Lessons Learned

Team Name: Superfly
Members:
Run Yang Andrew ID: runy ;
Mujing Zhou Andrew ID: mujingz;
Haoran Yu Andrew ID haorany:

1.An Activity is a single focused thing that is used for interaction with user. In an Android Project, Activity class takes care of creating a window in which developers can place their UI. Developer can adjust the size of window and manage UI very conveniently with Fragments. With a interface based on Activity, the Android project will be easy to use, manage and test.

2.In an android project without an activity, there is no interface and the app can not interact with user directly. The program will only run in background and only act as a background service like Google Play Services. It do not have any activity mentioned in AndroidManifest.xml and is not easy to start/close. Without UI, the program is not easy to test and debug, you need to ask other program to send intent to start it and close it after stoping the service.

3.An android application is a collection of activities and an activity correlates to a screen. An android application is mainly composed of four parts: Activity, Service, BroadcastReceiver and Content Provider. The execution lifecycle contains lifestyles of these four parts.

4.Life cycle of Activity, there are four activity states: Running(Activity is visible and interacts with the user), Paused(Activity is still visible but partially obscured, instance is running but might be killed by the system),Stopped(Activity is not visible, instance is running but might be killed by the system),Killed(Activity has been terminated by the system of by a call to its finish() method).

5.For Service, the lifecycle starts from onCreate() and ends after calling of onDestroy().We can use Context.startService() to start a service and end it with Context.stopService(). Or we can use Context.bindService() to start and use Context.unbindService() to stop.

6.For a normal mobile application, its working state contains ‘not running’, ‘inactive’, ‘active’, ‘background’ and ‘suspended’.

7.Usually,we use android studio for development of android application. At the core of Android Studio is an intelligent code editor capable of advanced code completion, refactoring, and code analysis.we can start projects using template code for patterns such as navigation drawer and view pagers, and even import Google code samples from GitHub. We can also build apps for Android phones, tablets, Android Wear, Android TV, Android Auto and Google Glass.

8.We can create multiple APKs for your Android app with different features using the same project, manage app dependencies with Maven and build APKs from Android Studio or the command line.

9.By default, the Android SDK does not include everything you need to start developing. The SDK separates tools, platforms, and other components into packages you can download as needed using the Android SDK Manager. So before you can start, there are a few packages you should add to your Android SDK.

10.To start adding packages, launch the Android SDK Manager in one of the following ways:
In Android Studio, click SDK Manager in the toolbar. For Mac/Linux, we need to open a terminal and navigate to the tools/ directory in the location where the Android SDK was installed, then execute android SDK.

11.Java is an object-oriented designed programming language, it is based on class(blue print),object(made using a class) and method(usually called by object).

12.JVM - Java Virtual Machine is an environment for classes files containing methods expressed in the Java byte code.

13.There are static variables and methods, which means we can call them will class name directly. We do not need to use object to call static method, static variables can be shared among threads.

14.object relationships contains Association, Containment, Inheritance, Polymorphism, Encapsulation and so on.Dynamic Binding means the code associated with a given procedure call is not known until the time of the call at runtime. Dynamic binding is associated with Polymorphism and

Inheritance.

15.Java Access Modifier - public, private, protected, default. ‘public’ variables and methods can be shared and used anywhere, ‘protected’ variables and methods can not be used from a class outside of a package, ‘default’ variables and methods can not be used from a class outside of a package and from a subclass located outside of the package, ‘private’ variables and methods can only be used within the class.

16.FileIo tool includes Stream, Buffer, Filter and Object. Streams are used to connect a file to a program, Buffers can save small pieces of memory to a stream, Filters refers to data conversion capability,can, Objects persist an object from memory to disk.

17.Exception contains Checked Exception and Unchecked Exception. Checked Exception means the exception which JVM is aware of and require try-catch. Unchecked Exception means runtime exception which is created at run time. It offers some flexibility in implementation of custom exception management.

18.Interface and abstract class are good thing to improve code reuse and information hiding , a interface only provides methods and can not be instantiated by itself. Abstract class can be extended by subclasses to extend its function.

19.We can use serialization to save file to disk or through web in byte, and deserialized the byte data to form a original object from what is saved in disk or from web.

20.Java data structures contains list(ArrayList, LinkedList) - small to medium data structures, set(HashSet, TreeSet) - a group of unique values, map(HashMap, TreeMap) - key and value pairs, key should be unique and values should be duplicate.

21.Multitasking can be divided to (1) Non-preemptive – once CPU gives attention to the process, it must complete the allocated block of time, (2) Pre-emptive – If CPU has given attention to the process, it doesn’t have to complete the allocated block of time, if the next process of higher priority comes into picture.

22. Support for multi-threading in java is through the Thread class. A

thread can occupy one of four states :spawned, running, blocked, dead. Java supports thread priorities, supports thread synchronization, supports explicit thread control

23. To create a thread, a instance of any class can become a thread through sub-classing
,class Widget extends Thread, Any object instance of a class can become a thread through interface implementation or class Widget implements Runnable.The run method is the body of the thread and should beoverridden. run is defined in Thread and has a null-action if not overridden

24.Monitors are locks on objects implemented by the JVM to facilitate synchronization. all the flavors of Object.wait() can cause a thread to yield a monitor it has obtained on an object and wait to be notified when the monitor is available again. the methods Object.notify() and Object.notifyAll() are used to tell threads waiting for a monitor that it's about to become available.

25.We need to lock data in critical sections of code to prevent two threads from modifying it at the same time. We should avoid starvation: one thread is locked out from access to a resource and can't do anything. We should avoid deadlock: two or more threads are waiting for each other to release resources the other has the classic Dining Philosophers' problem.

26.Deadlock occurs when two or more threads are waiting on common resources that are not exclusively owned.

27.Threads can usually be grouped into consumers or producers. Producers produce output. Consumers require input. Producer - Consumer relationships can be modeled as client - server threads. Communication between them can be achieved through. PipedOutputStream and PipedInputStream.

28. A Socket is an abstraction of a "communications link" between machines over some network. Socket communication is the same regardless of whether the network connection is via a phone line, cable modem, ethernet, or fiber-optic line.

29 A packet is a discrete quantity of information suitable for routed transport over a shared network. Packet sizes are limited, so a packet

may be a fragment of a large file or message.

30.The Hypertext Transfer Protocol (HTTP) is an application-level protocol for distributed, collaborative, hypermedia information systems. HTTP has been in use by the World-Wide Web global information initiative since 1990. The first version of HTTP, referred to as HTTP/0.9, was a simple protocol for raw data transfer across the Internet.

31.All HTTP transaction follow the same format for client and server transactions.
Each client transaction has 3 parts: The request or response line. A Header Section. After the header information additional information - data for CGI programs etc might be sent using the POST method.

32.Get - is a request for information located at a specified URI on the server. Get asks for a file usually. Head - is a request like get but doesn't send back the entire file but only returns the header information.

33.Post - is a request that allows data to be sent to the server on a client request. The data is typically passed to the data handling program from the webserver - e.g. CGI, etc.

34.proxy is an intermediary program which acts as both a server and a client for the purpose of making requests on behalf of other clients. Requests are serviced internally or by passing them on, with possible translation, to other servers. A proxy must implement both the client and server requirements of this specification.

35.database objects includes: Table – Basic unit of storage, composed of rows and columns,
View – Subsets of data from one or more tables, Sequence – Generates primary key values, Index – Improves the performance of queries, Synonymn – Gives alternative names to objects.

36.different packages should be made in over to encapsulation the whole system and make it easy to control.

37.some classes can be set as inside classes, such as option to options in oder to better control it.

38.FileIO, we have some methods to get the file from a text file and we should pay attention to the method of buffer reader and the split method.

39.We should make diagrams in order to test everything in the program. And it helps us to make a planning of the whole system.

40.We have to write some test file to test the whole problem and find whether we have rightly construct the whole program.

41.The action bar is one of the most important design elements you can implement for your app's activities. It provides several user interface features that make your app immediately familiar to users by offering consistency between other Android apps.

42.The interaction between activities is based on the use of intent. There are two ways to send object with intent. 1. Use Serializable interface to send object. For example, Bundle.putSerializable(key,Object); 2. Use Parcelable interface to send object. For example, Bundle.putParcelable(key,Object)

43.Usually we use Log to record and solve exception. Like Log.e, Log.d, Log.i. For example, Log.e(String tag, String msg, Throwable tr) will show a message with a tag on the Error screen.

44.Android devices come in many shapes and sizes all around the world. With a wide range of device types, you have an opportunity to reach a huge audience with your app. In order to be as successful as possible on Android, your app needs to adapt to various device configurations. Some of the important variations that you should consider include different languages, screen sizes, and versions of the Android platform. We need ti design different resolution for different screens and devices.

45.To create a dynamic and multi-pane user interface on Android, you need to encapsulate UI components and activity behaviors into modules that you can swap into and out of your activities. You can create these modules with the Fragment class, which behaves somewhat like a nested activity that can define its own layout and manage its own lifecycle.

46.To allow a Fragment to communicate up to its Activity, you can define an interface in the Fragment class and implement it within the Activity. The Fragment captures the interface implementation during its onAttach() lifecycle method and can then call the Interface methods in order to communicate with the Activity.

47.If you want an activity to communicate with a fragment. The host activity can deliver messages to a fragment by capturing the Fragment instance with findFragmentById(), then directly call the fragment's public methods.

48.An Intent can be explicit in order to start a specific component (a specific Activity instance) or implicit in order to start any component that can handle the intended action (such as "capture a photo").

49.SQLite is an Open Source database. SQLite supports standard relational database features like SQL syntax, transactions and prepared statements. The database requires limited memory at runtime (250 KB) which makes it a good candidate from being embedded into other runtimes.

50.SQLite is embedded into every Android device. Using an SQLite database in Android does not require a setup procedure or administration of the database.You only have to define the SQL statements for creating and updating the database. Afterwards the database is automatically managed for you by the Android platform.Access to an SQLite database involves accessing the file system. This can be slow. Therefore it is recommended to perform database operations asynchronously.

51.To create and upgrade a database in your Android application you create a subclass of the SQLiteOpenHelper class. In the constructor of your subclass you call the super() method of SQLiteOpenHelper, specifying the database name and the current database version.In this class you need to override the following methods to create and update your database.onCreate() - is called by the framework, if the database is accessed but not yet created.onUpgrade() - called, if the database version is increased in your application code. This method allows you to update an existing database schema or to drop the existing database and recreate it via the onCreate() method.Both methods receive an SQLiteDatabase object as parameter which is the Java representation of the database.

52.It is good practice to create a separate class per table. This class defines static onCreate() and onUpgrade() methods. These methods are called in the corresponding methods of SQLiteOpenHelper. This way your implementation of SQLiteOpenHelper stays readable, even if you have several tables.

53.SQLiteDatabase is the base class for working with a SQLite database in Android and provides methods to open, query, update and close the database. It provides the insert(), update() and delete() methods. The object ContentValues allows to define key/values. The key represents the table column identifier and the value represents the content for the table record in this column. ContentValues can be used for inserts and updates of database entries.

54.One good example of query() of SQLite is like below:
return database.query(DATABASE_TABLE, new String[] { KEY_ROWID, KEY_CATEGORY, KEY_SUMMARY, KEY_DESCRIPTION }, null, null, null, null, null);

55.A query returns a Cursor object. A Cursor represents the result of a query and basically points to one row of the query result. This way Android can buffer the query results efficiently; as it does not have to load all data into memory.To get the number of elements of the resulting query use the getCount() method.To move between individual data rows, you can use the moveToFirst() and moveToNext() methods. The isAfterLast() method allows to check if the end of the query result has been reached.

56.Cursor provides typed get*() methods, e.g. getLong(columnIndex), getString(columnIndex) to access the column data for the current position of the result. The "columnIndex" is the number of the column you are accessing.Cursor also provides the getColumnIndexOrThrow(String) method which allows to get the column index for a column name of the table.A Cursor needs to be closed with the close() method call.

57.A provider allows applications to access data. The data can be stored in an SQLite database, on the file system, in flat files or on a remote server.While a content provider can be used within an application to access data, its is typically used to share data with other application. As application data is by default private, a content provider is a convenient to share you data with other application based on a structured interface.A content provider must be declared in the manifest file for the application.

58.We can use Google Map API and Android location API to realize location function.The Google Location Services API, part of Google Play services, is the preferred way to add location-awareness to your app. It offers a simpler API, higher accuracy, low-power geofencing, and more. If you are currently using the android.location API, you are strongly encouraged to switch to the Google Location Services API as soon as possible.

59.The IntentService class provides a straightforward structure for running an operation on a single background thread. This allows it to handle long-running operations without affecting your user interface's responsiveness.

60.To create a work request and send it to an IntentService, create an explicit Intent, add work request data to it, and send it to IntentService by calling startService().

61.Once you call startService(), the IntentService does the work defined in its onHandleIntent() method, and then stops itself.

62.A CursorLoader runs an asynchronous query in the background against a ContentProvider, and returns the results to the Activity or FragmentActivity from which it was called. This allows the Activity or FragmentActivity to continue to interact with the user while the query is ongoing.

63.To use a CursorLoader with an Activity or FragmentActivity, use the LoaderCallbacks<Cursor> interface. A CursorLoader invokes callbacks defined in this interface to communicate with the class; this lesson and the next one describe each callback in detail.

64.Amazon Simple Storage Service (Amazon S3), provides developers and IT teams with secure, durable, highly-scalable object storage. Amazon S3 is easy to use, with a simple web services interface to store and retrieve any amount of data from anywhere on the web. With Amazon S3, you pay only for the storage you actually use. There is no minimum fee and no setup cost.

65.Amazon S3 can be used alone or together with other AWS services such as Amazon Elastic Compute Cloud (Amazon EC2), Amazon Elastic Block Store (Amazon EBS), and Amazon Glacier, as well as third party storage repositories and gateways. Amazon S3 provides cost-effective object storage for a wide variety of use cases including cloud applications, content distribution, backup and archiving, disaster recovery, and big data analytics.

66.searchView is a widget that provides a user interface for the user to enter a search query and submit a request to a search provider. Shows a list of query suggestions or results, if available, and allows the user to pick a suggestion or result to launch into.When the SearchView is used in an ActionBar as an action view for a collapsible menu item, it needs to be set to iconified by default using setIconifiedByDefault(true). This is the default, so nothing needs to be done.

67.The Android platform enables you to extend your app to work with in-vehicle console systems running Android Auto. These systems provide a simplified interface for apps that can be used in a car, allowing users to take your app with them on the way to the grocery store or on a long road trip.

68.In the mysql we have different ways to write it. First, we can write it int he linux window and then we can write it in java.

69.Whenever we want to use it we should first create a my class and then create the connect so that we can access it.

70.Different packages should be built to help manage classes. For instance, all test classes should be hold in package driver. For all the constant variables, we should use upper case.

71.For the SQL Function about date. Oracle stores dates in an internal numeric format: century, year, month, day, hours, minutes, seconds.
The default date format is DD-MON-YY.SYSDATE is a function returning date and time. DUAL is a dummy table used to view SYSDATE.

72.NVL Function is used to convert null to an actual value. Datatypes that can be used are date, character, and number and datatypes must match.

73.JDBC (Java Database Connectivity) provides a standard library for accessing relational databases. It standardizes the mechanism for connecting to databases, sending queries, committing transactions etc. It does not standardize the SQL syntax.

74.To load a driver you need to load the appropriate class. In order to do that, you have to make a call to Class.forName method. This method takes String argument, which is a full class name (i.e. one that includes the package name) and loads the appropriate class. You will need the JDBC driver specific to the database, you are using.

75.Once you have loaded the JDBC driver, you need to specify the database server. URL's referring to databases use the JDBC : protocol and have the server host, port and database name embedded within the URL.

76.The simplest way to handle results is to process them one row at a time, using ResultSet's next method to move through the table one row at a time. Within a row, the ResultSet provides various getXxx methods that take a column index or name as an argument and return the results as a variety of different Java types. For example, getInt can be used to get integer datatype.

77.Pay attention, connection need to be close after using. To close the connection, you will use this statement: connection.close().

78.Uncompressed data like images or data file can be mapped into memory using a call to mmap(). On 32 bit processors data should be aligned on a 4- byte memory boundary – entire data set in .apk file looks like data in memory If data is not aligned to processor requirements then read/writes can cost extra time.

79.For situations where the volume of data is relatively light (less than a megabyte), like the user's preferences, notes, game high scores or other stats, the Backup API provides a lightweight solution. This lesson walks you through integrating the Backup API into your application, and restoring data to new devices using the Backup API.

80.The easiest way to create your backup agent is by extending the wrapper class BackupAgentHelper. Creating this helper class is actually a very simple process. Just create a class with the same name as you used in the manifest in the previous step (in this example, TheBackupAgent), and extend BackupAgentHelper. Then override the onCreate().

81.Inside the onCreate() method, create a BackupHelper. These helpers are specialized classes for backing up certain kinds of data. The Android framework currently includes two such helpers: FileBackupHelper and SharedPreferencesBackupHelper. After you create the helper and point it at the data you want to back up, just add it to the BackupAgentHelper using the addHelper() method, adding a key which is used to retrieve the data later. In most cases the entire implementation is perhaps 10 lines of code.

82.In all cases, the only secure way to offer a file from your app to another app is to send the receiving app the file's content URI and grant temporary access permissions to that URI. Content URIs with temporary URI access permissions are secure because they apply only to the app that receives the URI, and they expire automatically. The Android FileProvider component provides the method getUriForFile() for generating a file's content URI.

83.A "touch gesture" occurs when a user places one or more fingers on the touch screen, and your application interprets that pattern of touches as a particular gesture. There are correspondingly two phases to gesture detection:
1.Gathering data about touch events. 2.Interpreting the data to see if it meets the criteria for any of the gestures your app supports.

84.When a user places one or more fingers on the screen, this triggers the callback onTouchEvent() on the View that received the touch events. For each sequence of touch events (position, pressure, size, addition of another finger, etc.) that is ultimately identified as a gesture, onTouchEvent() is fired several times.The gesture starts when the user first touches the screen, continues as the system tracks the position of the user's finger(s), and ends by capturing the final event of the user's fingers leaving the screen. Throughout this interaction, the MotionEvent delivered to onTouchEvent() provides the details of every interaction. Your app can use the data provided by the The getFromLocation() method provided by the Geocoder class accepts a latitude and longitude, and returns a list of addresses. The method is synchronous, and may take a long time to do its work, so you should not call it from the main, user interface (UI) thread of your app.MotionEvent to determine if a gesture it cares about happened.

85.To create a work request and send it to an IntentService, create an explicit Intent, add work request data to it, and send it to IntentService by calling startService().
Once you call startService(), the IntentService does the work defined in its onHandleIntent() method, and then stops itself.

86.Using the Geocoder class in the Android framework location APIs, you can convert an address to the corresponding geographic coordinates. This process is called geocoding. Alternatively, you can convert a geographic location to an address. The address lookup feature is also known as reverse geocoding.

87.The last known location of the device is a useful starting point for the address lookup feature. The lesson on Getting the Last Known Location shows you how to use the getLastLocation() method provided by the fused location provider to find the latest location of the device.To access the fused location provider, you 88.need to create an instance of the Google Play services API client. To learn how to connect your client, see Connect to Google Play Services.In order for the fused location provider to retrieve a precise street address, set the location permission in your app manifest to ACCESS_FINE_LOCATION