Documentation

Memento Android App

Table of Contents

Introduction

This document is a full specification of the Memento Android App.

Memento is an Android App, that lets the user manage his tasks. It has been implemented under independent research, with college professor's guidance. The app is useful for appointment management. The purpose of the app is to ease appointment management and to be a convenient and easy to use, managing your plans. The app is based on a DB where you can add all your appointments and you will be notified when the task deadline arrives. Above all, we hope to provide a confortable user experience, along with efficiency.

Layout

- ListView.axml [1]

XML used to build every element of the list. It has a check box for enabling and disabling the alarm, and two text views to show the name and the date of the memento.

- Main.axml [2]

XML used to build main activity layout. It has three Buttons, one for adding a task, one for sorting by date and one for sorting by priority. It also has the list of tasks.

- TaskEdit.axml [3]

XML used to build task edit activity. It has text boxes for title and description, time picker, radio buttons for priority and buttons for saving and deleting the task.

Task.cs - [4] Data structure

string name – name of the task which is also the primary key for the database
string description – description of the task
Boolean status – bool to check whether alarm for this task is activated or not
DateTime deadline – the time when the user should be notified
int priority – 0, 1, 2 for low, medium, high priority

All the above attributes have getters and setters.

There are also two constructors, one with arguments and one without arguments.

DataBase

- TaskDataBase.cs - [6] Data Structure

A class that uses SQLite library, and it's basically a database

string DatabaseFilePath – attribute that holds the database path. It has a getter and a setter

TaskDatabase(string path) – creates the database at the specified path
Task GetTask(string name) – returns a task from the database based on the primary key: name
IEnumerable<Task> GetTasks() – returns the list of tasks found in the database
int UpdateTask(Task item) – updates task, returns -1 if task does not exist in the database
int SaveTask(Task item) – add task to the database, returns -1 on fail
int DeleteTask (Task item) – removes the task, returns -1 if task not present in database

- TaskRepository.cs - [5] Data Structure

A class that has all the necessary methods needed to work with the task database. It has an instance of the previously explained class, TaskDataBase

Task GetTask(string name) – returns a task from the database based on the primary key: name
IEnumerable<Task> GetTasks() – returns the list of tasks found in the database
int UpdateTask(Task item) – updates task, returns -1 if task does not exist in the database
int SaveTask(Task item) – add task to the database, returns -1 on fail
int DeleteStock(Task item) – removes the task, returns -1 if task not present in database

Notifications

AlarmManagerOverRepeat.cs - [7]
NotificationService.cs - [8]

- AlarmManagerOverRepeat.cs - [7] Utility

It implements the BroadcastReceiver class and it is used to create and publish the notification.

void OnReceive(Context context, Intent intent) iterates the list of tasks, and when it finds a task with the deadline passed, it ships a notification with all the information about the task. It also checks the android version because notification channels are new in API 26, so for lower versions it uses the old way.

- NotificationService.cs - [8] Utility

It implements the Service class and it's used as a background service that, once every 60 seconds calls AlarmManagerOverRepeat to check whether a task met its deadline.

ListAdapter.cs - [9]

It is used to display the list of tasks on the list view from the main activity.

ListAdapter(Activity context, List<Task> items) – constructor that needs the list of tasks to be displayed and the context where it needs to be displayed.

GetItemId(int position) – method for position of the item

View GetView(int position, View convertView, ViewGroup parent) – method that uses the an existing view if it exists and displays the list. It also checks if the user clicks on an item.

ActivityTaskEdit.cs - [10]

It is used for the task edit activity behavior.

void OnCreate(Bundle savedInstanceState) - puts the details in every field of the activity if needed. Manages button clicks for save and delete.

void Save() – saves the current states of the fields and puts them in a Task object, adding it to the list.

void Delete() – deletes the task from the list with the same primary key as the name field from the activity.

MainActivity.cs - [11]

Main activity that uses every other class made.

List<Task> items – the list of tasks

ListAdapter adapter – adapter object used to display the list

void OnCreate(Bundle bundle) – method that sets everything up and gets the items from the layout. It also checks for button clicks and their appropriate response: sort by priority, sort by deadline, toggle item check box, add task, item click.

void OnActivityResult(int requestCode, [GeneratedEnum] Result resultCode, Intent data) – this method is called when an Task element from the list is updated and the list adapter needs to rebuild the list.

References

[1]https://github.com/catalinvlad192/MementoIC/blob/master/MementoIC/MementoIC/Resources/layout/ListView.axml

[2]https://github.com/catalinvlad192/MementoIC/blob/master/MementoIC/MementoIC/Resources/layout/Main.axml

[3]https://github.com/catalinvlad192/MementoIC/blob/master/MementoIC/MementoIC/Resources/layout/TaskEdit.axml

[4]https://github.com/catalinvlad192/MementoIC/blob/master/MementoIC/MementoIC/Resources/Task.cs

[5]https://github.com/catalinvlad192/MementoIC/blob/master/MementoIC/MementoIC/Resources/TaskRepository.cs

[6]https://github.com/catalinvlad192/MementoIC/blob/master/MementoIC/MementoIC/Resources/TaskDatabase.cs

[7]https://github.com/catalinvlad192/MementoIC/blob/master/MementoIC/MementoIC/Resources/AlarmManagerOverRepeat.cs

[8]https://github.com/catalinvlad192/MementoIC/blob/master/MementoIC/MementoIC/Resources/NotificationService.cs

[9]https://github.com/catalinvlad192/MementoIC/blob/master/MementoIC/MementoIC/Resources/ListAdapter.cs

[10]https://github.com/catalinvlad192/MementoIC/blob/master/MementoIC/MementoIC/Resources/ActivityTaskEdit.cs

[11]https://github.com/catalinvlad192/MementoIC/blob/master/MementoIC/MementoIC/MainActivity.cs