

Project Doc of PlantSchedule

Content

Project Doc of PlantSchedule	1
1. Introduce & Requirements analysis	2
2. Function design	2
MainActivity	2
SearchActivity & ResultActivity	3
SpecialInfoActivity & InfoActivity	3
AddActivity	4
MyActivity	4
CurrentActivity	4
EditActivity	5
3. System Design	5
Database Design	5
Tables:	5
DFD	6
Interface & API	7
Entitles Definition	7
Class design	7
E-R diagram	8
4. User guidance	8

1.Introduce & Requirements analysis

PlantSchedule is an app for the gardening and plants lover. It comes up with my mind when I'm thinking of a way to learn more about plants and record whatever plants I meet. And there's no app made for recording plants no matter they are wild-growing or planted in the garden. Many of us amateur plants lover have faced this problem, so I think it's a good way to make an app for them. In spite of many app can record things, they are not designed for plants and they have not got a plants database for record plants. This app separate plants and their records, and it separate one's own plants and the database. It has a clear logic for plants lover and really easy to use.

After thinking of the plants information that everyone meets is too difficult to gather together in one online database. I choose to make an offline app. Using a local database, you can add whatever plants you happen to meet. Use this app to have a close contact to amazing plant world!

2.Function design

Basically, this app is like a recorder and a dairy made specially for plants. Let's learn about those function through each Activity:

- a. MainActivity
- b. SearchActivity
- c. AddActivity
- d. ResultActivity
- e. SpecialInfoActivity
- f. InfoActivity
- g. MyActivity
- h. CurrentActivity
- i. EditActivity
- j. RecordActivity

MainActivity

MainActivity is quite simple. It only has two button and a title for this app. It just a bridge and navigator to other activities.

SearchActivity & ResultActivity

Tap on button “Search” to start this activity. In this activity, you can search for current plants with the help of edittext at the top of this view.

The button at the left is the search button, and when you can’t find the plant you want to see, tap the plus button to make a record of this plant. The database is ready so you can find this current plant in the future.

Meanwhile, you can sild one item to the left, with the help of swipemenulistview, you can delete one item from the list and database at the same time.

After your search, you will arrive at the ResultActivity. If the plant names contain your key, they will be shown in the listview.

If you tap on a item, you will turn to SpecialInfoActivity.



SpecialInfoActivity & InfoActivity

In this activity, you can scan the specific information which you add in AddActivity. The sci-name sometime will be too long to be shown, so it’s designed to be marquee. And you can scroll the description so you will not miss the information.

The plus button at the bottom tells that you can add this plant to “my plants” list, where you can do much more to record your own plants.

InfoActivity is nearly the same. But you will not find a plus button because InfoActivity is used when you want to check on your own plants’ information



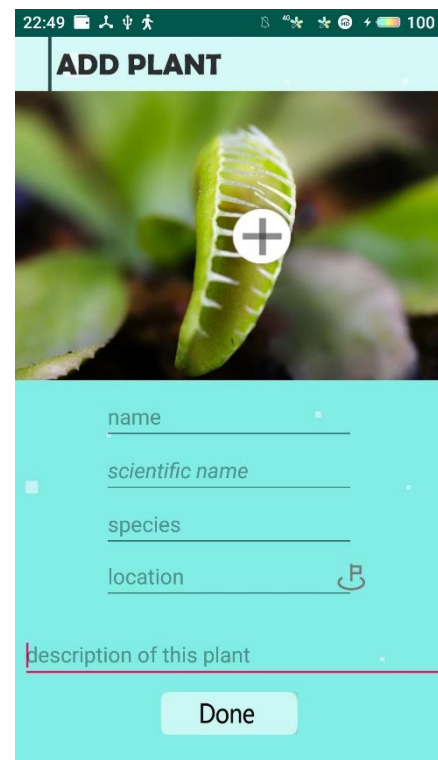
AddActivity

When you can't find your plant in the SearchActivity, just add your own plant!

You can choose take a picture immediately or pick one from your album. And add the description you want to check in the future.

Type all the information you need, and tap on flag button to locate. You can simply record a plant and where you find it.

Press button Done to finish your typing, add this information will be stored in your database unless you choose to delete it.



MyActivity

It's also a list where you can find the plants you choose to add in SpecialInfoActivity. After tap on current plant, you will turn into CurrentActivity.

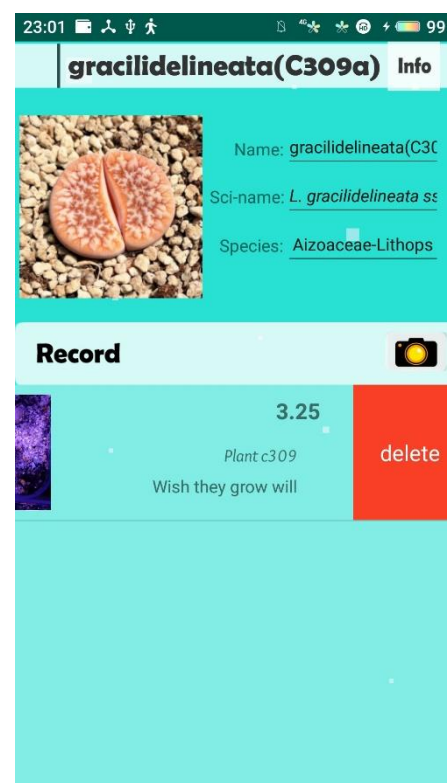
CurrentActivity

It's a activity about one specific plant of your own. Here you can see and add records to your plant. The records contain many information such as weather, location and so on, this will be introduced in EditActivity.

You can scroll and see the whole basic information of plant in the three EditTexts. And you can tap on Info button for more information.

Then you can choose the camera button to take a pic for your plant. Add and an event in EditActivity.

And when you think the records is out of date, or you just don't like one, just slide to left and choose to delete one!




EditActivity & RecordActivity

In this activity, you can record much information about what you have done to your plant. If you think just few choices are not enough, you can write whatever you want to record in the edittext below.

Click ✓ to set up and X to cancel. Your pic will be delete so it will not take your phone space if you choose X.

RecordActivity is almost the same. You can tap on record item in the CurrentActivity to enter this activity. The title RECORD EDITER will be replaced by Event title. And you can choose to share this record with a share button below.



3.System Design

Database Design

This app has a database to save all the plants and records. Attention, this app encourage you to learn the plants you meet, so add plants to database one by one. Because this is an offline app, collection new plants by your own!

Tables:

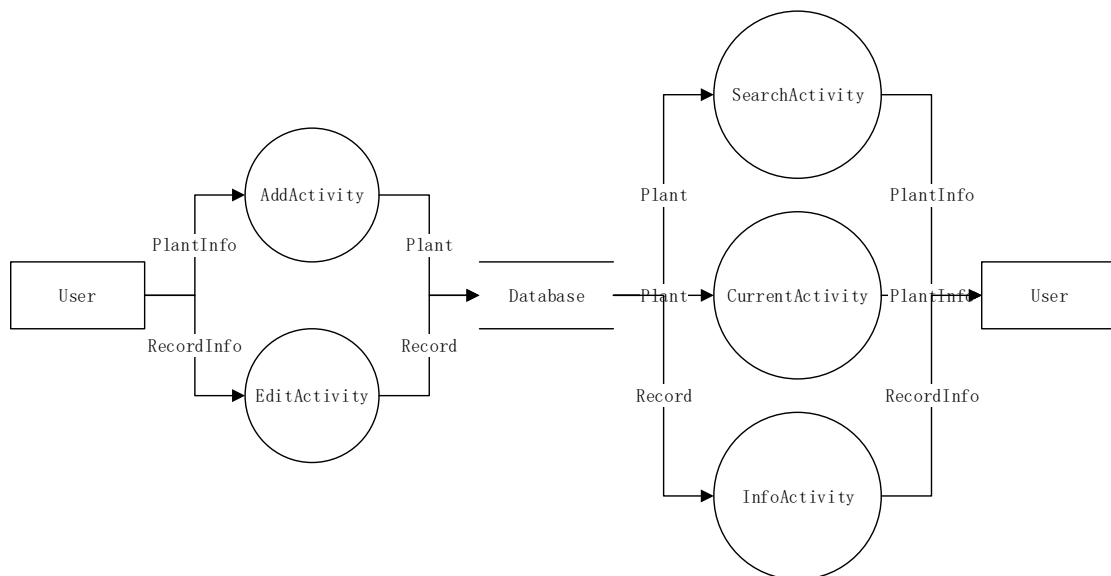
a. plants

NAME	TYPE	CONSTRAINT	DEFAULT	NOTE
name	Text	Primary key		Name of plants
sname	Text		Null	Scientific name
species	Text		Null	Species of plant
ismy	INTEGER	Not null	0	if it's in myplant list
picpath	Text		Null	Path of pics
descri	Text		Null	Description of plants

b. records

NAME	TYPE	CONSTRAINT	DEFAULT	NOTE
_id	Int	Primary key	Autoincrement	Id of records
name	Text	Not null, foreign key		Name of plants
date	Text		Null	Date of records
iswater	Int		0	If it's watering
isferti	Int		0	If it's fertilizing
isdruginf	Int		0	If using drug
weather	Text		Null	Weather of the day
lon	Real		Null	Longitude of the location
lat	Real		Null	Latitude of the location
events_title	Text		Null	Title of the events
events	Text		Null	Operation on your plants
loc	Text		Null	Location
picpath	Text		Null	Path of record pics

DFD



Interface & API

a. Weather forecast API

Link:	http://api.shujuzhihui.cn/api/weather/dailyweather?appKey=efcdee809802446f9e3c5f291195052f&city=Chongqing
Return type	json
Provider	数据智慧 (shujuzhihui)
Interface	“RESULT”—json first layer array
	“weather_now”—json second layer array
	Arr[0] ---- “weather”—weather situation now
	“weather_next”—weather forecast (include today)
	Arr[0] ---- “fd”, “fc”—temperature range from lowest to highest today

b. Reverse geocoding

Link:	http://api.map.baidu.com/geocoder/v2/?callback=renderReverse&location=30.000,106.000&output=json&pois=0&latest_admin=1&ak=Xm1dkkU6dT8B1ply2kOBmKCOwx8BwQjg
Return type	json
Provider	百度地图 (baidu map)
Interface	“RESULT”—json first layer array
	“formatted_address”—address in Chinese

Entitles Definition

Class design

a. Plant:

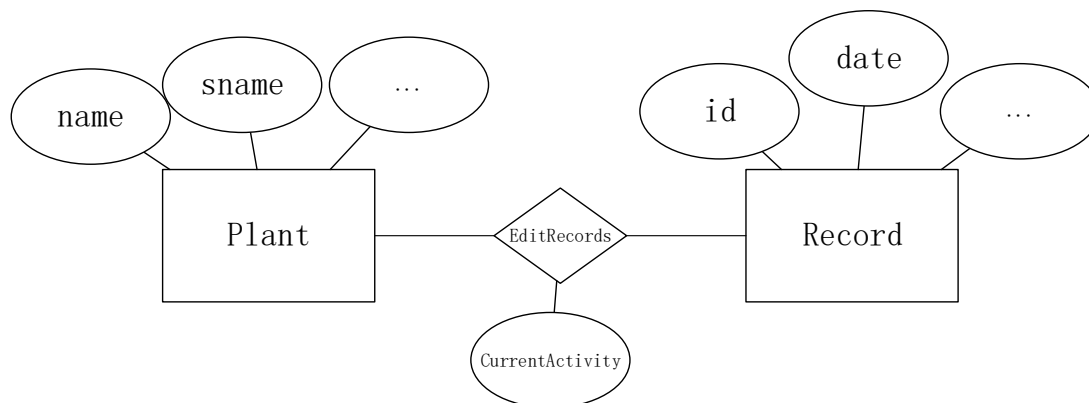
NAME	TYPE	ACCESS PERMISSION	NOTE
name	String	public	
descri	String	public	description
speci	String	public	species
sname	String	public	sci-name
path	String	public	

FUNCTION	DESCRIPTION
Plant()	Constructor without parameter

b. Record:

NAME	TYPE	ACCESS PERMISSION	NOTE
id	int	public	
plantname	String	public	
date	String	public	
iswater	int	public	use as boolean
isferti	int	public	use as boolean
isdrug	int	public	use as boolean
weather	String	public	
lon	double	public	longitude
lat	double	public	latitude
title	String	public	
event	String	public	
loc	String	public	location
path	String	public	

E-R diagram



4. User guidance

Connect to github: <https://github.com/RobbieOvO/PlantSchedule.git>

And Download or clone this project.

A. Run with Android Studio

You should import this project to Android Studio, and after sync process, you can run with your phone connecting to AS or run with an AVD.

B. Run by .apk

The Android apk file is ready in path: .\app\release. You can transport apk file to your phone and run to install it on your phone.

Then, please follow the Function design to use this app. All the functions are well explained in [Function design](#).

Notice that you may find there're no item in SearchActivity, just create an item for your plant and record where to find it!