BTEC Level 2 IT
Unit 7: Mobile Apps Development
Revision Course Answers
Academic Year 2025–2026

Blue Lakes International School
Kigali - Rwanda, Bugesera, Nyamata
Prepared for: Tr. Kitoga Elisee

October 2025

BTEC Level 2 First in Information Technology
Unit 7: Mobile Apps Development
Revision Course Answers for Mid-Term Examination
October/November 2025

This document provides complete answers to all practice questions in the BTEC Level 2 IT Unit 7 Revision Course, including detailed explanations and a comprehensive report for the EduRwanda case study, tailored to the BTEC assessment criteria.

Prepared for Students of Blue Lakes International School

Contents

1 Characteristics of Native Mobile Apps

1.1 Explanation

Native mobile apps are developed for a specific platform (e.g., Android or iOS) using platform-specific tools and languages, ensuring optimal performance and integration with device features.

1.2 Key Characteristics

- **Platform-Specific Design**: Built for one operating system, leveraging native APIs for seamless integration with device features like GPS or camera.
- **High Performance**: Optimized for the platform, providing fast load times and smooth operation due to direct hardware access.

1.3 Why It Matters

Native apps are ideal for applications requiring high performance and deep device integration, such as educational apps for real-time data processing or offline functionality in rural settings.

1.4 Answers to Practice Questions

1. **State two characteristics of a native mobile app.**
   1. Platform-specific design, tailored to a single operating system like Android or iOS.
   2. High performance due to direct access to device hardware and optimized coding.
2. **Explain why native apps offer high performance.**
   Native apps are coded using platform-specific languages (e.g., Swift for iOS, Kotlin for Android) that interact directly with the device's hardware and operating system, reducing processing delays and enabling faster load times compared to apps relying on web technologies or cross-platform frameworks.
3. **Describe how a native app could benefit a school's mobile application.**
   A native app for a school could provide a fast, reliable interface for accessing resources like timetables or grades, even offline, and integrate with device features like push notifications to alert students about deadlines, enhancing engagement and usability.

2 Types of Mobile Apps

2.1 Explanation

Mobile apps are categorized by their development approach and functionality, each suited to different user needs and contexts.

2.2 Key Types

- **Native Apps**: Designed for a specific platform, offering high performance. *Example*: A school app for iOS to manage assignments.
- **Web Apps**: Run in a browser, requiring no installation. *Example*: An online quiz portal accessible via a mobile browser.

2.3 Why It Matters

Choosing the right app type ensures the application meets user needs, budget constraints, and functionality requirements, critical for educational contexts.

2.4 Answers to Practice Questions

1. **Name two types of mobile apps.**
   1. Native apps.
   2. Web apps.
2. **Provide an example of a web app used in education.**
   An online learning portal where students access course materials and quizzes through a mobile browser, such as a virtual classroom platform.
3. **Explain one difference between a native app and a web app.**
   A native app is installed on the device and uses platform-specific code for high performance, while a web app runs in a browser, requiring internet access and offering limited device feature integration.

3 Hybrid Mobile Apps

3.1 Explanation

Hybrid apps combine native and web app features, using web technologies (HTML, CSS, JavaScript) within a native container for cross-platform compatibility.

3.2 What is a Hybrid App?

- **Definition**: Built with web technologies but packaged as a native app, enabling deployment on multiple platforms.
- **Cost-Effectiveness**: Uses a single codebase, reducing development time and costs compared to separate native apps for Android and iOS.

3.3 Why It Matters

Hybrid apps are cost-effective for organizations like schools, allowing broad reach across devices without the high costs of native development.

3.4 Answers to Practice Questions

1. **Define a hybrid mobile app.**
   A hybrid mobile app is developed using web technologies (HTML, CSS, JavaScript) and packaged in a native container, allowing it to run on multiple platforms like Android and iOS with a single codebase.
2. **Explain why hybrid apps are cost-effective for development.**
   Hybrid apps use one codebase for multiple platforms, reducing development time and costs compared to creating separate native apps for Android and iOS, which require distinct coding and testing processes.
3. **Describe a scenario where a school might choose a hybrid app over a native app.**
   A school with a limited budget might choose a hybrid app for a library system to allow both Android and iOS users to access book catalogs, as it reduces development costs while maintaining basic functionality across platforms.

4 Contexts of Mobile App Use

4.1 Explanation

Mobile apps serve various contexts, each requiring tailored functionality to meet user needs in specific environments.

4.2 Key Contexts

- **Education**: Supports learning and administration. *Example*: A library app for reserving books.
- **Productivity**: Enhances efficiency. *Example*: A task management app for tracking assignments.

4.3 Why It Matters

Understanding contexts ensures apps are designed to address specific user requirements, such as accessibility for students or efficiency for teachers.

4.4 Answers to Practice Questions

1. **Describe one context in which mobile apps are used, with an example.**
   Education: A library app allows students to search and reserve books, supporting academic research and resource access.
2. **Explain how a productivity app can benefit students in a school.**
   A productivity app, like a task manager, helps students organize assignments and deadlines, improving time management and reducing missed submissions.
3. **Name an educational app and describe its purpose.**
   A timetable app helps students view class schedules and receive reminders, ensuring they stay organized and punctual.

5 User Needs in App Design

5.1 Explanation

Designing apps requires addressing user needs to ensure usability and adoption, especially in educational settings.

5.2 Example User Need

- **Ease of Use**: Users need intuitive interfaces to navigate apps without extensive training.

5.3 Influence on Design

- **Simple Navigation**: Clear menus and large buttons enhance accessibility.
- **Clear Instructions**: Tooltips or guidance improve user understanding.

5.4 Why It Matters

Meeting user needs ensures apps are functional and accessible, increasing adoption in schools.

5.5 Answers to Practice Questions

1. **State one user need for a school library app.**
   Ease of use, ensuring students can navigate the app without technical expertise.
2. **Explain how ease of use influences the design of a mobile app.**
   Ease of use leads to simple, intuitive interfaces with clear menus and minimal steps, making the app accessible to users with varying technical skills, such as young students or non-tech-savvy teachers.
3. **Describe how a school can ensure a library app is accessible to all students.**
   The school can implement large, tappable buttons, high-contrast text, and screen reader compatibility to support students with visual impairments or limited technical skills.

6 Constraint of Limited Screen Size

6.1 Explanation

Mobile devices have smaller screens than desktops, requiring design adjustments for usability.

6.2 Impact on Design

- **Simplified Layouts**: Use single-column designs and larger text/buttons for readability on small screens, e.g., a library app with a clear book search interface.

6.3 Why It Matters

Adapting to screen size ensures apps are user-friendly across devices, critical for diverse school users.

6.4 Answers to Practice Questions

1. **Explain one way limited screen size affects mobile app design.**
   Limited screen size requires simplified layouts, such as single-column designs, to avoid clutter and ensure content is easily readable on smaller displays.
2. **Describe how a simplified layout benefits a mobile app user.**
   A simplified layout reduces visual clutter, making it easier for users to find and interact with key features, like searching for books in a library app, improving efficiency and reducing frustration.
3. **Suggest one design feature to improve usability on a small screen.**
   Use large, tappable buttons (at least 44x44 pixels) to ensure easy interaction, especially for users with limited dexterity.

7 GPS Functionality in Mobile Apps

7.1 Explanation

GPS integration enables location-based services, enhancing app functionality.

7.2 Example and Benefits

- **Example**: A navigation app for school field trips uses GPS for real-time directions.
- **Enhancement**: Provides location tracking and context-aware services, improving user experience.

7.3 Why It Matters

GPS is vital for apps requiring location data, such as attendance tracking in rural areas.

7.4 Answers to Practice Questions

1. **Name one mobile app that uses GPS.**
   A navigation app for school field trips.
2. **Explain how GPS improves the functionality of a navigation app.**
   GPS tracks the user's location in real time, providing accurate directions and route updates, ensuring users reach their destination efficiently.
3. **Describe a scenario where GPS could be useful in a school app.**
   A school app could use GPS to track student attendance at off-site events, ensuring accurate records in rural areas with limited internet.

8 Other Device Functions in Mobile Apps

8.1 Explanation

Apps leverage device features like cameras or accelerometers to enhance functionality.

8.2 Key Device Functions

- **Camera**: Scans or captures images. *Example*: A study app digitizing handwritten notes.
- **Accelerometer**: Detects motion. *Example*: A fitness app tracking student activity.

8.3 Why It Matters

Device functions create interactive apps, improving engagement in educational contexts.

8.4 Answers to Practice Questions

1. **Name one device function (other than GPS) used in mobile apps.**
   Camera.
2. **Provide an example of an app that uses the camera and describe its purpose.**
   A study app that scans handwritten notes to digitize them, allowing students to store and share notes easily.
3. **Explain how an accelerometer can enhance a mobile app's functionality.**
   An accelerometer detects device motion, enabling a fitness app to track steps or activity, encouraging students to stay active by monitoring their progress.

9 Android vs. iOS in App Development

9.1 Explanation

Android and iOS differ in development tools, market reach, and processes, impacting app development decisions.

9.2 Key Differences

- **Development Tools**: Android uses Android Studio (Java/Kotlin); iOS uses Xcode (Swift/Objective-C).
- **Market Reach**: Android has a larger global share; iOS dominates premium markets.
- **Fragmentation**: Android supports diverse devices, requiring more testing; iOS has a controlled ecosystem.
- **Publishing Process**: iOS has stricter app store guidelines than Android.

9.3 Why It Matters

These differences guide platform choice based on audience and resources.

9.4 Answers to Practice Questions

1. **Explain one difference between Android and iOS in app development.**
   Android uses Android Studio with Java or Kotlin, while iOS uses Xcode with Swift or Objective-C, affecting the tools and skills needed for development.
2. **Describe how Android's fragmentation affects app development.**
   Android's support for diverse devices requires developers to test apps on multiple screen sizes and hardware, increasing development time and complexity compared to iOS's standardized ecosystem.
3. **Suggest one reason a developer might choose iOS over Android for a school app.**
   A developer might choose iOS for a school app targeting a wealthier student demographic, as iOS users are often in premium markets, ensuring higher engagement with a consistent user experience.

10 Device Permissions and User Trust

10.1 Explanation

Device permissions allow apps to access features or data, impacting user trust.

10.2 Implications for User Trust

- **Privacy Concerns**: Access to sensitive data (e.g., contacts) may raise fears of misuse. Clear communication about data use builds trust.

10.3 Why It Matters

Balancing functionality with trust is critical for app adoption, especially in schools.

10.4 Answers to Practice Questions

1. **Define a device permission in the context of mobile apps.**
   A device permission is a setting allowing an app to access specific device features or data, such as location or contacts.
2. **Explain one way requiring contact access affects user trust.**
   Requiring contact access may make users wary, fearing their personal data could be shared or misused, potentially reducing app adoption unless justified clearly.
3. **Describe how a school app can build trust when requesting permissions.**
   A school app can build trust by displaying a clear pop-up explaining why permissions (e.g., camera for scanning) are needed and ensuring data is securely stored, complying with privacy laws.

11 Programming Languages for Mobile Apps

11.1 Explanation

Different languages are used for mobile app development based on the platform.

11.2 Key Languages

- **Kotlin**: Used for Android, modern and interoperable with Java.
- **Swift**: Used for iOS, fast and designed for Apple's ecosystem.

11.3 Why It Matters

Choosing the right language ensures compatibility and performance.

11.4 Answers to Practice Questions

1. **Name one programming language used for Android app development.**
   Kotlin.
2. **Name one programming language used for iOS app development.**
   Swift.
3. **Explain why Swift is suitable for iOS app development.**
   Swift is fast, safe, and optimized for Apple's ecosystem, allowing seamless integration with iOS features like notifications and ensuring high performance.

12 Cross-Platform Development Tools

12.1 Explanation

Cross-platform tools enable apps to run on multiple platforms with one codebase.

12.2 Advantages and Disadvantages

- **Advantage**: Reduced development time and cost using a single codebase.
- **Disadvantage**: Limited access to native features, potentially reducing performance.

12.3 Why It Matters

Cross-platform tools are cost-effective but may compromise on performance.

12.4 Answers to Practice Questions

1. **State one advantage of cross-platform development tools.**
   Reduced development time and cost by using a single codebase for multiple platforms.
2. **State one disadvantage of cross-platform development tools.**
   Limited access to native device features, reducing performance compared to native apps.
3. **Explain when a school might choose a cross-platform tool for app development.**
   A school might choose a cross-platform tool for a simple app, like a news app, to save costs and reach both Android and iOS users with a limited budget.

13 Development Environments

13.1 Explanation

Development environments provide tools for coding, testing, and deploying apps.

13.2 Example Environment

- **Android Studio**: Offers emulators, code editors, and debugging tools for Android apps.

13.3 Why It's Appropriate

- **Comprehensive Tools**: Supports coding, testing, and deployment.
- **Integration**: Provides Android-specific libraries and emulators.

13.4 Answers to Practice Questions

1. **Name one development environment for mobile apps.**
   Android Studio.
2. **Explain why Android Studio is suitable for Android app development.**
   Android Studio provides tools like emulators and debugging features, enabling developers to test and optimize apps for various Android devices efficiently.
3. **Describe one feature of a development environment that aids app creation.**
   An emulator allows developers to test apps on virtual devices, ensuring compatibility across different screen sizes and hardware without physical devices.

14 Touch Screen Influence on UI Design

14.1 Explanation

Touch screens require designs optimized for touch-based interaction.

14.2 Key Influences

- **Large, Tappable Elements**: Buttons/icons sized 44x44 pixels or larger for easy tapping.
- **Gesture Support**: Swipes or taps enhance interactivity, e.g., swiping to navigate menus.

14.3 Why It Matters

Touch-friendly designs improve usability for school users on mobile devices.

14.4 Answers to Practice Questions

1. **Explain one way touch screens influence mobile app UI design.**
   Touch screens require large, tappable buttons to ensure users can interact easily without accidental taps, especially on small screens.
2. **Describe how large buttons improve a mobile app's usability.**
   Large buttons make it easier for users, including those with limited dexterity, to tap accurately, reducing errors and improving navigation speed.
3. **Suggest one gesture that could be used in a school app's interface.**
   A swipe gesture could allow students to switch between app sections, like moving from a timetable to a grades page, enhancing navigation efficiency.

15 Push Notifications

15.1 Explanation

Push notifications are messages sent to users when the app is not open.

15.2 Benefits in a School Context

- **Timely Updates**: Notify students of deadlines or events.
- **Encourage Interaction**: Prompt users to engage with tasks.

15.3 Why It Matters

Notifications enhance communication and engagement in educational apps.

15.4 Answers to Practice Questions

1. **Describe one benefit of push notifications in a school app.**
   Push notifications alert students to assignment deadlines, ensuring timely submissions and increasing engagement.
2. **Explain how push notifications can increase student engagement.**
   By sending reminders for events or tasks, push notifications prompt students to open the app regularly, fostering consistent interaction with school resources.
3. **Provide an example of a push notification for a school productivity app.**
   “Your math assignment is due tomorrow at 5 PM. Submit now!” sent by a task management app to remind students of deadlines.

16 Native vs. Web Apps for Attendance Tracking

16.1 Explanation

Choosing between native and web apps is critical for projects like attendance tracking, balancing performance, functionality, and cost.

16.2 Case Study: EduRwanda

EduRwanda, a Rwandan NGO, delivers training programs in rural areas. Its current paper-based attendance system is slow, error-prone, and difficult to analyze. A mobile app is needed to streamline tracking and reporting.

16.3 Report: Evaluation of Native and Web Apps for EduRwanda

16.3.1 Challenges of Current System

- **Slow Processing**: Manual entry of attendance data is time-consuming, delaying reporting.
- **Error-Prone**: Handwritten records lead to mistakes, such as incorrect names or dates.
- **Lack of Analysis**: Paper records are hard to compile for reporting, hindering program evaluation.
- **Limited Accessibility**: Records are not easily accessible in rural areas, complicating coordination.

16.3.2 Evaluation of Native and Web Apps

- **Performance**:
  - *Native Apps*: Offer faster performance due to direct hardware access, enabling quick data entry and processing, ideal for real-time attendance tracking.
  - *Web Apps*: Slower, as they rely on browser performance and internet connectivity, which can delay data entry in rural settings.
- **Device Integration**:
  - *Native Apps*: Seamlessly integrate with device features like cameras for QR code scanning or offline storage, critical for rural areas with unreliable internet.
  - *Web Apps*: Limited access to native features, requiring constant internet for full functionality, which is impractical in remote locations.
- **Development Cost**:
  - *Native Apps*: Higher costs due to separate development for Android and iOS, requiring distinct codebases and testing.
  - *Web Apps*: Lower costs, using a single codebase accessible across devices via browsers, reducing development expenses.

- **Usability**:
  - *Native Apps*: Provide intuitive, touch-optimized interfaces tailored to the platform, enhancing user experience for trainers in rural settings.
  - *Web Apps*: May have less responsive interfaces, especially on mobile browsers, potentially frustrating users with limited technical skills.

16.3.3 Recommendation

A **native app** is recommended for EduRwanda due to its superior performance, offline capabilities, and device integration, which are critical for rural areas with unreliable internet. Despite higher development costs, the reliability and functionality outweigh the drawbacks, ensuring efficient attendance tracking and reporting.

16.3.4 Benefits for EduRwanda

- **Error Reduction**: Digital entry via a native app minimizes mistakes, improving data accuracy by an estimated 90
- **Offline Functionality**: Offline storage allows trainers to record attendance without internet, syncing data later, crucial for rural settings.
- **Device Integration**: Camera-based QR code scanning speeds up attendance tracking, reducing session time by 20
- **Real-Time Reporting**: Native apps enable instant data compilation, facilitating quick program analysis and stakeholder reporting.

16.3.5 Design Considerations

  - **User Interface**: Use large, tappable buttons and a single-column layout to accommodate limited screen sizes and ensure ease of use for trainers.
  - **Permissions**: Request camera and storage permissions with clear explanations to build user trust, e.g., “Camera access is needed for QR code scanning.”
  - **Push Notifications**: Send reminders for attendance submission deadlines to enhance engagement.
  - **Accessibility**: Include high-contrast text and screen reader support to accommodate users with disabilities.

16.4 Answers to Practice Questions

1. **Describe one challenge EduRwanda faces with its paper-based attendance system.**
   The paper-based system is error-prone, as handwritten records often contain mistakes like incorrect names or dates, reducing data reliability.

2. **Compare the performance of native and web apps for attendance tracking.**
   Native apps offer faster performance with direct hardware access, enabling quick data entry, while web apps are slower, relying on internet and browser performance, which can delay tracking in rural areas.
3. **Explain why a native app is suitable for EduRwanda's needs.**
   A native app supports offline functionality and camera integration for QR code scanning, critical for reliable attendance tracking in rural areas with limited internet connectivity.
4. **Suggest one way a native app can improve data accuracy for EduRwanda.**
   A native app can use QR code scanning to automatically record attendance, reducing manual entry errors and ensuring accurate data capture.

17 Conclusion

This revision course provides comprehensive answers to all practice questions for the BTEC Level 2 IT Unit 7 Mid-Term Examination, covering characteristics of native apps, types of apps, hybrid apps, contexts of use, user needs, screen size constraints, GPS and other device functions, Android vs. iOS, permissions, programming languages, cross-platform tools, development environments, touch screen design, push notifications, and a detailed case study on EduRwanda's attendance tracking app. The report recommends a native app for EduRwanda due to its performance, offline capabilities, and device integration, addressing the organization's challenges with a paper-based system. By studying these answers, students will be well-prepared for short-answer, descriptive, and report-style questions, ensuring success in the exam.