Session 14: Saving Data & Working with System Permissions

Assignment - 1

ACADGILD

a) What is the difference between Internal Storage & External Storage?

❖ Internal Storage:

✓ Files belong to the Application and will be deleted when user uninstalls the app.
✓ Other users or Application cannot access these private files.
✓ Deleted when user uninstalls the Application.
✓ The Internal Storage is reserved for your Operating System
✓ Memory is less(many Mb's few GB's)
✓ Whenever we reset factory settings, all this storage gets erased.
✓ This memory is reserved for your operating system and personal data.
✓ It will never show up whenever you connect your device to the computer.

❖ External Storage:

✓ External Storage is the storage capacity of your phone. It it the storage which can be removed easily and we can use this storage to save all your songs,pics,videos and etc like SD card.
✓ Only available as long as SD card is mounted on the phone.
✓ These are available to other Applications and users so not secure.
✓ Cannot be deleted when user uninstalls the Application unless they are located inside directory returned by getExternalFilesDir().
✓ Example are pictures or other downloaded files.
✓ More memory in many GB's.
✓ It depends on the compatibility of the memory card slot and to what extent is it supported.It is the storage which can be removed easily by you (your memory card) and can be used for storing pictures, music, videos and the likes. You may or may not be able to install applications on it.

b)For how long the data resides in the cache?

- This is app-specific so when we uninstall the app the cache will be automaticlly cleared.
- If the system gets low on storage, the first place it is going to free resources is from your cache directory.
- The cache directory can typically be cleared manually from the app info screen.
- Clearing the files directory will also clear the cache directory.

- The intent for this directory is store temporary data your application may need to keep around between sessions, but may not be vital to keep them forever.We can access this directory with Context.getCacheDir(). This will show up as "Cache" on app settings
- The system will not clear any data from the files directory.

- Example for cache can be a recently opened mail.Once opened, cache the data so when the user wants to read that email again, it loads instantly rather using the network again to retrieve the same data. It need not be kept forever, because eventually the user will be finished with the email.
- Cache will be deleteted if user uninstalls the application.

- File cacheDir = context.getCacheDir();//gets a reference to the cache directory
- long size = getDirSize(cacheDir);//we can determine the space available in the cahce directory

- getCacheDir()gives reference to internal cache.
- getExternalCacheDir() for external storage specific to app path.

c) What are the critical Permissions and Normal Permissions? What are the examples of each?

➢ Normal Permissions:

- Normal permissions do not directly risk the user's privacy. If your app lists a normal permission in its manifest, the system grants the permission automatically.Many permissions are designated as PROTECTION_NORMAL, which indicates that there's no great risk to the user's privacy or security in letting apps have those permissions.
- There's no great risk in allowing an app to vibrate the device, so that permission is designated as normal.
- Normal permissions cover areas where your app needs to access data or resources outside the app's sandbox, but where there's very little risk to the user's privacy or the operation of other apps. For example, permission to set the time zone is a normal permission. If an app declares in its manifest that it needs a normal permission, the system automatically grants the app that permission at install time. The system does not prompt the user to grant normal permissions, and users cannot revoke these permissions

➢ Normal Permission

✓ ACCESS_LOCATION_EXTRA_COMMANDS
✓ ACCESS_NETWORK_STATE
✓ ACCESS_NOTIFICATION_POLICY
✓ ACCESS_WIFI_STATE
✓ BLUETOOTH
✓ BLUETOOTH_ADMIN
✓ BROADCAST_STICKY
✓ CHANGE_NETWORK_STATE

✓ CHANGE_WIFI_MULTICAST_STATE
✓ CHANGE_WIFI_STATE
✓ DISABLE_KEYGUARD
✓ EXPAND_STATUS_BAR
✓ GET_PACKAGE_SIZE
✓ INSTALL_SHORTCUT
✓ INTERNET
✓ KILL_BACKGROUND_PROCESSES
✓ MODIFY_AUDIO_SETTINGS
✓ NFC
✓ READ_SYNC_SETTINGS
✓ READ_SYNC_STATS
✓ RECEIVE_BOOT_COMPLETED
✓ REORDER_TASKS
✓ REQUEST_IGNORE_BATTERY_OPTIMIZATIONS
✓ REQUEST_INSTALL_PACKAGES
✓ SET_ALARM
✓ SET_TIME_ZONE
✓ SET_WALLPAPER
✓ SET_WALLPAPER_HINTS
✓ TRANSMIT_IR
✓ UNINSTALL_SHORTCUT
✓ USE_FINGERPRINT
✓ VIBRATE
✓ WAKE_LOCK
✓ WRITE_SYNC_SETTINGS

➢ Critical Permissions:

- If you list a dangerous permission, the user has to explicitly give approval to your app.Critical permissions cover areas where the app wants data or resources that involve the user's private information, or could potentially affect the user's stored data or the operation of other apps. For example, the ability to read the user's contacts is a dangerous permission.

- If your app needs a dangerous permission, you must check whether you have that permission every time you perform an operation that requires that permission. The user is always free to

revoke the permission, so even if the app used the camera yesterday, it can't assume it still has that permission today.

- If the app has the permission, the method returns PackageManager.PERMISSION_GRANTED, and the app can proceed with the operation. If the app does not have the permission, the method returns PERMISSION_DENIED, and the app has to explicitly ask the user for permission.
- int permissionCheck = ContextCompat.checkSelfPermission(thisActivity,
- Manifest.permission.WRITE_CALENDAR);
- To check if you have a permission, call the ContextCompat.checkSelfPermission() method.
- If the app has the permission, the method returns PackageManager.PERMISSION_GRANTED, and the app can proceed with the operation. If the app does not have the permission, the method returns PERMISSION_DENIED, and the app has to explicitly ask the user for permission.
- These permissions have to be specified in the manifest file of the Application.These Applications may or maynot be granted these permisions.It has to request for the permission and can be revoked anytime.
- For example, users would reasonably want to know whether an app can read their contact information, so users have to grant this permission explicitly.

➢ Dangerous permissions :

✓ READ_CALENDAR
✓ WRITE_CALENDAR
✓ CAMERA
✓ READ_CONTACTS
✓ WRITE_CONTACTS
✓ GET_ACCOUNTS
✓ ACCESS_FINE_LOCATION
✓ ACCESS_COARSE_LOCATION
✓ RECORD_AUDIO
✓ READ_PHONE_STATE
✓ CALL_PHONE
✓ READ_CALL_LOG
✓ WRITE_CALL_LOG
✓ ADD_VOICEMAIL
✓ USE_SIP

- ✓ PROCESS_OUTGOING_CALLS
- ✓ BODY_SENSORS
- ✓ SEND_SMS
- ✓ RECEIVE_SMS
- ✓ READ_SMS
- ✓ RECEIVE_WAP_PUSH
- ✓ RECEIVE_MMS
- ✓ READ_EXTERNAL_STORAGE
- ✓ WRITE_EXTERNAL_STORAGE