Qualification specification

NCFE Level 3 Certificate in Mobile Application Development
QN: 610/0568/X

Qualification summary

Qualification title	**NCFE Level 3 Certificate in Mobile Application Development**		
Ofqual qualification number (QN)	610/0568/X	**Aim reference**	6100568X
Guided learning hours (GLH)	185	**Total qualification time (TQT)**	260
Minimum age	16		
UCAS	This qualification has been allocated UCAS points. Please refer to the UCAS website for further details of the points allocation and the most up-to-date information.		
Qualification purpose	This qualification is designed for learners who want to increase their knowledge, skills and understanding of mobile app development. It will also support progression into relevant employment and further study in mobile app development.		
Grading	Achieved/not yet achieved.		
Assessment method	Internally assessed and externally quality assured portfolio of evidence.		
Work/industry placement experience	Work/industry placement experience is not required.		

Contents

Summary of changes

This document summarises the changes to this qualification specification since the last version (Version 1.0 July 2022). Please check the NCFE website for the most recent version.

Version	Publication date	Summary of amendments
v1.0	July 2022	First publication
v1.1	July 2023	Information regarding UCAS added to About this qualification, Qualification Summary.

Section 1: introduction

If you are using this qualification specification for planning purposes, please make sure that you are using the most recent version.

Aims and objectives

This qualification aims to:

- focus on the study of mobile app development in the digital sector
- offer breadth and depth of study, incorporating a key core of knowledge
- provide opportunities to acquire a number of practical and technical skills

The objective of this qualification is to:

- provide the learner with the knowledge and skills of mobile app development

Support handbook

This qualification specification must be used alongside the mandatory support handbook, which can be found on the NCFE website. This contains additional supporting information to help with planning, delivery and assessment.

This qualification specification contains all the qualification-specific information you will need that is not covered in the support handbook.

Entry guidance

This qualification is designed for learners who want to increase their knowledge, skills and understanding of mobile app development.

It may also be useful to learners studying qualifications in the following areas:

- programming
- software development
- web development

Registration is at the discretion of the centre, in accordance with equality legislation, and should be made on the Portal.

There are no specific prior skills/knowledge a learner must have for this qualification. However, learners may find it helpful if they have already achieved a level 2 digital or information technology qualification.

Centres are responsible for ensuring that all learners are capable of achieving the learning outcomes and complying with the relevant literacy, numeracy and health and safety requirements.

Learners registered on this qualification should not undertake another qualification at the same level, or with the same/a similar title, as duplication of learning may affect funding eligibility.

Achieving this qualification

To be awarded this qualification, learners are required to successfully achieve **6** mandatory units.

Please refer to the list of units in appendix A or the unit summaries in section 2 for further information.

To achieve this qualification, learners must successfully demonstrate their achievement of all learning outcomes of the units as detailed in this qualification specification. A partial certificate may be requested for learners who do not achieve their full qualification but have achieved at least one whole unit.

Progression

Learners who achieve this qualification could progress to the following:

- employment:
  - mobile app developer
  - web developer
- further education:
  - level 4 qualifications in IT
  - level 4 qualifications in programming

Progression to higher level studies

Level 3 qualifications aim to facilitate progression to higher level study, which requires knowledge and skills different from those gained at levels 1 and 2. Level 3 qualifications enable learners to:

- apply factual, procedural and theoretical subject knowledge
- use relevant knowledge and methods to address complex, non-routine problems
- interpret and evaluate relevant information and ideas
- understand the nature of the area of study or work
- demonstrate an awareness of different perspectives and approaches
- identify, select and use appropriate cognitive and practical skills
- use appropriate research to inform actions
- review and evaluate the effectiveness of their own methods

Resource requirements

To assist in the delivery of this qualification, centres and learners should have access to the following mandatory resources to cover all the appropriate learning outcomes:

- a digital device used for app development
- internet connectivity
- web browser software/applications
- programming software

Real work environment (RWE) requirement/recommendation

Experience in the real work environment is not required.

Work/industry placement experience

Work/industry placement experience is not required.

How the qualification is assessed

Assessment is the process of measuring a learner's skill, knowledge and understanding against the standards set in a qualification.

This qualification is internally assessed and externally quality assured.

The assessment consists of **one** component:

- an internally assessed portfolio of evidence, which is assessed by centre staff and externally quality assured by NCFE (internal quality assurance (IQA) must still be completed by the centre as usual)

Learners must be successful in this component to gain the Level 3 Certificate in Mobile Application Development.

All the evidence generated by the learner will be assessed against the standards expected of a level 3 learner for each learning outcome.

Unless stated otherwise in this qualification specification, all learners taking this qualification must be assessed in English and all assessment evidence presented for external quality assurance must be in English.

Internal assessment

Each learner must create a portfolio of evidence generated from appropriate assessment tasks, which demonstrates achievement of all the learning outcomes associated with each unit. The assessment tasks should allow the learner to respond to a real-life situation that they may face when in employment. On completion of each unit, learners must declare that the work produced is their own and the assessor must countersign this. Examples of suitable evidence for the portfolio for each unit are provided in section 2.

A centre may choose to create their own internal assessment tasks. There are 4 essential elements in the production of successful centre-based assessment tasks.

These are:

- ensuring the assessment tasks are meaningful with clear, assessable outcomes
- appropriate coverage of the content, learning outcomes, or assessment criteria
- having a valid and engaging context or scenario
- including sufficient opportunities for stretch and challenge for higher attainers

Please see the guidance document for creation of internal assessment tasks on our website.

Assessment guidance is provided for each unit. Assessors can use other methods of assessment as long as they are valid and reliable and maintain the integrity of the assessment and of the standards required of this qualification.

Section 2: unit content and assessment guidance

This section provides details of the structure and content of this qualification.

The types of evidence listed are for guidance purposes only. Within learners' portfolios, other types of evidence are acceptable if all learning outcomes are covered and if the evidence generated can be internally and externally quality assured. For approval of methods of internal assessment other than portfolio building, please contact your external quality assurer.

The explanation of terms explains how the terms used in the unit content are applied to this qualification. This document can be found in section 3.

Unit 01 Introduction to mobile app development (J/650/1676)

Unit summary			
This unit aims to introduce the learner to the broad concepts, tools and approaches in mobile app development. Learners will also apply their knowledge to build a simple mobile app.			
Assessment			
This unit is internally assessed, via a portfolio of evidence.			
Mandatory	**Achieved/not yet achieved**	**Level 3**	**30 GLH**

Learning outcomes The learner will:	**Assessment criteria** The learner can:
1. Know about mobile apps	1.1 Explain what is meant by a mobile app
	1.2 Describe the standard **types of mobile apps** and their differences
2. Understand different mobile app development environments	2.1 Identify the requirements for iOS mobile app development
	2.2 Identify the requirements for Android mobile app development
	2.3 Identify the **hardware requirements** for mobile app development
	2.4 Explain the **similarities and differences** between iOS and Android in mobile app development
	2.5 Give examples of issues that may occur with **cross-platform** mobile app development
	2.6 Describe the purpose of a **software development kit (SDK)** in mobile app development
	2.7 Compare the differences between the **languages** used for native and hybrid app development
3. Be able to implement the components of a simple mobile app	3.1 Identify installation of environments
	3.2 Create a simple mobile app
	3.3 Explain how **emulators** are used for testing the app
	3.4 Describe how to transfer a mobile app to a hardware **device**
4. Understand the deployment process for iOS and Android mobile apps	4.1 Compare the iOS and Android app store review and release processes
	4.2 Identify the need for build signing on both platforms
	4.3 Identify the key stages of the submission process for both platforms
5. Understand the different app store listing requirements	5.1 Compare requirements for deploying iOS and Android apps into the store application
	5.2 Discuss marketing content for an app store listing in relation to: • listing title • description • category • keywords • launch icon (logo) • marketing screenshots

Range
1. Know about mobile apps
1.2 Types of mobile apps must include: • native • web • hybrid
2. Understand different mobile app development environments
2.3 Hardware requirements should cover the range of equipment utilised when developing and testing mobile apps. **2.4 Similarities and differences** could include but not limited to: • both highly popular platforms • both controlled by single large global tech company • both demonstrate rapid evolution • healthy app ecosystems • large community of active developers • Android software stack available for wide number of devices • Android development is not platform dependent • native development programming languages differ for both platforms • Apple review process takes significantly longer to access their app store • Apple devices far more tied to operating system (OS) updates **2.5 Cross-platform** issues could include but not limited to: • double the development work • different set of programming skills, knowledge, and experience • often inconsistent end-user experience • app store process varies • media assets different formats • range of hardware and software requirements • complicates debugging and testing **2.6 SDK** contain but not limited to: • collection of software development tools in one installable package • support the production of software applications • built-in compiler • built-in debugger • application programming interface (API) support • third-party plug-in support • often specific to a software framework • often specific to a hardware platform • often specific to an operating system

Range
2.7 Languages must include: • iOS development • Android development • hybrid development • web development (for mobile apps)
3. Be able to implement the components of a simple mobile app
3.3 Emulators are software models of physical devices. **3.4 Device** can be mobile phone, tablet, wearable, automotive, and laptop/desktop PC.

Delivery and assessment guidance
At this stage, a platform-independent hybrid approach (for example, Xamarin, Flutter, Cordova) may be the simplest introduction. Equally, if already installed in the centre and planned for main development environment throughout the course, both Android Studio and XCode have simple-to-build mobile apps (for example, 'Hello World' apps), which demonstrate and simulate basic mobile app functionality. **3.1** The learner should have some awareness of the centre's chosen mobile app device and choice of platform and tools, particularly after awareness gained from learning outcome (LO) 2. **3.2** In the centre's initial chosen platform for mobile app device, there will be a simple mobile app boilerplate (for example, a 'Hello World' app) that should require no more than a line or two to build and demonstrate basic functionality to learners. This should include practical demonstration of transfer. **5.1** The learner will gain an understanding that requirements can come from many sources, and that there are various techniques used for extracting them and tools and processes for documenting and enhancing them. The learner must include different types of requirements (for example, business, functional, non-functional) and roles and responsibilities during requirements capture.

Unit 02 Programming fundamentals for mobile app development (K/650/1677)

Unit summary			
In this unit, the learner will know about programming languages and learners will explore, write and debug a programme for mobile app development.			
Assessment			
This unit is internally assessed, via a portfolio of evidence.			
Mandatory	**Achieved/not yet achieved**	**Level 3**	**35 GLH**

Learning outcomes The learner will:	**Assessment criteria** The learner can:
1. Know about programming languages	1.1 Identify what is meant by programming
	1.2 Identify what is meant by a programming language
	1.3 State differences between procedural and object-oriented languages
2. Understand key mobile app development programming concepts	2.1 Demonstrate common mobile app development **programming concepts**
	2.2 Demonstrate the following object-orientated programming concepts: • objects • fields • interfaces • classes • properties
	2.3 Explain the **main principles** of object-orientated programming in relation to: • encapsulation • abstraction • inheritance • polymorphism • classes
3. Be able to use an algorithm to resolve a problem	3.1 Create a flowchart to resolve a problem
	3.2 Develop pseudocode to meet the created flowchart
4. Be able to debug and write code	4.1 Explain the importance of debugging code
	4.2 Show how to **write and debug interpreted code**
	4.3 Show how to **debug compiled code**
	4.4 Identify issues that can occur cross-platform
	4.5 Implement a coded solution
	4.6 Explain the differences between compiled and interpreted coding solutions

Range
2. Understand key mobile app development programming concepts
2.1 The common **programming concepts** should be appropriate to the centre's chosen device environment/platform. For example: If working in Swift (iOS), these should include but not limited to: • variables/constants • strings • collections/arrays • control flow (loops) • functions • closures • tuples • optionals • objects (enums, structs, classes) • protocols • extensions • data types If working with Kotlin (Android), these should include but not limited to: • basic types (numbers, operations, comparisons, integers, literals, booleans, characters, strings) • arrays • type checks and casts • control flow (conditions and loops) • returns and jumps • exceptions • packages and imports • classes and objects (including, but not limited to, inheritance, properties, interfaces, modifiers and extensions) • functions (including, but not limited to, lambdas, inline functions, operator overloading, builders) • null safety • equality • this expressions • asynchronous programming techniques • coroutines • annotations • destructuring declarations • reflection If working in another programming language (for example, JavaScript, Java, Python) adjust the above lists accordingly and in equivalent demonstrable depth. **2.3** The **main principles** include: • object-oriented design (OOD) • the concept of instantiation • loose and tight coupling

Range
4. Be able to debug and write code
4.2 The learner will show how to **write and debug interpreted code**, considering: • how the code is written and structured • the role of an interpreter • the steps taken when the code is executed and how the code can be debugged **4.3** The learner will show how to **debug compiled code** and should cover: • how the code is written and structured • the role of a compiler • the steps taken before the code can be executed • how the code can be debugged • why there is a necessity of compilation for some languages **4.6** The learner should understand some of the advantages/disadvantages of the coding solution.

Delivery and assessment guidance
4.5 For interpreted code, the learner should be given a small task, such as squaring 2 numbers. The learner should be able to write code in an interpreted language that is functional. For compiled code, the learner should be given the same small problem and should be able to create a similar solution (an object-oriented approach is preferred).

Unit 03 Explore methods of mobile app development (L/650/1678)

Unit summary			
In this unit, the learner will explore methods of mobile app development. Starting with a pre-existing concept and codebase, the learner will gather requirements; explore appropriate platforms, integrated development environments (IDEs), application programming interfaces (APIs) and debugging methods; collating and presenting their findings.			
Assessment			
This unit is internally assessed, via a portfolio of evidence.			
Mandatory	**Achieved/not yet achieved**	**Level 3**	**35 GLH**

Learning outcomes The learner will:	**Assessment criteria** The learner can:
1. Know about planning requirements for mobile app development	1.1 Identify the target audience for the app
	1.2 Identify the **functional requirements** of the mobile app
	1.3 Identify the platform operating system and hardware requirements for the mobile app
2. Understand project management methods to gather requirements for mobile app development	2.1 Summarise the **key features** of the following: • scrum • Kanban • waterfall • agile • rapid application development (RAD)
	2.2 Describe the similarities and differences between**:** • scrum • Kanban • waterfall • agile • RAD
	2.3 Consider the **strengths and weaknesses** of each methodology
	2.4 Give an example of when each **methodology** is most suitable
3. Be able to develop a detailed project plan for a mobile app development	3.1 Identify the detailed requirements of the mobile app
	3.2 Analyse the target audience and their requirements
	3.3 Create a visual mock-up of the customer journey
	3.4 Justify the technical solutions chosen to meet customer requirements
	3.5 Present project plan to others
4. Understand APIs	4.1 Explain what is meant by **API**
	4.2 Explain how APIs **work and why** they are used
	4.3 Compare **common types** of API
	4.4 Explain the use and implications of **third-party** APIs in mobile app development

Learning outcomes The learner will:	**Assessment criteria** The learner can:
5. Understand API implementation and permissions for the codebase's target device	5.1 Explain how to **implement** an API in the project's codebase
	5.2 Identify how to test the API **functionality**
	5.3 Identify **required permissions** to use within a mobile app
	5.4 Identify application **technical permission** settings
6. Be able to develop, build and debug a mobile app	6.1 Develop the project's codebase using a relevant language and platform within the chosen IDE
	6.2 Build and run the project's codebase
	6.3 Demonstrate debugging the codebase with appropriate tools and techniques

Range
1. Know about planning requirements for mobile app development
1.2 Functional requirements: The learner must include different types of requirements (business, functional, non-functional), with the knowledge that often requirements are not always complete or hold all required information.
2. Understand project management methods to gather requirements for mobile app development
2.1 Key features: The learner will need to summarise key features, considering the following: • the key aims of the methodology • any dependencies/requirements for use • roles and responsibilities • how work is divided between team members • how the software delivery lifecycle is utilised • tools and processes • checkpoints and milestones • how progress is measured • any impact to the customer **2.3 Strengths and weaknesses:** The learner should consider: • requirements collection • task breakdown • task workflow • change management • the feedback loop

Range
2.4 Methodology: The learner will be able to give examples of different methodologies to use and the circumstances when they become more suitable/unsuitable for selection. The learner should consider: • project and product types • requirements and documentation • compliance and governance • visibility and scope • plan and timescales • staffing and resourcing • volatility
4. Understand APIs
4.1 API: The learner must include a set of definitions and protocols for building, communicating, and integrating application software. **4.2 Work and why:** The learner must include client ‘request’ and application or server ‘response’ and the standardisation of ‘input’ and ‘output’ (IO). **4.3 Common types** must include but not limited to: • web APIs • operating system (OS) APIs • remote APIs • libraries and frameworks (acting as interface) **4.4 Third-party** examples could include but not limited: • security (for example, OAuth2.0 or OAuth, server and client certificates and OpenID) • social media (for example, Twitter, TikTok, Instagram) • cloud-based data storage (for example, mobile backend as a service (MbaaS), (such as, Firebase, Kinvey, Amazon Web Service (AWS) or Apple CloudKit)) • payment methods (for example, Payment Request API, Google Pay, PayPal, Stripe)

Range
5. Understand API implementation and permissions for the codebase's target device
5.1 Implement: API implementation should focus on intended device form factor (for example, phone, tablet, TV, wearable, automotive). **5.2 Functionality:** API functionality should focus on OS APIs (for example, camera, Bluetooth, GPS, and audio) **5.3 Required permissions** should focus on user privacy, including but not limited to: • control: this is when the user has control over the data that is shared with apps • transparency: this is when the user understands what data an app uses, and the reasons for accessing the data • minimisation: this is when an app accesses and uses only the data necessary for that task or action that the user raises **5.4 Technical permissions** should include but not limited to: • install time • normal • signature (optional) • runtime • special

Delivery and assessment guidance
2.4 Learners could be given a scenario where they have to select the appropriate methodology to suit that mobile app development task.

Unit 04 UX/UI in mobile app development (M/650/1679)

Unit summary			
In this unit, the learner will explore the broad concepts of user experience/user interface (UX/UI) and their application and implementation in mobile app development.			
Assessment			
This unit is internally assessed, via a portfolio of evidence.			
Mandatory	**Achieved/not yet achieved**	**Level 3**	**30 GLH**

Learning outcomes The learner will:	**Assessment criteria** The learner can:
1. Know about UX/UI	1.1 Explain what **UX/UI** is
	1.2 Explain the importance of **UX** in mobile app development
	1.3 Explain the importance of **UI** in mobile app development
2. Understand how UX/UI design is applied to mobile app development	2.1 Explain how **UI** is utilised in mobile app development to enhance **UX**
	2.2 Identify the purpose of **usability guidelines** in relation to mobile app development
	2.3 Describe how **accessibility guidelines** enhance UX/UI in mobile app development
	2.4 Explain how UX should be adaptable based on: • **end user device** • **target audience**
3. Understand the stages of the UX/UI design process for mobile app development	3.1 Explain the purpose of the 5 stages of the **design thinking** process in relation to mobile app development
	3.2 Explain **methods** to map initial UX/UI requirements in mobile app development
	3.3 Describe **key planning** methods for UX/UI in mobile app development
	3.4 Identify an appropriate framework to a UX/UI problem for mobile app development
4. Be able to prepare a range of assets for a mobile app development project	4.1 Identify a **range** of software required to develop **assets**
	4.2 Develop interface elements for a mobile app
	4.3 Design and develop a range of assets for a mobile app
5. Be able to develop and test a prototype	5.1 Develop a UI prototype for a pre-existing mobile app codebase
	5.2 **Test** a UX prototype using a range of functional and non-functional testing methods
	5.3 Justify how the prototype meets usability and accessibility guidelines

Range
1. Know about UX/UI
1.1 UX/UI should include a brief overview and history of human computer interaction (HCI) design. **1.2 UX** should include but not limited to: • factors in HCI • usability • user-centred design (UCD) • principles of design **1.3 UI** should include but not limited to: • principles of design • factors of UI • the history of UI • a graphical user interface (GUI) • historical device milestones (for example, wireless, tracking, wearable devices)
2. Understand how UX/UI design is applied to mobile app development
2.1 UI utilisation heuristics for **UX** enhancement could include but not limited to: • scale • visual hierarchy • balance • contrast • Gestalt principles • visibility of system status • match between system and the real world • user control and freedom • consistency and standards • error prevention • recognition rather than recall • flexibility and efficiency of use • aesthetic and minimalist design • recognise, diagnose and recover from errors • help and documentation **2.2 Usability guidelines:** The learner could use Google and Apple usability guidelines, or Nielsen's heuristics, to provide examples to improve usability. **2.3 Accessibility guidelines** should be based on W3C's most recent web content accessibility guidelines (WCAG) Guideline principles must include: • perceivable • operable

Range
• understandable • robust Each principle's structure should model the following: • guideline • success criteria • techniques **2.4 End user device** and **target audience:** Examples should include but not limited to: • customer requirements • target devices • intended platforms • end user preferences • accessibility • variability (for example, screen orientation)
3. Understand the stages of the UX/UI design process for mobile app development
3.1 Design thinking stages must include: • empathise • define (the problem) • ideate • prototype • test **3.2 Methods** examples should include but not limited to: • user research • intended platform/environment of use (how and where) • context strategy • user stories **3.3 Key planning** examples should include but not limited to: • mind maps • prototypes • information architecture • cross-platform/device/browser • UX copywriting

Range
4. Be able to prepare a range of assets for a mobile app development project
4.1 Range of software to develop assets must include, but not limited to, the design of: • icons • logos • background images • fonts • sounds **4.1 Assets** should include but not limited to: • icons • logos • background images • fonts • sounds
5. Be able to develop and test a prototype
5.2 A UX **test** must include but not limited to: • business testing • regression testing Other testing methods can be used. **5.3** This can be an annotated prototype to identify why and where particular elements have been used.

Delivery and assessment guidance
3.4 The tutor will need to provide a scenario or case study, for example: • missing tutorial • cluttered interface • compatibility between devices (phone vs tablet) **5.1** UI prototype must be a minimum of 2 screens.

Unit 05 Legal and security requirements in mobile app development (Y/650/1680)

Unit summary			
In this unit, the learner will cover key legislation in mobile app development. The learner will understand the security and best practice requirements when managing data in mobile app development.			
Assessment			
This unit is internally assessed, via a portfolio of evidence.			
Mandatory	**Achieved/not yet achieved**	**Level 3**	**15 GLH**

Learning outcomes The learner will:	**Assessment criteria** The learner can:
1. Know about key legislation in mobile app development	1.1 Identify copyright and trademark **legislation**
	1.2 Identify key legal requirements in relation to mobile app development
	1.3 Identify licensing requirements for open-source software and **third- party** libraries
	1.4 Explain **ethical considerations** for computing professionals
2. Know about legislative requirements when managing data in mobile app development	2.1 Identify the data protection principles relating to mobile app development
	2.2 Describe ways to protect **stored data**
	2.3 Explain how data can be used to support app development and maintenance
	2.4 Discuss why data disposal is important and the steps taken to implement this
	2.5 Identify differences between internal and external data sources
3. Understand security best practice for mobile app development	3.1 Discuss the importance of writing secure code
	3.2 Explain what is meant by encryption and why it is needed when developing apps
	3.3 Discuss the effectiveness of common **authentication methods** used in app development
	3.4 Explain solutions for **secrets management**
	3.5 Explain the purpose of adding tampering detections to an app

Range
1. Know about key legislation in mobile app development
1.1 UK **legislation** should include, but not limited to, the Copyright, Designs and Patents Act 1988 and the Copyright (Computer Programs) Regulations 1992. **1.2** Global implications using the app overseas learners will need to be aware that legislation differs across nations. **1.3** Sometimes apps cannot go into app stores because of licensing requirements. **1.3 Third-party** packages should include consideration for legislation. Research in third-party libraries should include:

Range
• what are they (pre-existing code maintained by external parties) • why make use of them (for example, reduce implementation effort, implement new functionality, provide additional security) **1.4 Ethical considerations** (including but not limited to): • Institute of Electrical and Electronics Engineers – Computer Society (IEEE-CS) software engineering ethics and professional practices • Association for Computing Machinery (ACM) code of ethics and professional conduct • British Computer Society (BCS) code of conduct • Information Commissioner's Office (ICO) • energy sustainability in data storage
2. Know about legislative requirements when managing data in mobile app development
2.2 Stored data: • data at rest • data in transit
3. Understand security best practice for mobile app development
3.3 Authentication methods: Should include but not limited to: • pin • password • one time password (OTP) • biometrics • two-factor authentication (2FA) **3.4 Secrets management** in source code (for example, environment variables) secrets – API keys, usernames, '.gitignore' files (so that secrets are not uploaded to source control platforms, such as GitHub). Should include but not limited to: • eliminating hardcoded and default passwords • enforce password security best practices such as password length, complexity, and expiration • log, audit, and monitor against suspicious activity • threat analytics

Delivery and assessment guidance
No recommendations for this unit.

Unit 06 Create and develop a mobile app (A/650/1681)

Unit summary			
In this unit, learners will create and develop a mobile app, following the full software development lifecycle, utilising a range of skills, techniques and requirements.			
Assessment			
This unit is internally assessed, via a portfolio of evidence.			
Mandatory	**Achieved/not yet achieved**	**Level 3**	**40 GLH**

Learning outcomes The learner will:	**Assessment criteria** The learner can:
1. Be able to create a project plan to develop a mobile app	1.1 Create a project plan for a new mobile app to include: • customer requirements • user requirements • design specification • development environment • methodology for project management • **app store marketing content**
2. Be able to select, develop and debug a skeleton design template	2.1 Select the app **template** for a project
	2.2 Create a screen **design template** to include: • colour scheme • icons • fonts • menu bars • assets
	2.3 Perform functional tests on the design template using common UI **elements**
3. Be able to implement the core app architecture	3.1 Identify and include required components, packages and libraries
	3.2 Identify data sources and APIs
	3.3 Identify and implement requested **permissions** for the chosen APIs
	3.4 Implement the app's **data sources and APIs**
4. Be able to create view models that reflect the app functionality	4.1 Define the data models
	4.2 Create **view models** that implement data functionality
	4.3 Create screens that are bound to the **view model** for each screen in the app
5. Be able to test, debug and iterate the app	5.1 Implement local testing
	5.2 Implement debugging measures based on **testing** errors
	5.3 Implement **end user** testing to enhance the development process
	5.4 Review, revise and iterate for a final complete mobile app

Range
1. Be able to create a project plan to develop a mobile app
1.1 App store marketing content could include**:** • listing title • description • category • keywords • launch icon (logo)
2. Be able to select, develop and debug a skeleton design template
2.1 Template: Android architecture components should include but not limited to: • the view • lifecycle library • intents • material theme iOS components should include but not limited to: • auto layout overview • UIStackView • a range of view controllers in interface builder All should include: • screen transitions • state transitions • navigation basics • call back functions • process transitions • tabs • grids • layout and styling • overall UI styles and themes **2.2 Design template** should be implemented through components listed in AC2.1. **2.3 Elements** – using common elements listed in AC2.2, tests should include buttons, labels, text, area/whitespace.
3. Be able to implement the core app architecture
3.3 Permissions could include key management and secrets • requests to remote web APIs • data representation and processing (for example, parsing JSON) • asynchronous behaviour • network integration (for example, hypertext transfer protocol (HTTP) requests)

Range
3.4 Data sources and APIs: This will be project dependent and could include but not limited to: • authentication/user management • push notification services • database services • social media integration • app analytics All the above are often bundled in an additional SDK with RESTful APIs (for example, Firebase or Parse) Common API examples would include: • Google Places API • Eureka • Google Play services
4. Be able to create view models that reflect the app functionality
4.1 Platform dependent models for mobile apps would include, but not limited to: • scrollable lists of items (for example, Android's RecyclerView and iOS's UITableView) • adaptors • view holders • locations • maps • Android Manifest **4.2 View models-** the view model links the model and the view and the following should be considered: • view ports • table views • event handling **4.3** The **view model** retrieves required data from the model for the view
5. Be able to test, debug and iterate the app
5.2 Testing: This includes functional and non-functional testing, pre-compilation testing, post-compilation testing, virtual device testing, and physical device testing. **5.3 End user:** Testing should be documented and evidenced within the final iterative development process and overall submitted documentation.

Delivery and assessment guidance
No recommendations for this unit.

Assessment strategies and principles relevant to this qualification

The units we offer have been developed in line with the specific assessment strategies or principles of different Sector Skills Councils (SSCs) or by us where there is no SSC lead.

The key requirements of the assessment strategies or principles that relate to units in this qualification are summarised below.

The centre needs to ensure that individuals undertaking assessor or quality assurer roles within the centre conform to the SSC or our assessment requirements for the unit they are assessing or quality assuring.

Assessment strategy

Knowledge learning outcomes:

- assessors will need to be both occupationally knowledgeable and qualified to make assessment decisions
- internal quality assurers will need to be both occupationally knowledgeable and qualified to make quality assurance decisions

Competence/skills learning outcomes:

- assessors will need to be both occupationally competent and qualified to make assessment decisions
- internal quality assurers will need to be both occupationally knowledgeable and qualified to make quality assurance decisions

Section 3: explanation of terms

This table explains how the terms used at level 3 in the unit content are applied to this qualification (not all verbs are used in this qualification).

Apply	Explain how existing knowledge can be linked to new or different situations in practice.
Analyse	Break the subject down into separate parts and examine each part. Show how the main ideas are related and why they are important. Reference to current research or theory may support the analysis.
Clarify	Explain the information in a clear, concise way.
Classify	Organise according to specific criteria.
Collate	Collect and present information arranged in sequential or logical order.
Compare	Examine the subjects in detail and consider the similarities and differences.
Critically compare	This is a development of compare where the learner considers the positive aspects and limitations of the subject.
Consider	Think carefully and write about a problem, action or decision.
Demonstrate	Show an understanding by describing, explaining or illustrating using examples.
Describe	Write about the subject giving detailed information in a logical way.
Develop (a plan/idea which…)	Expand a plan or idea by adding more detail and/or depth of information.
Diagnose	Identify the cause based on valid evidence.
Differentiate	Identify the differences between 2 or more things.
Discuss	Write a detailed account giving a range of views or opinions.
Distinguish	Explain the differences between 2 or more items, resources, or pieces of information.
Draw conclusions (which…)	Make a final decision or judgement based on reasons.
Estimate	Form an approximate opinion or judgement using previous knowledge or considering other information.

Evaluate	Examine strengths and weaknesses, arguments for and against and/or similarities and differences. Judge the evidence from the different perspectives and make a valid conclusion or reasoned judgement. Reference to current research or theory may support the evaluation.
Explain	Provide detailed information about the subject with reasons showing how or why. Responses could include examples to support these reasons.
Extrapolate	Use existing knowledge to predict possible outcomes that might be outside the norm.
Identify	Recognise and name the main points accurately (some description may also be necessary to gain higher marks when using compensatory marking).
Implement	Explain how to put an idea or plan into action.
Interpret	Explain the meaning of something.
Judge	Form an opinion or make a decision.
Justify	Give a satisfactory explanation for actions or decisions.
Perform	Carry out a task or process to meet the requirements of the question.
Plan	Think about and organise information in a logical way using an appropriate format.
Provide	Identify and give relevant and detailed information in relation to the subject.
Reflect	Learners should consider their actions, experiences or learning and the implications of this for their practice and/or professional development.
Review and revise	Look back over the subject and make corrections or changes.
Select	Make an informed choice for a specific purpose.
Show	Supply evidence to demonstrate accurate knowledge and understanding.
State	Give the main points clearly in sentences or paragraphs.
Summarise	Give the main ideas or facts in a concise way.

Section 4: support

Support materials

The following support materials are available to assist with the delivery of this qualification and are available on the NCFE website:

- learner's evidence tracking log (LETL)
- learning resources
- qualification factsheet

Useful websites

Centres may find the following websites helpful for information, materials and resources to assist with the delivery of this qualification:

- www.w3schools.com/w3css/w3css_mobile.asp
- https://dotnet.microsoft.com/en-us/learn/xamarin/hello-world-tutorial/intro
- https://developer.android.com/training/basics/firstapp

These links are provided as sources of potentially useful information for delivery/learning of this subject area. NCFE do not explicitly endorse any learning resources available on these websites. For official NCFE endorsed learning resources, please see the additional and teaching materials sections on the qualification page on the NCFE website.

Other support materials

The resources and materials used in the delivery of this qualification must be age-appropriate and due consideration should be given to the wellbeing and safeguarding of learners in line with your institute's safeguarding policy when developing or selecting delivery materials.

Reproduction of this document

Reproduction by approved centres is permissible for internal use under the following conditions:

- you may copy and paste any material from this document; however, we do not accept any liability for any incomplete or inaccurate copying and subsequent use of this information
- the use of PDF versions of our support materials on the NCFE website will ensure that correct and up-to-date information is provided to learners
- any photographs in this publication are either our exclusive property or used under license from a third-party:
    - they are protected under copyright law and cannot be reproduced, copied or manipulated in any form
    - this includes the use of any image or part of an image in individual or group projects and assessment materials
    - all images have a signed model release

Contact us

NCFE
Q6
Quorum Park
Benton Lane
Newcastle upon Tyne
NE12 8BT

Tel: 0191 239 8000*
Fax: 0191 239 8001
Email: customersupport@ncfe.org.uk
Website: www.ncfe.org.uk

Version 1.1 July 2023
Information in this qualification specification is correct at the time of publishing but may be subject to change.

NCFE is a registered charity (Registered Charity No. 1034808) and a company limited by guarantee (Company No. 2896700).

To continue to improve our levels of customer service, telephone calls may be recorded for training and quality purposes.

Appendix A

Units

To make cross-referencing assessment and quality assurance easier, we have used a sequential numbering system in this document for each unit.

Knowledge only units are indicated by a star. If a unit is not marked with a star, it is a skills unit or contains a mix of knowledge and skills.

Mandatory units

Unit number	Regulated unit number	Unit title	Level	GLH	Notes
Unit 01	J/650/1676	Introduction to mobile app development	3	30	
Unit 02	K/650/1677	Programming fundamentals for mobile app development	3	35	
Unit 03	L/650/1678	Explore methods of mobile app development	3	35	
Unit 04	M/650/1679	UX/UI in mobile app development	3	30	
Unit 05	Y/650/1680	Legal and security requirements in mobile app development	3	15	
Unit 06	A/650/1681	Create and develop a mobile app	3	40	

The units above may be available as stand-alone unit programmes. Please visit our website for further information.