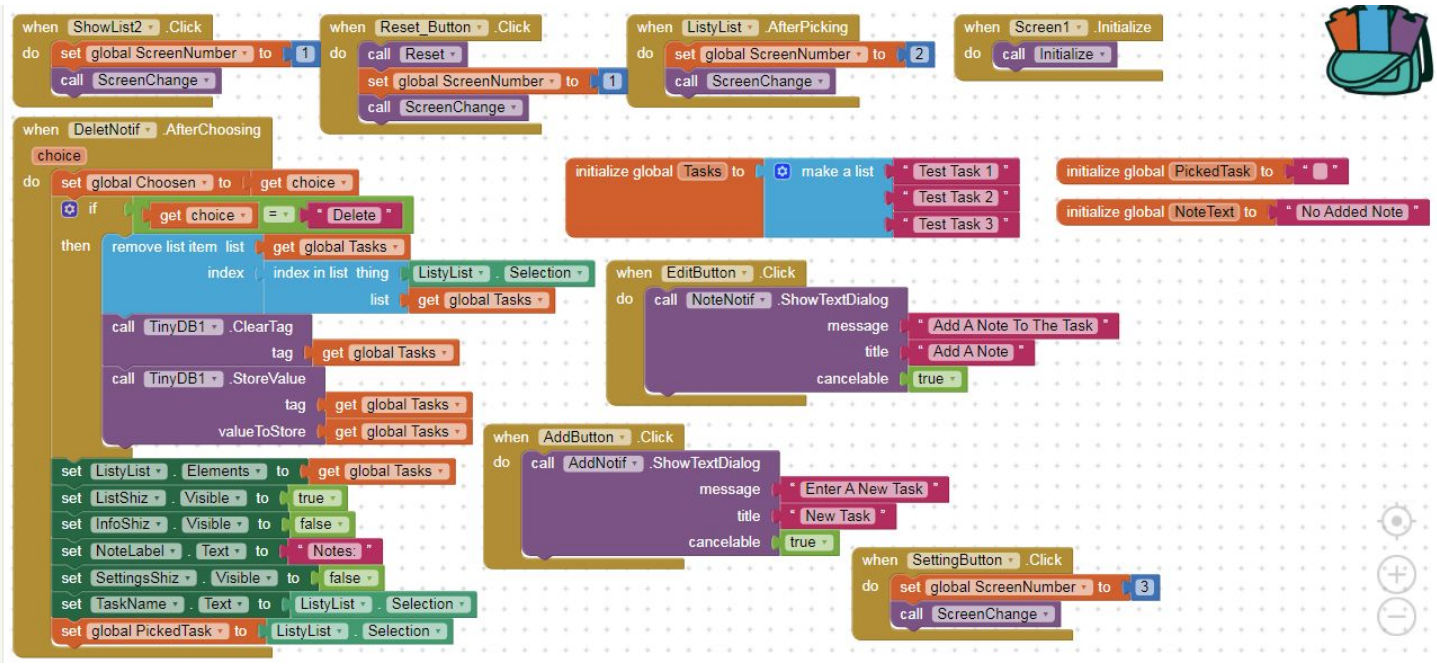
Write your 2d response here (< 200 words)

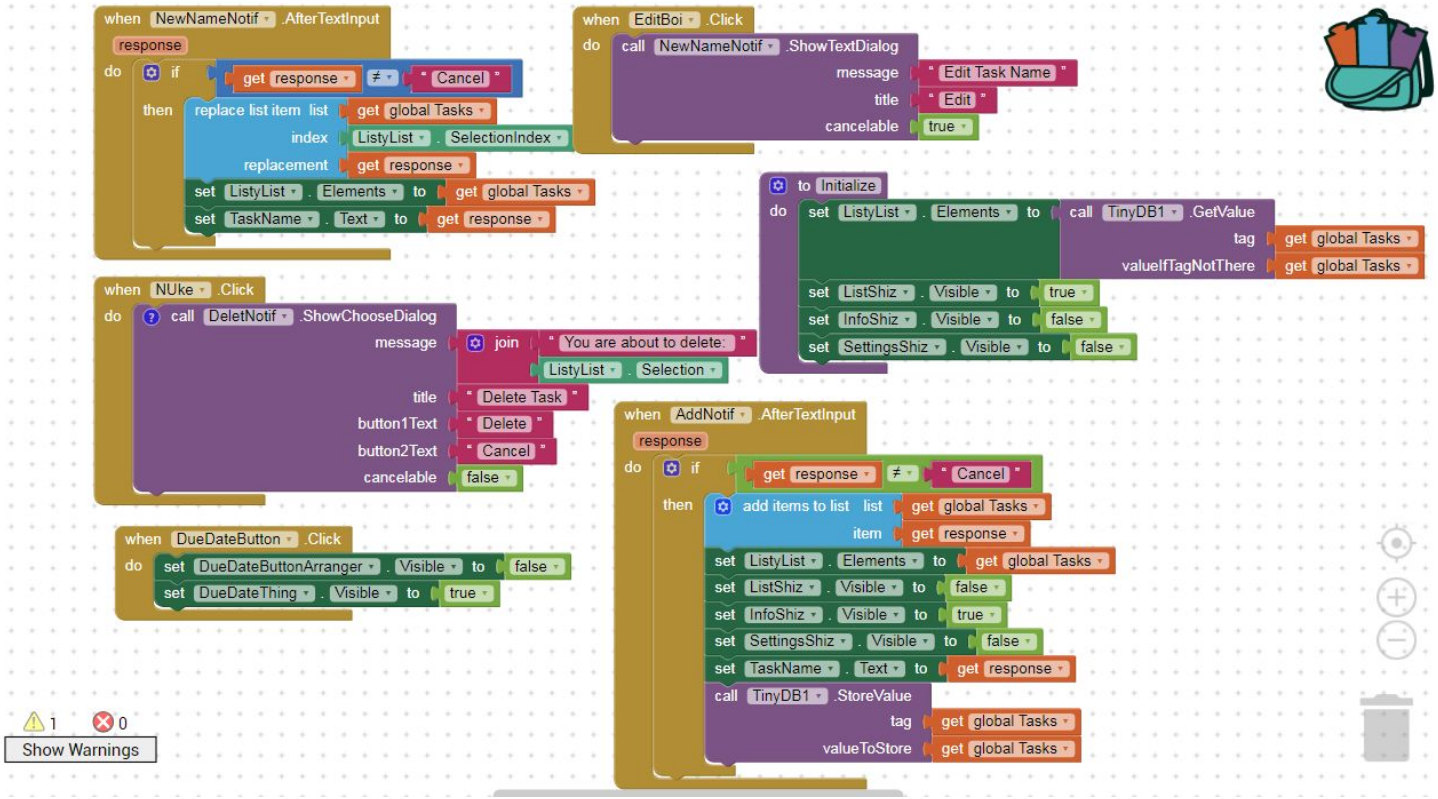
The code I have below manages the complexity of my program by logically choosing the size of text and easily storing variables in a storage system for the user. This system allows me to put multiple lines of code that would normally be spread out across the program into one section of code that is all in one place. The first else-if statement could have been put into other parts of the code but if the code was put into other places it would make the program cluttered and the section of code would have to be repeated many times. Since I put the code at this point in the program it runs after all the other places the code in this algorithm would run if they were separated and repeated many times. This allows me to put condense my code and make it less complex but with the same amount of functionality with less room for error.

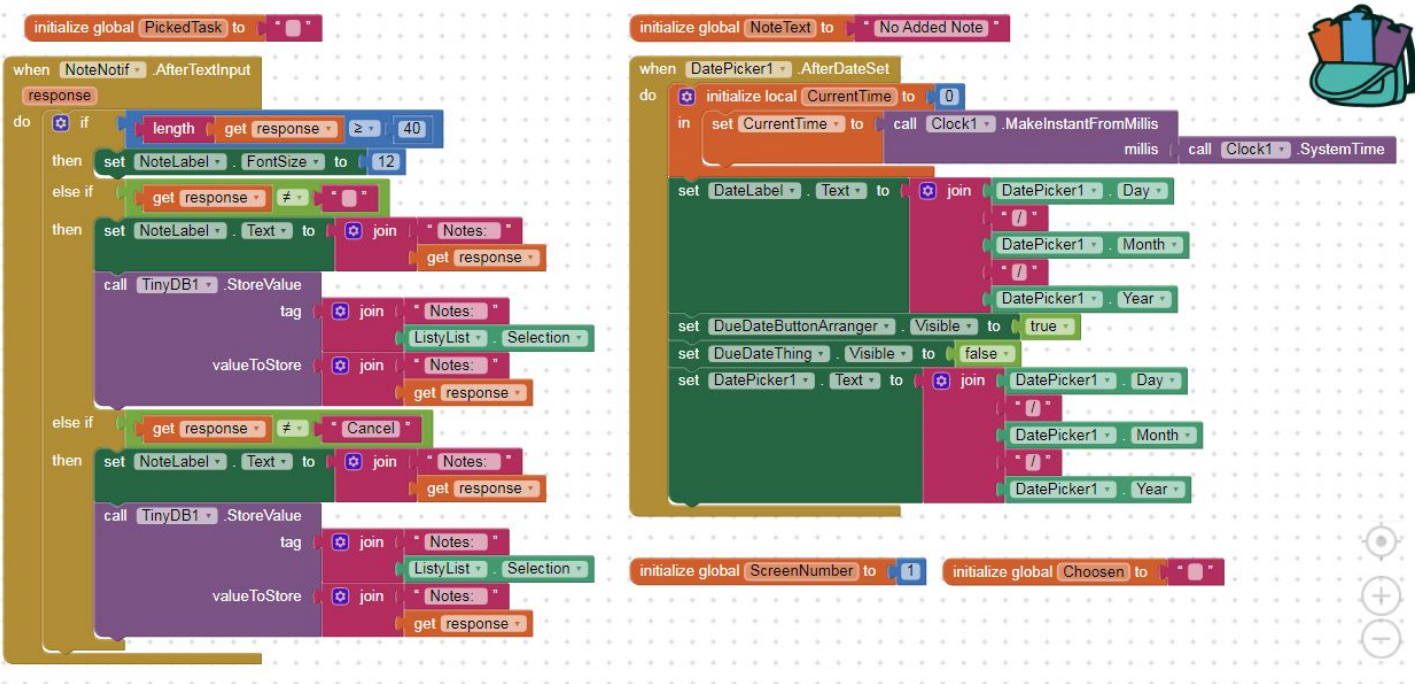

when NoteNotif .AfterTextInput
response
do if length get response ≥ 40
then set NoteLabel . FontSize to 12
else if get response ≠ " "
then set NoteLabel . Text to join " Notes: " get response
call TinyDB1 .StoreValue
tag join " Notes: " ListyList . Selection
valueToStore join " Notes: " get response
else if get response ≠ " Cancel "
then set NoteLabel . Text to join " Notes: " get response
call TinyDB1 .StoreValue
tag join " Notes: " ListyList . Selection
valueToStore join " Notes: " get response


to Reset
do
call TinyDB1 .ClearAll
call AlertNotif .ShowAlert
notice All Storage Has Been Cleared
set global NoteText to
set global PickedTask to
set global Choosen to
set global Tasks to make a list Test Task 1 Test Task 2 Test Task 3
set ListyList . Elements to get global Tasks
set global ScreenNumber to 1
call ScreenChange
Screen 1: List -- 1
Screen 2: Info -- 2
Screen 3: Setting -- 3
CommentBlock
1
0
Show Warnings
when Screen1 .BackPressed
do set global ScreenNumber to 1
call ScreenChange
when ShowList .Click
do set global ScreenNumber to 1
call ScreenChange
to ScreenChange
do if get global ScreenNumber = 1
then set ListShiz . Visible to true
set InfoShiz . Visible to false
set SettingsShiz . Visible to false
set DueDateThing . Visible to false
set DueDateButtonArranger . Visible to true
else if get global ScreenNumber = 2
then set SettingsShiz . Visible to false
set ListShiz . Visible to false
set InfoShiz . Visible to true
set TaskName . Text to ListyList . Selection
set global PickedTask to ListyList . Selection
set NoteLabel . Text to call TinyDB1 .GetValue
tag join Notes: ListyList . Selection
valueIfTagNotThere Notes:
else if get global ScreenNumber = 3
then set ListShiz . Visible to false
set InfoShiz . Visible to false
set SettingsShiz . Visible to true


to Reset
do
call TinyDB1 .ClearAll
call AlertNotif .ShowAlert
notice All Storage Has Been Cleared
set global NoteText to
set global PickedTask to
set global Choosen to
set global Tasks to make a list Test Task 1 Test Task 2 Test Task 3
set ListyList . Elements to get global Tasks
set global ScreenNumber to 1
call ScreenChange
Screen 1: List -- 1
Screen 2: Info -- 2
Screen 3: Setting -- 3
CommentBlock
1
0
Show Warnings
when Screen1 .BackPressed
do set global ScreenNumber to 1
call ScreenChange
when ShowList .Click
do set global ScreenNumber to 1
call ScreenChange
to ScreenChange
do if get global ScreenNumber = 1
then set ListShiz . Visible to true
set InfoShiz . Visible to false
set SettingsShiz . Visible to false
set DueDateThing . Visible to false
set DueDateButtonArranger . Visible to true
else if get global ScreenNumber = 2
then set SettingsShiz . Visible to false
set ListShiz . Visible to false
set InfoShiz . Visible to true
set TaskName . Text to ListyList . Selection
set global PickedTask to ListyList . Selection
set NoteLabel . Text to call TinyDB1 .GetValue
tag join Notes: ListyList . Selection
valueIfTagNotThere Notes:
else if get global ScreenNumber = 3
then set ListShiz . Visible to false
set InfoShiz . Visible to false
set SettingsShiz . Visible to true


when ShowList2 .Click
do set global ScreenNumber to 1
call ScreenChange
when Reset_Button .Click
do call Reset
set global ScreenNumber to 1
call ScreenChange
when ListyList .AfterPicking
do set global ScreenNumber to 2
call ScreenChange
when Screen1 .Initialize
do call Initialize
when DeletNotif .AfterChoosing
choice
do set global Choosen to get choice
if get choice = " Delete "
then remove list item list get global Tasks
index index in list thing ListyList . Selection
list get global Tasks
call TinyDB1 .ClearTag
tag get global Tasks
call TinyDB1 .StoreValue
tag get global Tasks
valueToStore get global Tasks
set ListyList . Elements to get global Tasks
set ListShiz . Visible to true
set InfoShiz . Visible to false
set NoteLabel . Text to " Notes: "
set SettingsShiz . Visible to false
set TaskName . Text to ListyList . Selection
set global PickedTask to ListyList . Selection
initialize global Tasks to make a list " Test Task 1 " " Test Task 2 " " Test Task 3 "
initialize global PickedTask to " "
initialize global NoteText to " No Added Note "
when EditButton .Click
do call NoteNotif .ShowTextDialog
message " Add A Note To The Task "
title " Add A Note "
cancelable true
when AddButton .Click
do call AddNotif .ShowTextDialog
message " Enter A New Task "
title " New Task "
cancelable true
when SettingButton .Click
do set global ScreenNumber to 3
call ScreenChange


when NewNameNotif .AfterTextInput
response
do if get response ≠ " Cancel "
then replace list item list get global Tasks
index ListyList . SelectionIndex
replacement get response
set ListyList . Elements to get global Tasks
set TaskName . Text to get response
when EditBoi .Click
do call NewNameNotif .ShowTextDialog
message " Edit Task Name "
title " Edit "
cancelable true
to Initialize
do set ListyList . Elements to call TinyDB1 .GetValue
tag get global Tasks
valueIfTagNotThere get global Tasks
set ListShiz . Visible to true
set InfoShiz . Visible to false
set SettingsShiz . Visible to false
when NUke .Click
do call DeletNotif .ShowChooseDialog
message join " You are about to delete: "
ListyList . Selection
title " Delete Task "
button1Text " Delete "
button2Text " Cancel "
cancelable false
when DueDateButton .Click
do set DueDateButtonArranger . Visible to false
set DueDateThing . Visible to true
when AddNotif .AfterTextInput
response
do if get response ≠ " Cancel "
then add items to list list get global Tasks
item get response
set ListyList . Elements to get global Tasks
set ListShiz . Visible to false
set InfoShiz . Visible to true
set SettingsShiz . Visible to false
set TaskName . Text to get response
call TinyDB1 .StoreValue
tag get global Tasks
valueToStore get global Tasks
1
0
Show Warnings


initialize global PickedTask to " "
when NoteNotif .AfterTextInput
response
do if length get response ≥ 40
then set NoteLabel . FontSize to 12
else if get response ≠ " "
then set NoteLabel . Text to join " Notes: "
get response
call TinyDB1 .StoreValue
tag join " Notes: "
ListyList . Selection
valueToStore join " Notes: "
get response
else if get response ≠ " Cancel "
then set NoteLabel . Text to join " Notes: "
get response
call TinyDB1 .StoreValue
tag join " Notes: "
ListyList . Selection
valueToStore join " Notes: "
get response
initialize global NoteText to " No Added Note "
when DatePicker1 .AfterDateSet
do initialize local CurrentTime to 0
in set CurrentTime to call Clock1 .MakeInstantFromMillis
millis call Clock1 .SystemTime
set DateLabel . Text to join DatePicker1 . Day
" / "
DatePicker1 . Month
" / "
DatePicker1 . Year
set DueDateButtonArranger . Visible to true
set DueDateThing . Visible to false
set DatePicker1 . Text to join DatePicker1 . Day
" / "
DatePicker1 . Month
" / "
DatePicker1 . Year
initialize global ScreenNumber to 1
initialize global Choosen to " "