Notes for final exam

Chapter 4_1

Data savings methods

Shared Preferences

- Store small amounts of data in key value pairs.
- For example, we can hold and store the comment yet to post as a draft such as [key = comment_id, value = comment_description] since the data will persist across sessions even if the app is killed. When the next time the user is back to the app, he/she still can retrieve the comment draft and post it out.

Data Files

- A File APIs to store large amount of the text, sound, images, documents etc
- For example, images of the plastics taken by the users can be stored in the data files in different format such as png, jpg, jpeg, etc

Internal Storage	External Storage
Internal storage is always available on all devices, making it a more reliable place to put data on which your app depends	External storage not always be available
The files stored in the internal storage can only be accessed by the app itself	The files stored in the external storage are world-readable
Stores small files, usually sensitive information since it is private such as app setting files	Storing larger files or app, usually for non-sensitive data such as media files and documents
Files in internal storage are non-removable	Files in external storage are removable

Database: SQLite

- A relational database to store structured and repeated data.
- For example, all the profile information can be stored inside a tables called user with data fields, user_id (INTEGER), username (TEXT), contact_number (TEXT), password (TEXT), gender (TEXT), profile_image (BLOB) since all the logged in user will has the same fixed fields and column in the database.

View Model

The ViewModel class is designed to store and manage UI-related data in a lifecycle conscious way.

ViewModel are responsible for preparing and holding the UI data. ViewModel objects are automatically retained during configuration changes so that data they hold is immediately available to the next activity or fragment instance. If the activity is re-created, it receives the same view model instance that was created by the first activity.

Live Data
LiveData is used to build data objects that notify views when the underlying database changes.

Coroutine

- Asynchronous programming
- Manage long-running tasks that could otherwise block the main thread.
- Provide main-safety or safely call the network or disk IO operation
- Launch & Async

Chapter 4_2

Foreground process

User interaction
Is the foreground process the same with foreground services?

Background process

Decoding bitmap, accessing storage, performing network request

Mobile Back-end options

DIY
Create own server - has complete control over the h/w, s/w, network, and services.

Subscribe
Back-end as a services

Mix and Bang
DIY + Subscribe

Back-end server function(s)

Push notification - Azure consists of personalized push notifications, which is, it can instantly send push notifications to Windows, Android, iOS or even Nokia X. Notification Hubs can be hooked to the backend of any existing app, regardless of whether the backend is hosted on Azure or on-premises.

Storage - To store and serve user-generated content, such as photos or videos. For example, cloud storage for Firebase is a powerful, simple, and cost-effective object storage service built for Google scale. The Firebase SDKs for Cloud Storage add Google security to file uploads and downloads for your Firebase apps, regardless of network quality.
You can use our SDKs to store images, audio, video, or other user-generated content.

Social Integration - It allows the user to link with their social media profiles such as Google or Facebook using API. Upon authenticating with these services, you can incorporate additional native integration like social activity lists.

Usage analytics - It allows administrators to find out when people use our app and what pages they're most interested in. They can also find out where your users are located and what browsers and operating systems they use. Knowing how people use your application lets you focus your development work on the scenarios that are most important to them, and gain insights into the goals that they find easier or more difficult to achieve.

Chapter 5

Methods to obtain user location

GPS	WiFi / Cell Tower
Although GPS is most accurate, it only works outdoors, it quickly consumes battery power, and doesn't return the location as quickly as users want.	Android's Network Location Provider determines user location using cell tower and Wi-Fi signals, providing location information in a way that works indoors and outdoors, responds faster, and uses less battery power.

Challenges in determining user location

- **Multitude of location sources**
  GPS, Cell-ID, and Wi-Fi can each provide a clue to users location. Determining which to use and trust is a matter of trade-offs in accuracy, speed, and battery-efficiency.
- **User movement**
  Because the user location changes, you must account for movement by re-estimating user location every so often.
- **Varying accuracy**
  Location estimates coming from each location source are not consistent in their accuracy. A location obtained 10 seconds ago from one source might be more accurate than the newest location from another or same source.

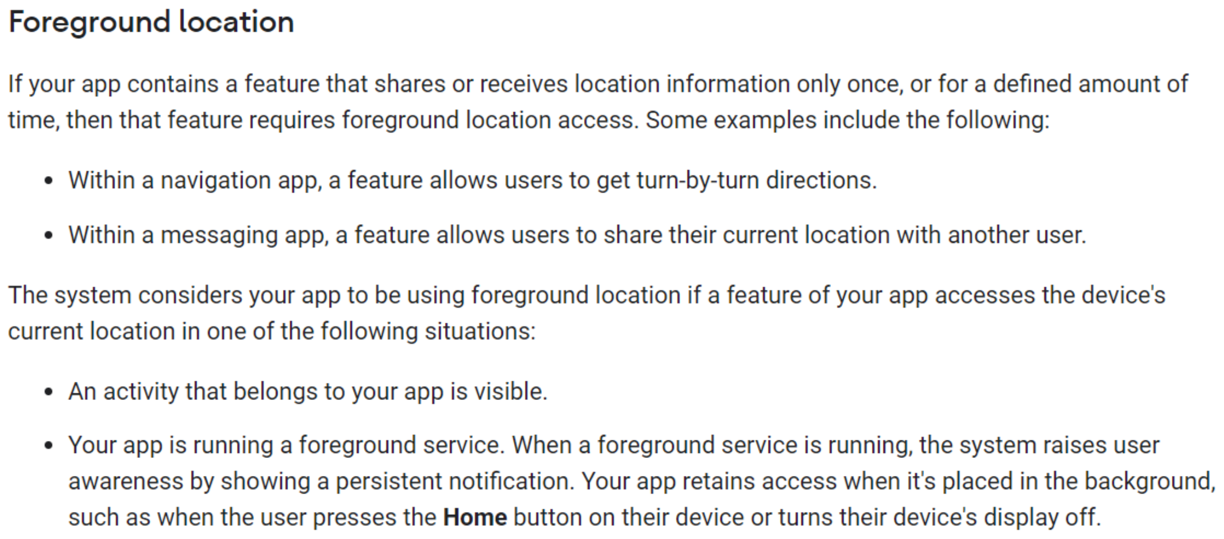
Location based service model

Accuracy

Update Interval

Power consumption

Types of Location permission



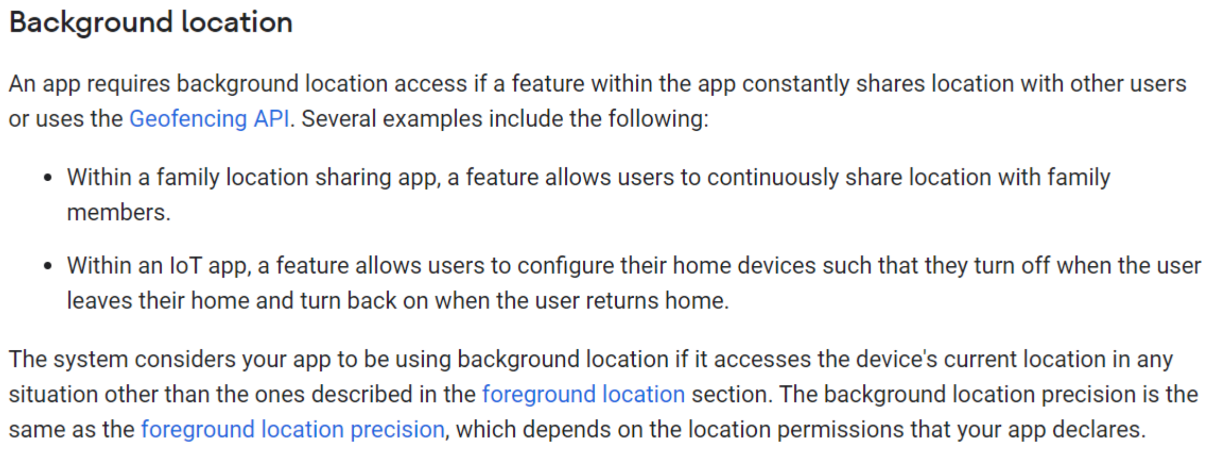
Foreground location

If your app contains a feature that shares or receives location information only once, or for a defined amount of time, then that feature requires foreground location access. Some examples include the following:

- Within a navigation app, a feature allows users to get turn-by-turn directions.
- Within a messaging app, a feature allows users to share their current location with another user.

The system considers your app to be using foreground location if a feature of your app accesses the device's current location in one of the following situations:

- An activity that belongs to your app is visible.
- Your app is running a foreground service. When a foreground service is running, the system raises user awareness by showing a persistent notification. Your app retains access when it's placed in the background, such as when the user presses the **Home** button on their device or turns their device's display off.



Background location

An app requires background location access if a feature within the app constantly shares location with other users or uses the Geofencing API. Several examples include the following:

- Within a family location sharing app, a feature allows users to continuously share location with family members.
- Within an IoT app, a feature allows users to configure their home devices such that they turn off when the user leaves their home and turn back on when the user returns home.

The system considers your app to be using background location if it accesses the device's current location in any situation other than the ones described in the foreground location section. The background location precision is the same as the foreground location precision, which depends on the location permissions that your app declares.

Chapter 6

Consideration on using camera

Camera requirements? If you do not want the user to download the application on a device that does not have a camera, declare the camera requirement in your manifest.

Quick Picture or Customized Camera? For getting a quick snap or clip, consider Using Existing Camera Apps. For developing a customized camera feature, check out the Building a Camera App section.

Using Existing Camera Apps

A quick way to enable taking pictures or videos in your application without a lot of extra code is to use an Intent to invoke an existing Android camera application. A camera intent makes a request to capture a picture or video clip through an existing camera app and then returns control back to your application.

Building a Camera Appl

Some developers may require a camera user interface that is customized to the look of their application or provides special features. Creating a customized camera activity requires more code than using an intent, but it can provide a more compelling experience for your users.

Mobile sensor / Hardware sensor

GPS Position Sensor

The GPS sensor in the phone uses the satellite's instantaneous position to calculate the distance between the mobile phone and the satellite-based on the time difference between the timestamp of the satellite's transmit coordinate and the time of reception (the total number of seconds so far). It can be used for navigation, location, speed measurement, and distance measuring.

Temperature Sensor (Thermometers)

To keep track of the phone's and battery's internal temperatures. The phone will shut down if the temperature of a certain component is discovered to be too high, in order to prevent harm to the phone. The temperature sensor can detect temperature changes in the outside air, as well as the user's current body temperature, among other things.

Magnetometers

Measures the magnetic fields and tell you which way is north by varying its voltage output to the phone

Ambient light sensor

Boost phone camera performance - Detect the intensity of light around the user environment and control the screen brightness accordingly. Utilizing a light sensor to help regulate the screen brightness can extend battery life even further.

Gyroscope sensor
Can be used in photo and video apps for image stabilization. It can also be used by fitness tracking devices in combination with an accelerometer and optical sensor that measures pulse.

Chapter 7

Distribution Channel Alternative distribution options | Google Play | Android Developers

Distributing through an app marketplace

Usually, to reach the broadest possible audience, you'd distribute your apps through a marketplace, such as Google Play.
Google Play is the premier marketplace for Android apps and is particularly useful if you want to distribute your apps to a large global audience. However, you can distribute your apps through any app marketplace you want or use multiple marketplaces.
Unlike other forms of distribution, Google Play allows you to use the In-app Billing service and Licensing service. The In-app Billing service makes it easy to sell in-app products like game jewels or app feature upgrades. The Licensing service helps prevent unauthorized installation and use of your apps.

Distributing your apps by email

A quick and easy way to release your apps is to send them to users by email. To do this, you prepare the app for release, attach it to an email, and send it to a user. When the user opens your email on their Android-powered device, the Android system recognizes the APK and displays an Install Now button in the email message. Users can install your app by touching the button. Users need to opt in for installing unknown apps if they haven't already to proceed with the installation.
Distributing apps through email is convenient if you're sending them to a few trusted users, as it provides few protections from piracy and unauthorized distribution; that is, anyone you send your apps to can simply forward them to others.

Distributing through a website

If you don't want to release your apps on a marketplace such as Google Play, you can make them available for download on your website or server, including on a private or enterprise server. To do this, first prepare your apps for release in the normal way, then host the release-ready APK files on your website and provide users with a download link. To install an app distributed in this way, users must opt-in for installing unknown apps.

User opt-in for installing unknown apps

Android protects users from inadvertent download and install of unknown apps, or apps from sources other than Google Play, which is trusted. Android blocks such installs until the user opts into allowing the installation of apps from other sources. The opt-in process depends on the version of Android running on the user's device:

Distribution Methods

Android Go

Android Go is a stripped-down version of the Android operating system, designed for low-end smartphones (normally less than 2GB ram). Android Go runs lighter and saves data, making it more possible on so many devices. In order to make the app compatible with Android Go,

1. Shrink the APK size to less than 40 MB. Less APK size boost app download.
2. Optimize the mobile app for memory usage, mostly likely with RAM usage under 50MB.
3. Ensure short startup time for the mobile app. It should launch within 5 seconds or less.

App bundle

Android App Bundle is a publishing format that includes all your app’s compiled code and resources, and defers APK generation and signing to Google Play.

Google Play uses your app bundle to generate and serve optimized APKs for each device configuration, so only the code and resources that are needed for a specific device are downloaded to run your app. You no longer have to build, sign, and manage multiple APKs to optimize support for different devices, and users get smaller, more-optimized downloads (dynamic delivery).

Publishing with Android App Bundles helps your users to install your app with the smallest downloads possible and increases the compressed download size limit to 150 MB.

Google Play Instant

Google Play Instant enables native apps and games to launch on devices running Android 5.0 (API level 21) or higher without being installed. You can build these types of experiences, called instant apps and instant games, using Android Studio. By allowing users to run an instant app or instant game, known as providing an instant experience, you improve your app or game's discovery, which helps drive more active users or installations.

Chrome OS

Supports the Google Play Store and Android apps

Monetization Apps

Premium

Users purchase the app from an app store for a one-time fee.One problem with paid apps is that, even with plenty of positive reviews, users are unlikely to purchase an app that they can’t try or preview before paying. There is such an abundance of free apps available, why purchase?

Of course, the likelihood users will take the leap and pull the trigger on a paid app depends on your product and audience. The more compelling the content or service you're offering (extensive features), the more likely they are to give in to premium apps.

Freemium

The most effective way to monetize apps. In the freemium model, users can download an app for free, but they cannot access the full set of features without upgrading to the paid version. Freemium apps typically make more money than paid apps, but it may take more time to build profits. If users choose to keep using the free app version, you can still earn some revenue by offering upgrades and premium content.

Subscription

It enables users to download the app for free, but requires a subscription to bypass feature or content restrictions.

Subscription apps offer users access to a particular service or content for a weekly, monthly, or annual fee.

Subscription apps haven't gained the same popularity as other options, but they do offer significant revenue potential. Subscriptions offer the benefit of a steady stream of income, but this model only works well for apps that continually offer fresh content.

Advertisement

Choose to make their mobile app completely free by using advertisements to generate revenue.

Like in-app purchases, advertising is a monetization model that is often combined with freemium or free-to-play apps. In general, users are willing to deal with ads in exchange for accessing your service or content at no cost. But this is usually contingent upon the way ads are integrated into the app.

Services

Users can download an app for free. It's an extension of physical or online services. For example, the Public Bank developed its app to allow users to do online transactions instead of needing to go to the bank physically.

Rewarded product

Users can download an app for free. They can get free stuff or earn rewards by performing simple tasks such as sharing content on social media, scan QR code, answer survey questions, etc

Data Collection

Users can download an app for free, but the client preferences or habits can be collected by the app. Afterward, selling those extensive data to interested third parties. If the app can generate a significant amount of data on client preferences or habits, then it is possible that this data would be useful for other businesses, whose products rely on knowing what people want and need.

E-commerce

This model is also straightforward – your application is a marketplace through which you sell physical goods.

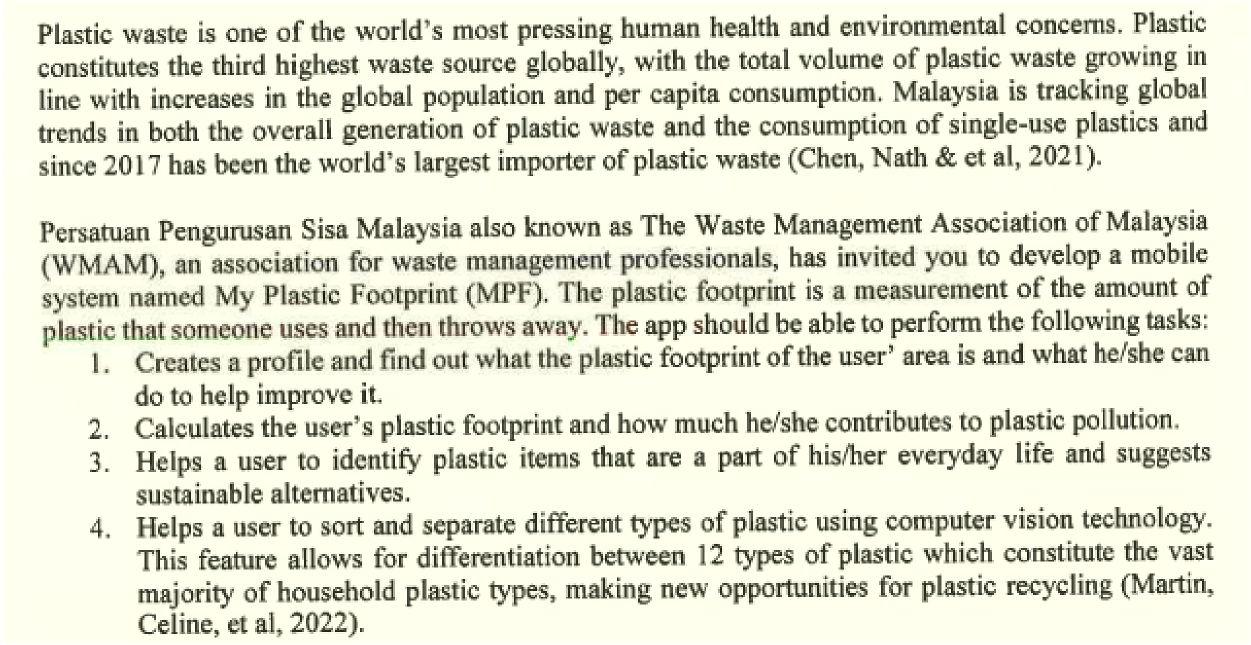
Taking advantage of the eCommerce model focuses on building a mobile application that serves as your digital storefront. And more and more brands are turning to this model due to higher profit margins and the fact that app stores don't charge a commission for selling physical goods through an app (as opposed to virtual goods that are sold as in-app purchases, and are subject to a 30% fee).

This monetization model is ideal for apps whose primary focus is selling physical goods to specific audiences.

If you want to maximize user engagement, you need to leverage push notifications. As users are more likely to see a notification than open an email, you can incite them to buy more by creating a push notification strategy and sending offers specifically tailored to their needs.

https://www.bornfight.com/blog/how-to-make-money-with-your-mobile-app-top-10-app-monetization-models/
http://thinkapps.com/blog/post-launch/monetize-apps-paid-apps-vs-app-purchases-vs-freemium-vs-subscription/

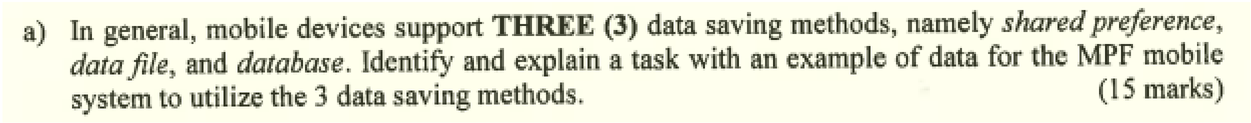
11 May 2022



Plastic waste is one of the world's most pressing human health and environmental concerns. Plastic constitutes the third highest waste source globally, with the total volume of plastic waste growing in line with increases in the global population and per capita consumption. Malaysia is tracking global trends in both the overall generation of plastic waste and the consumption of single-use plastics and since 2017 has been the world's largest importer of plastic waste (Chen, Nath & et al, 2021).

Persatuan Pengurusan Sisa Malaysia also known as The Waste Management Association of Malaysia (WMAM), an association for waste management professionals, has invited you to develop a mobile system named My Plastic Footprint (MPF). The plastic footprint is a measurement of the amount of plastic that someone uses and then throws away. The app should be able to perform the following tasks:

1. Creates a profile and find out what the plastic footprint of the user' area is and what he/she can do to help improve it.
2. Calculates the user's plastic footprint and how much he/she contributes to plastic pollution.
3. Helps a user to identify plastic items that are a part of his/her everyday life and suggests sustainable alternatives.
4. Helps a user to sort and separate different types of plastic using computer vision technology. This feature allows for differentiation between 12 types of plastic which constitute the vast majority of household plastic types, making new opportunities for plastic recycling (Martin, Celine, et al, 2022).



a) In general, mobile devices support **THREE (3)** data saving methods, namely *shared preference*, *data file*, and *database*. Identify and explain a task with an example of data for the MPF mobile system to utilize the 3 data saving methods. (15 marks)

a.

Shared Preferences

Shared preferences are used to store primitive data in key-value pairs. It will persist across the session even though the app is killed. Data stored in shared preferences cannot repeat and must have one instance of record only.

Task 1: Shared preferences can be used to check if the user is logged in, if the user status is logged in, the MPF mobile system will redirect the user to the profile page with the user_id extracted from database to be used within the MPF application; Otherwise, he/she will be redirected to the login page.

Assumption: Since question mentions about profile, we can assume that user need to login before access to this feature

Data Files

Data files are used to read and write a large amount of data. It is suitable for storing images such as png, jpg, file format such as pdf or anything that needs to be exchanged over the network.

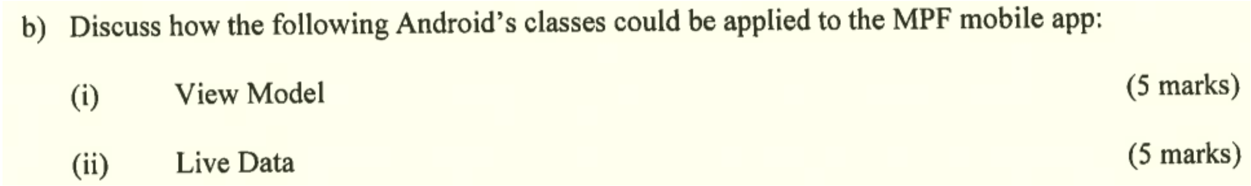
Task 3: After the user captures a photo of a plastic item, the image can be saved into data files and extracted later to be processed by computer vision technology.

Database

Databases such as Firebase, SQLite can be used to store repeated and structured data. It can be used to perform create, read, update, and delete data.

Task 2: For calculation for daily plastic footprint, users have to record everyday plastic waste that was created by himself or herself. Based on the scenario, when a user wants to recycle the plastic food packaging, he takes a photo of the plastic waste and then the system identifies the type of plastic. Besides, users need to fill in the date and time. The filled in data such as date and time and plastic type identified by the system need to be stored in a table within a database in order to let the system calculate the daily plastic footprint of the user.

Task 3: The system has to store the suggestion of a sustainable alternative that is inserted by the admin. Therefore all the suggested data inserted by the admin needs to be stored in the database in order to broadcast to all the users of the system.



b) Discuss how the following Android's classes could be applied to the MPF mobile app:

(i) View Model (5 marks)

(ii) Live Data (5 marks)

b. (i)

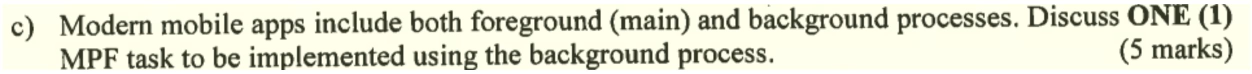
View model is designed to hold and manage UI-related data. View model object is automatically retained from the configuration change so that the data it holds is immediately available to the next activity or fragment instance. If an activity is recreated, it receives the same view model instance that was created by the first activity.

Since data can be destroyed during configuration change, View Model can be used to hold the UI data in order to avoid data missing or change back to the default data. For example, the app retrieving the total plastic waste of the user from the database will change back to the default value (i.e zero) when the user rotates the phone (configuration change) and cause the system to recalculate again which will make the system inefficient. With View Model, the number of total plastic waste will remain even if the user rotates the phone screen.

(ii)

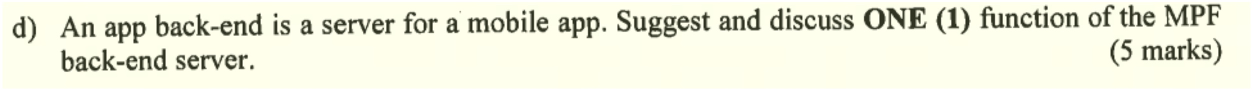
LiveData is an observable data holder class and it is lifecycle-aware. Live Data is observed by an observer. If the data is changed, the observer will be notified and the observer will update the data that displays at the user interface.

For example, data retrieved from a database is best to store in Live Data. When users perform any CRUD functions, the UI data will update automatically.



c) Modern mobile apps include both foreground (main) and background processes. Discuss **ONE (1)** MPF task to be implemented using the background process. (5 marks)

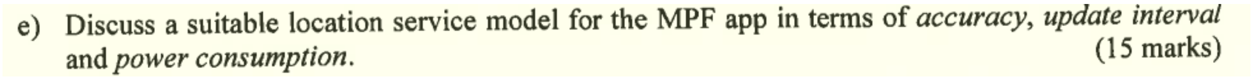
c. One MPF task that will be implemented using the background process is **performing network requests** such as fetching or updating the database. For instance, task 1 requires the app to fetch the profile information of a logged in user from the database. Thus, an Async Task can be used to fetch the data without blocking the main thread. Which is, the process will run on a background thread and the result is published on the UI thread. It ensures the application is always responsive.



d) An app back-end is a server for a mobile app. Suggest and discuss **ONE (1)** function of the MPF back-end server. (5 marks)

Azure Backend as a Service. Azure consists of **personalized push notifications**, which is, it can instantly send push notifications to the Android app. Notification Hubs can be hooked to the backend of the MPF Android app. For example, when the total plastic footprints of the user's area is exceed the threshold, the MPF app sends a push notification to every registered user to alert them so that they have the awareness and start to do waste management.

Social integration. Besides using login credentials such as username and email to login to the app, MPF apps can use social integration which links users with their social media profiles such as Google or Facebook using API. Upon authenticating with these services, the user can log in in a convenient way.



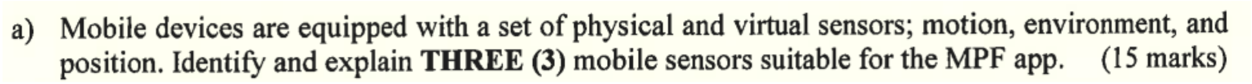
e) Discuss a suitable location service model for the MPF app in terms of *accuracy*, *update interval* and *power consumption*. (15 marks)

Accuracy indicates the precision of the location data.
The accuracy of the location is medium as the app only needs to access the approximate location that the user produces the plastic waste. For example, the user is currently at Jalan Emas, maybe the person who records his plastic waste record is on a bus or car. The location of the user may keep changing for a certain meter or kilometer. Based on this situation, the system only needs to detect if the user is located within the Jalan Emas area. Therefore the accuracy of the location can be medium.

Update Interval refers to how frequently the system needs to update the current user location.
The update interval is low because this app does not need to keep track of the exact place of the user oftenly. It only needs the user location when the user wants to record his or her plastic footprint into the MPF app.

Power consumption is interrelated with the accuracy and update interval of the MPF app. The higher accuracy and update interval, the higher power consumption. For this MPF app, it does not need to update the location of the user frequently, so the power consumption is low.

Question 2

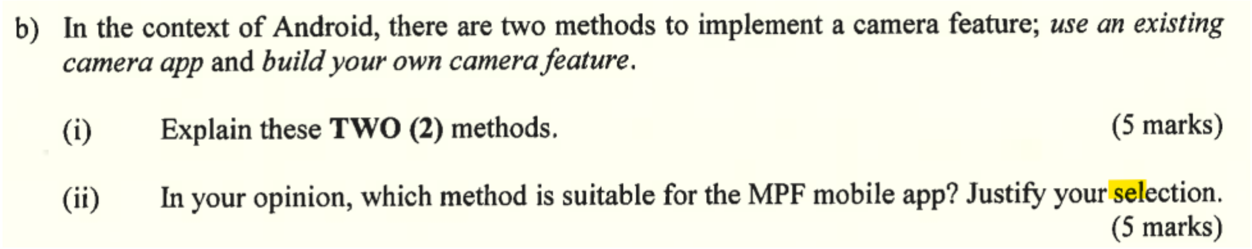


a) Mobile devices are equipped with a set of physical and virtual sensors; motion, environment, and position. Identify and explain **THREE (3)** mobile sensors suitable for the MPF app. (15 marks)

GPS sensor. After setting up the location (i.e Kuala Ulu Langat), the MPF app can generate an analysis result of the plastic footprint based on the user area. To illustrate, when the user performs some activities that might make more plastic footprint such as whether the user takes away the food or buys groceries using plastics, the MPF mobile will record down the footprint along with the location configured by the user. All in all, the analysis result can be generated so that all the nearby users that use the MPF mobile app can view the result.

Ambient Light sensor. It can be configured near the front or back camera to detect the light around the user environment and control the screen brightness accordingly. This is to ensure the photo quality taken by the user is good so that the computer vision technology can detect the plastics item easily and sort them into the appropriate plastic type.

Accelerometers sensor. It enables users to have an upgraded experience such as better viewing experience by adjusting the orientation of the MPF app screen in the smartphone. The device can adapt the orientation as per the device position from horizontal to vertical and vice-versa. For example, when the user rotates the screen into the portrait screen, it allows the user to read and edit their profile information more easily within the MPF app.



b) In the context of Android, there are two methods to implement a camera feature; *use an existing camera app* and *build your own camera feature*.

(i) Explain these **TWO (2)** methods. (5 marks)

(ii) In your opinion, which method is suitable for the MPF mobile app? Justify your selection. (5 marks)

(i)

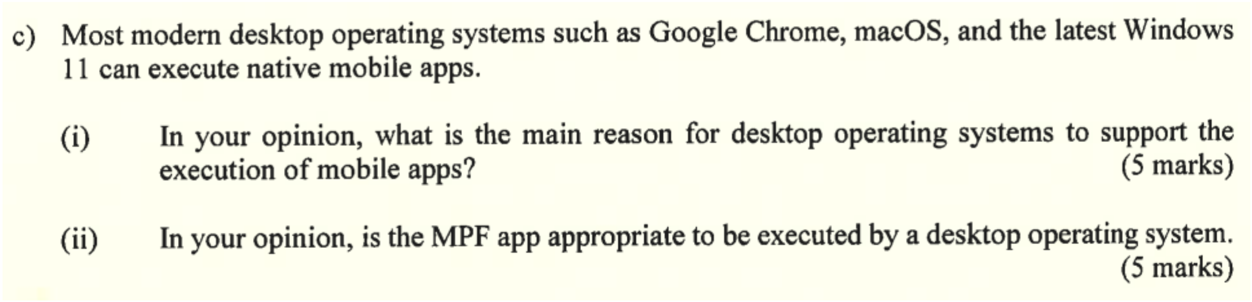
Use an existing camera app

A quick way to take photos in the app with minimal code effort. To illustrate, it uses minimum code to achieve this purpose. Which is, we create an Intent using MediaStore.ACTION_IMAGE_CAPTURE, execute it using startActivity(), and lastly set up an onActivityResult() to receive Intent result.

Build your own camera features

It requires us to write our own code to implement the camera features. For example, we can use Camera X, a Jetpack library to develop a camera app.

(ii)
In my opinion, I think using an existing camera app is suitable for the MPF mobile app because it is the quick way to enable taking pictures or videos in MPF without a lot of extra code. Since the MPF is only interested in snapping a quick picture without any further special processing, an existing camera is good enough to be implemented in the MPF mobile app.



c) Most modern desktop operating systems such as Google Chrome, macOS, and the latest Windows 11 can execute native mobile apps.

(i) In your opinion, what is the main reason for desktop operating systems to support the execution of mobile apps? (5 marks)

(ii) In your opinion, is the MPF app appropriate to be executed by a desktop operating system. (5 marks)

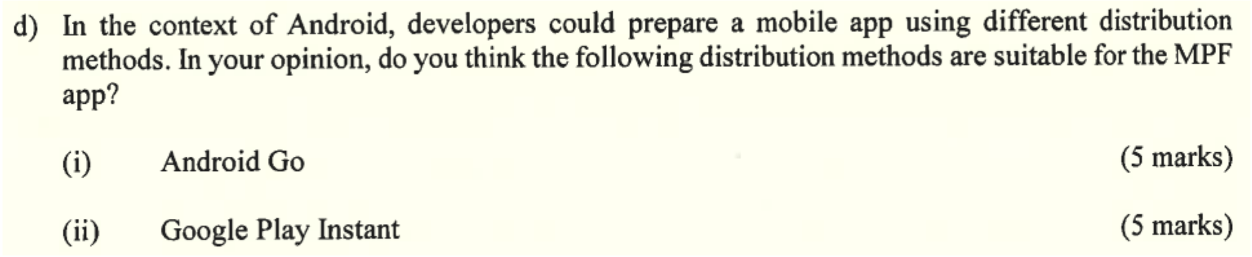
(i)
In short, Android has a presence and popularity. It's just missing that vital pivot in the minds of its users -- the realization that, yes, it can be used as a desktop operating system. It has the apps, it has the games, and it has the familiarity.

Firstly, it is easy for a user who is using a computer to search for a specific platform that provides the function he wants. If the mobile app cannot be supported by a desktop operating system, the user stops continuing to explore the system because the user never has the chance to test the system on the computer and find out how useful the system is. Therefore, mobile apps should be supported by desktop operating systems and this will be very useful to attract the first time user.

Besides that, the mobile app supported by the desktop operating system allows users to feel free to use a computer or mobile phone to use the system. This helps to enhance user experiences since the user does not need to switch to other devices to use the system when he is using a mobile phone or computer.

(ii)
In my opinion, the MPF app is not appropriate to be executed by the desktop operating system. Even though laptops or some computers support camera features, it is impossible for users to bring heavy devices everywhere. This system will be easy for users using mobile phones, users can record the daily plastic waste anywhere anytime. Mobile phones will be the best choice because of their light weight and we can bring our mobile phone anywhere anytime.

Another choice is the MPF app which can be executed by the desktop operating system and can include some basic features such as guidelines to use the system so that users can have a system functionality preview and then download the system on their mobile phone to enjoy the full functionality of the system.



d) In the context of Android, developers could prepare a mobile app using different distribution methods. In your opinion, do you think the following distribution methods are suitable for the MPF app?

(i) Android Go (5 marks)

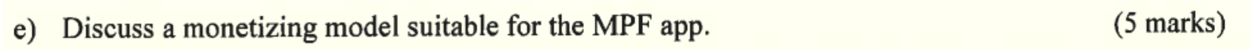
(ii) Google Play Instant (5 marks)

(i) **Android Go**

Yes, The MPF app size is small (<40MB), and the ram usage is below 50MB. Android Go is a stripped-down version of the Android Operating system , designed for low-end smartphones that have less than 2 GB of RAM. It enables users to use less memory, storage space, and mobile data to download and enjoy the MPF app's features.

(ii) **Google Play Instant**

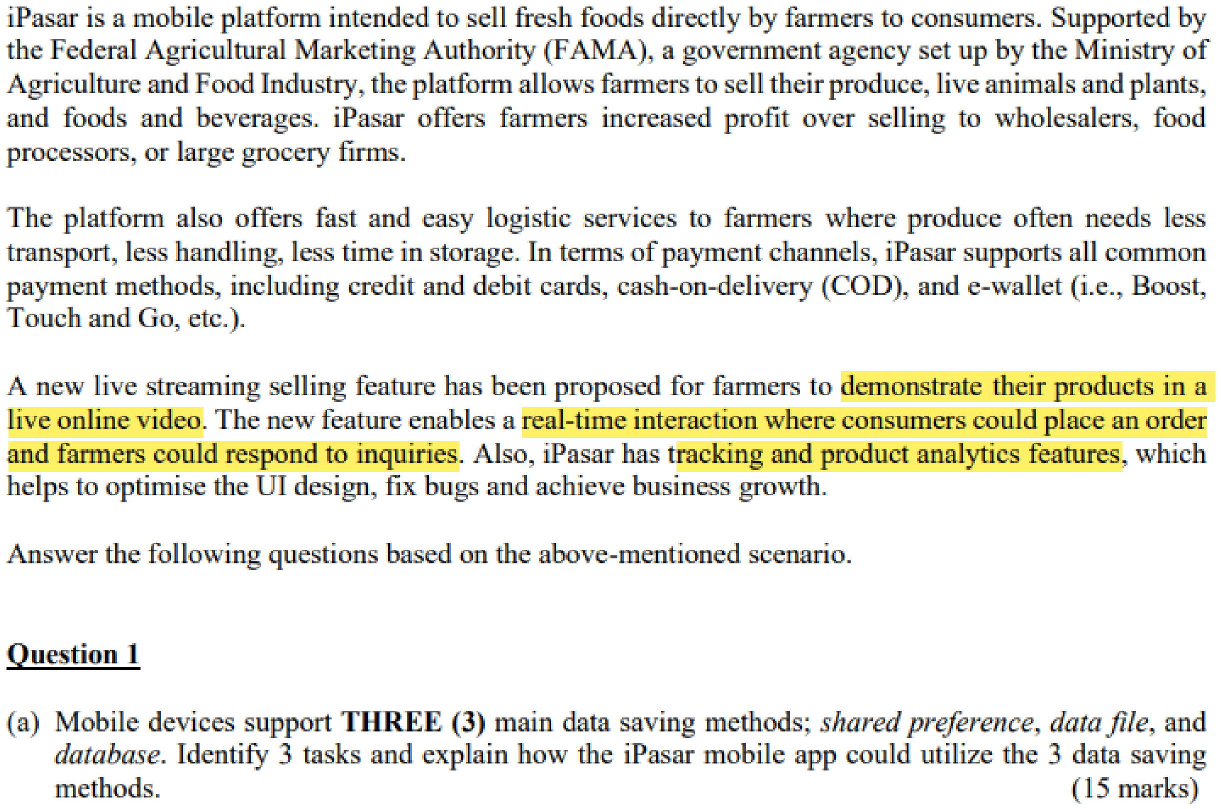
Yes, I think Google Play Instant is suitable for the MPF app. For example, we can have a "Try it Now" button on the store listings to let Android users know what main feature is available. It lets the users launch the MPF apps without being installed to provide instant experience and improve the app discovery to help drive more active users and installations.



e) Discuss a monetizing model suitable for the MPF app. (5 marks)

Rewarded product. Users can download the MPF app for free and earn some reward by performing the task required by the The waste management association of Malaysia. To illustrate, after sharing and contributing the user plastics footprint within the user area, and the app detected that the the contribution to plastic pollution is low, he/she can get free stuff and earn rewards such as shopping e-vouchers, promotion code, etc

Dec 2021



iPasar is a mobile platform intended to sell fresh foods directly by farmers to consumers. Supported by the Federal Agricultural Marketing Authority (FAMA), a government agency set up by the Ministry of Agriculture and Food Industry, the platform allows farmers to sell their produce, live animals and plants, and foods and beverages. iPasar offers farmers increased profit over selling to wholesalers, food processors, or large grocery firms.

The platform also offers fast and easy logistic services to farmers where produce often needs less transport, less handling, less time in storage. In terms of payment channels, iPasar supports all common payment methods, including credit and debit cards, cash-on-delivery (COD), and e-wallet (i.e., Boost, Touch and Go, etc.).

A new live streaming selling feature has been proposed for farmers to demonstrate their products in a live online video. The new feature enables a real-time interaction where consumers could place an order and farmers could respond to inquiries. Also, iPasar has tracking and product analytics features, which helps to optimise the UI design, fix bugs and achieve business growth.

Answer the following questions based on the above-mentioned scenario.

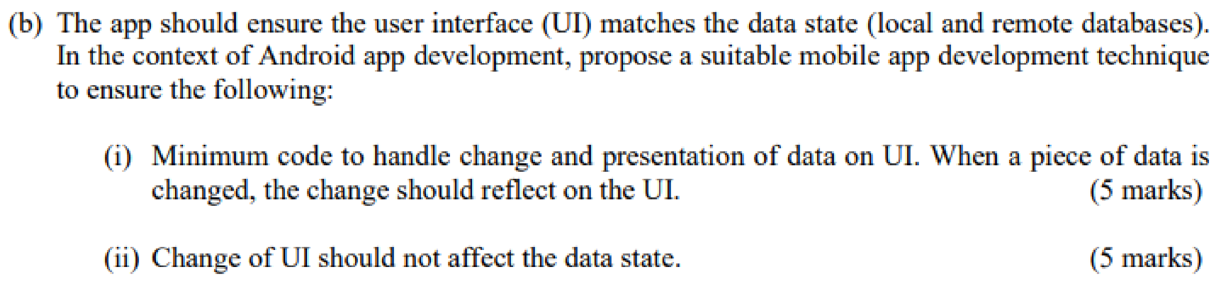
Question 1

(a) Mobile devices support **THREE (3)** main data saving methods; *shared preference*, *data file*, and *database*. Identify 3 tasks and explain how the iPasar mobile app could utilize the 3 data saving methods. (15 marks)

Shared preferences store primitive data in key value pairs. Its data will persist across the session even though an app is killed. The key in shared preferences cannot be repeated and must only have one instance record. For instance, assume that the iPasar requires the user to login before he/she can enjoy the features provided by iPasar. The system will need to store the user login session in shared preference. Afterward, the shared preference can be used to check whether the user is logged in or not; If the user is logged in, the iPasar will redirect the user to the homepage; otherwise, it requires the user to enter the login credentials in order to login or register a new account.

Data files are used for reading and writing a large amount of data. It is suitable for storing image files in format such as png, jpg, file format such as pdf, sound or anything that could be exchanged over the network. It can be stored within the internal storage or external storage. For instance, the real-time interaction, texts between consumer and farmers during the live streaming, which can be considered a large amount of info can be stored in the data files.

Databases such as Firebase, SQLite are used for storing repeated and structured data. It can be used to perform CRUD operations. For instance, after the consumer confirms to buy the product from iPasar, he/she will need to make payment through any channel provided by the iPasar. Afterward, the payment information such as payment date, address, total amount, telephone number, order item will be stored in a table using the database and can be retrieved as a receipt to be displayed to the consumer.



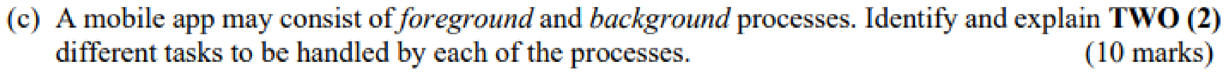
(b) The app should ensure the user interface (UI) matches the data state (local and remote databases). In the context of Android app development, propose a suitable mobile app development technique to ensure the following:

(i) Minimum code to handle change and presentation of data on UI. When a piece of data is changed, the change should reflect on the UI. (5 marks)

(ii) Change of UI should not affect the data state. (5 marks)

(i) Live data is an observer data holder class which is lifecycle-awareness. It means the live data will only update app component observers that are in the active lifecycle state. Live data is observed by an observer. If the data has been changed, the observer will be notified and it updates the UI.

(ii) View model. The change of UI indicates a configuration change. View models hold UI data. The view model object automatically retains during configuration change so that the data they hold is immediately available to the next activity or fragment instance. If the activity is recreated, it receives the same view model instance that is created by the first activity.



(c) A mobile app may consist of *foreground* and *background* processes. Identify and explain **TWO (2)** different tasks to be handled by each of the processes. (10 marks)

A foreground process is one that is required for what the user is currently doing. It refers to the process on the main thread such as user interactions.

Example 1:

User input event. For example, the textOnChanged event on an edit text will be triggered and show an error message to the user if the validation fails.

Example 2:

When the user uploads an image (before stored into the database) from the gallery or existing camera app, it allows the user to preview the image by using setImageURL.

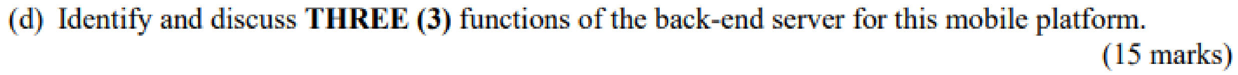
A background process is the execution of long-running operations in different threads (i.e background thread) other than the main thread.

Example 1:

Perform network operations such as fetching data from a database.

Example 2:

Decoding bitmap to display the image within the mobile app.

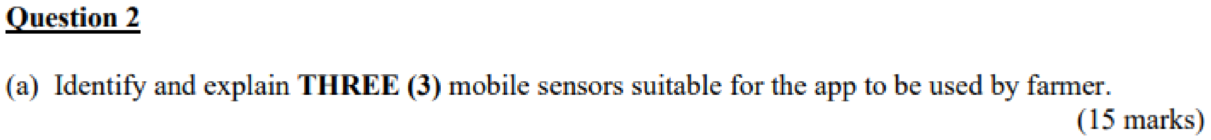


(d) Identify and discuss **THREE (3)** functions of the back-end server for this mobile platform.
(15 marks)

Push notification - Azure allows a personalized push notification. Which is, it can instantly send push notifications to the app. Notification hubs can be hooked into the backend of the iPasar app. For example, if the user successfully places an order and makes payment, a push notification about the successful order message will be displayed to inform him/her.

Storage - Firebase has a powerful, scalable,and cost-effective cloud storage to store user generated content such as video, and photo. The Firebase SDKs for Cloud Storage add Google security to file uploads and downloads regardless of network quality. You can use firebase SDKs to store images, audio, video, or other user-generated content. For instance, after the live streaming video finishes, the recorded video can be uploaded and stored onto the Firebase cloud storage.

Social integration - the iPasar should allow users to link with their social media profiles such as Google or Facebook using API. By having this functionality, it allows users to log in conveniently without entering the login credentials which are username and password.



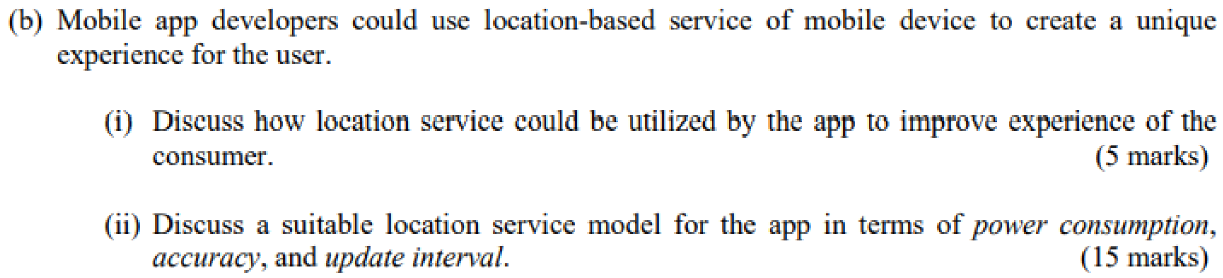
Question 2

(a) Identify and explain **THREE (3)** mobile sensors suitable for the app to be used by farmer.
(15 marks)

Accelerometer sensor. It provides users with an upgraded experience such as better viewing experience by adjusting the orientation of the iPasar app screen in the smartphone. The device can adapt the orientation as per the device position from horizontal to vertical and vice-versa. For example, if the user rotates the phone screen as portrait mode to view the live streaming video, the sensor will automatically adjust the screen and let the user view the video in a comfortable way.

Ambient light sensor. It senses the amount of ambient light present, and appropriately dim the device's screen to match it. This avoids having the screen be too bright when the user's pupils are adapted for vision in a dark room, or too dim when the device is used outdoors in the daytime. It also helps to save battery.

GPS sensor. This sensor can be incorporated into the iPasar app to enable remote position monitoring. For example, customers can track their order product via GPS.



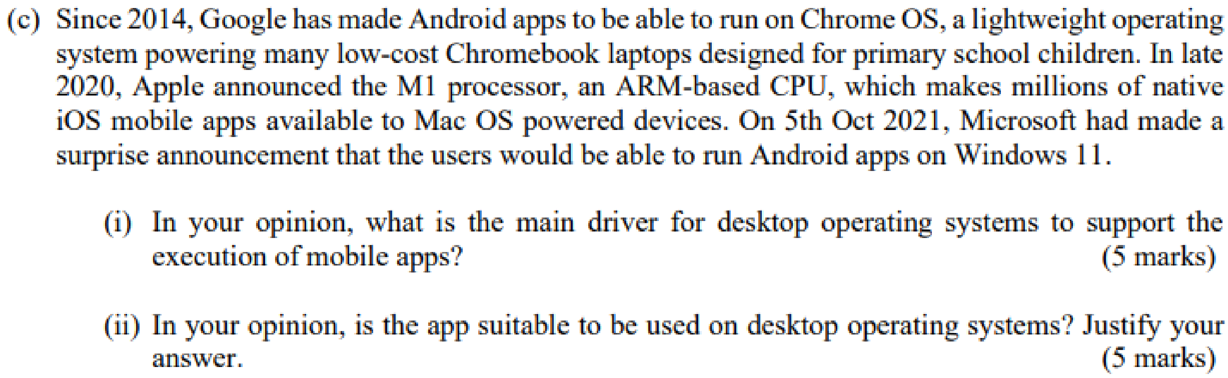
(b) Mobile app developers could use location-based service of mobile device to create a unique experience for the user.

(i) Discuss how location service could be utilized by the app to improve experience of the consumer. (5 marks)

(ii) Discuss a suitable location service model for the app in terms of *power consumption*, *accuracy*, and *update interval*. (15 marks)

(i)
Find and locate the nearby farmer locations, so that they can pick up the product instead of using the delivery option which is faster.

(ii)
Accuracy - precision to the location data - low
Update interval - the frequency of obtaining the current user location - low
Power consumption - related to the accuracy and update interval of the app, high, high - low

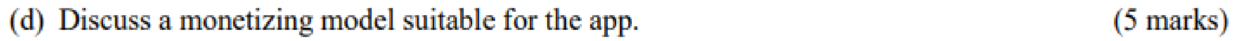


(c) Since 2014, Google has made Android apps to be able to run on Chrome OS, a lightweight operating system powering many low-cost Chromebook laptops designed for primary school children. In late 2020, Apple announced the M1 processor, an ARM-based CPU, which makes millions of native iOS mobile apps available to Mac OS powered devices. On 5th Oct 2021, Microsoft had made a surprise announcement that the users would be able to run Android apps on Windows 11.

(i) In your opinion, what is the main driver for desktop operating systems to support the execution of mobile apps? (5 marks)

(ii) In your opinion, is the app suitable to be used on desktop operating systems? Justify your answer. (5 marks)

Same as may 2022



(d) Discuss a monetizing model suitable for the app. (5 marks)

E-commerce. The iPasar app can be downloaded for free. E-commerce monetization model is suitable for business to consumers (B2C) which combine the technology, logistics and payment solutions. This monetization model is ideal for apps whose primary focus is selling physical goods (live animals and plants, and foods and beverages) to specific audiences. The app will charge sales commission for selling the goods for profit.