

# Mobile App Development with LongRange

*A White Paper*

October 2014



The information contained in this document represents the current view of LANSA on the issues discussed as of the date of publication. Because LANSA must respond to changing market conditions, it should not be interpreted to be a commitment on the part of LANSA, and LANSA cannot guarantee the accuracy of any information presented after the date of publication. In many cases, information in this document is dependent on information from third-party vendors.

The statements in this document about specific product features represent the current status or planned intentions of LANSA within a one-year time frame of date of publication.

LANSA development plans are subject to change or withdrawal without further notice. Any reliance on this document is at the relying party's sole risk and will not create any liability or obligation for LANSA.

This document is for informational purposes only. LANSA MAKES NO WARRANTIES, EXPRESSED OR IMPLIED, IN THIS DOCUMENT.

V1.3

© 2014 LANSA. All rights reserved.

IBM Emblem Version Logotype IBM i, System i and iSeries are trademarks of International Business Machines Corporation (IBM). All other trademarks are acknowledged.

**The Americas:**

Headquarters — Chicago, USA  
Tel: +1 630 874 7000  
Email: [info@lansa.com](mailto:info@lansa.com)

**Europe:**

Headquarters — London, UK  
Tel: +44 1727 790300  
Email: [info@lansa.co.uk](mailto:info@lansa.co.uk)

[www.lansa.com](http://www.lansa.com)

**Asia Pacific:**

Headquarters — Sydney, Australia  
Tel: +61 2 8907 0200  
Email: [info@lansa.com.au](mailto:info@lansa.com.au)

## Contents

Executive Summary	4
Why do companies need mobile apps?	5
Why did we build LongRange?	5
What business problem does LongRange address?	6
Manage development costs	6
Reduce time-to-market and total-cost-of-ownership	7
How does LongRange reduce development costs?	7
How to decide what your mobile apps will do?	8
What is LongRange?	10
How does LongRange work?	10
The LongRange mobile app	11
What does a LongRange app look like?	12
Developer tools	13
IBM i development tools	13
LongRange Studio	13
Testing and debugging applications	13
Applications can access mobile device features	14
Applications can send files to a mobile device from the server	14
Operate offline without a network connection	14
Sample applications and templates	14
Deployment	14
Prototyping and designing applications	15
Security	15
Why use LongRange?	16
Mobile device platforms	16
Application architecture	16
Consistent quality and application management	17
Productive development	17
Maintenance and enhancement	17
Choosing development tools	18
Is the same user interface appropriate for every audience?	18
Q&A	19

## Executive Summary

Mobile devices have become increasingly popular with consumers for accessing email, social media, movies, books, music and apps. Companies have also begun using mobile device apps to streamline business processes and improve employee mobility, especially for employees who travel to visit customers and suppliers. The challenge is deciding the best way to build the apps. Building a mobile device app is a new programming experience, especially for IBM i developers who have a deep knowledge of IBM i application development but little or no knowledge about building apps for mobile devices.

LongRange is a bridge that transforms RPG, COBOL or CL programs into apps for mobile devices. With LongRange, IBM i applications can run as native mobile apps. Developers can build and maintain native mobile apps for Apple iOS or Android using RPG, COBOL or CL development tools and methods. LongRange requires no programming on the mobile device. However, LongRange can take photos, make videos, capture audio recordings, work with documents, use maps and geo-location information in IBM i applications. Developers don't have to learn any new programming languages such as Java (Google Android), Objective-C (Apple iOS) or other coding techniques like HTML, CSS and JavaScript.

This paper describes LongRange and suggests how it can help companies to build mobile device apps using only IBM i developer skills. Deciding what an app should do is key to its success. The sections about business problems and how to decide what an app should do, will help you to identify where a mobile device app is the best fit. Managing development costs is an important consideration and LongRange offers assistance in this area as well.

If you are a CIO or RPG developer who wants to get into mobile device apps for business but lack the essential knowledge and skills – LongRange provides the solution.

## Why do companies need mobile apps?

It is no longer practical for most companies to rely on established channels to publish product information in a printed brochure and/or a Web site. Mobile devices have changed the way people consume information, purchase products and pay bills, and they do so from anywhere, including their office, home, car or a coffee shop. Companies must find new ways to publish and distribute information to employees and customers and they need not only a printed brochure and a Web site, but now also at least one mobile device app.

The growing use of mobile devices has brought a new paradigm to software application design and development tools. This evolution has two effects. Firstly, it changes the way developers understand what constitutes an application. Developers skilled in building multi-tiered, line-of-business applications for mainframe and mid-range servers need to adopt new design patterns, principles and tools to equip themselves for developing mobile device apps. Secondly, the evolution offers opportunities for companies to optimize and enhance business processes. Mobile devices are personal and users are enthusiastic about using the devices. Companies can make use of this positive attitude by designing mobile apps that provide information from line-of-business applications, perform discrete business transactions and present information targeted to the needs of individual employees and customers.

Customers and employees now expect to work with mobile devices and this expectation will force companies to develop mobile applications that must integrate with or extend current business processes.

## Why did we build LongRange?

LANSA built LongRange to enable the IBM i community to make their applications available to mobile device users quickly and easily. LongRange fast-tracks the transition of IBM i developers to building mobile device applications.

Normally, developers who are familiar with RPG, COBOL or CL must learn new programming languages, such as Java or Objective-C before they can build native mobile device apps. Similarly, they need to learn HTML, CSS and JavaScript to build Web or hybrid apps for mobile devices. These activities will incur training costs and reduce developer productivity while developers take time to become familiar with new tools and the eccentricities of the languages.

If developers were able to use their existing tools and skills to produce native mobile apps there would be no new languages to learn and developers would be productive immediately.

LongRange makes this a reality by providing a native mobile device app, server software and development tools that allow IBM i developers to use RPG, COBOL or CL to build applications. The final result is a native mobile device app that looks and behaves as one would expect on a mobile device.

## What business problem does LongRange address?

LongRange allows companies to achieve more from their existing information technology investments by extending applications to mobile devices without a large investment in infrastructure or a long software development project timeline.

For many years now, companies have been using desktop (rich client) and Web applications (rich Internet applications) to interact with customers and employees. These applications are optimized for desktop and laptop computers with large screens and with few constraints on power and local storage. Today there is an increasing number of people who prefer to use mobile devices such as smart phones and tablets to view information and use applications. Accessing rich client applications from a mobile device is problematic. Accessing existing Web applications from mobile devices using the installed browser is easier. However, these applications were not designed with the device constraints in mind and may not be fully compatible with the mobile device's browser and screen size. The problem for companies is how best to create new applications and/or redesign Web applications for use with mobile devices. Evaluating the situation also offers an opportunity to consider whether existing business processes are appropriate for applications on mobile devices.

The growth in the use of mobile devices coupled with expanding capabilities of these devices offers opportunities for more flexible work practices and productivity improvements. Device mobility provides access to line-of-business applications and corporate data from anywhere. Device size provides a convenient user interface easily held in one hand. Because customers can obtain information about products and services, place orders and make payments from a mobile device at any time and from any location, the administrative overhead of order management is greatly reduced. Also, employees can work with line-of-business applications using smart phones and tablets without having to carry a larger laptop-style device.

Mobile apps built using LongRange can use storage on a mobile device and operate online when connected to a network, or offline when no network is available. You can use these features to streamline business processes that use data when employees can't connect to a corporate or public network. Your mobile workforce need not suffer reduced productivity when communications are unavailable.

Productivity improvement offers more flexibility in managing profitability than cost cutting as there is a limit to the costs companies can reduce. LongRange enables developers to create applications for mobile devices using their existing development tools, thereby allowing companies to implement a mobile strategy without an extended period of reduced productivity while developers learn new programming languages and gain experience developing native apps for mobile devices.

## Manage development costs

The business world changes at a rapid rate. To keep pace with change, companies must be able to develop applications quickly at a minimum cost. Developing applications quickly means that you can respond to business changes in the shortest possible time. LongRange can reduce the time-to-market because developers can build applications without a delay in learning new skills. Similarly, LongRange can reduce the total-cost-of-ownership by using development tools and practices already familiar to developers.

LongRange encourages results-oriented programming by allowing developers to concentrate on business requirements and development rather than mobile device technology.

## Reduce time-to-market and total-cost-of-ownership

Using LongRange, companies can develop and deploy applications for mobile devices quickly and reduce the time-to-market. The total-cost-of-ownership for a mobile device app becomes a viable proposition, especially for companies without any previous experience in developing mobile device apps.

- Requirements are understood and agreed quickly as your developers already know your business and systems.
- Cost of developing mobile device apps is less as your developers already have development skills and experience building applications so there are no additional training costs.
- Maintenance costs are controllable as your developers maintain the applications in-house.
- Prototyping applications reduces the development time.
- No need to hire new developers with costly and specialized mobile device app development skills and no time wasted in the hiring process.

With LongRange, no large scale projects are required as development is incremental and evolutionary. Developers use familiar tools and LongRange complements these tools.

## How does LongRange reduce development costs?

LongRange reduces the need for developers to learn new design paradigms and programming languages. Suppose you were to assign the task of developing a native mobile device app to an RPG developer who had no training or experience in developing apps for mobile devices.

Without LongRange, the developer has to perform the following tasks before producing a useful mobile device app:

- Understand the mobile app design paradigm (which is very different to IBM i applications).
- Learn a new programming language.
- Study the user interface design guide.
- Learn the APIs.

Using LongRange, the same developer would be able to produce a working native mobile device app using only RPG, COBOL or CL. Your developers already have the training and experience with these development languages.

If you outsource the development of a native mobile app to a contract programmer or a company specializing in native mobile device app development, you will incur these additional costs:

- Project and engagement management
- Understanding and agreeing requirements
- Software development and testing

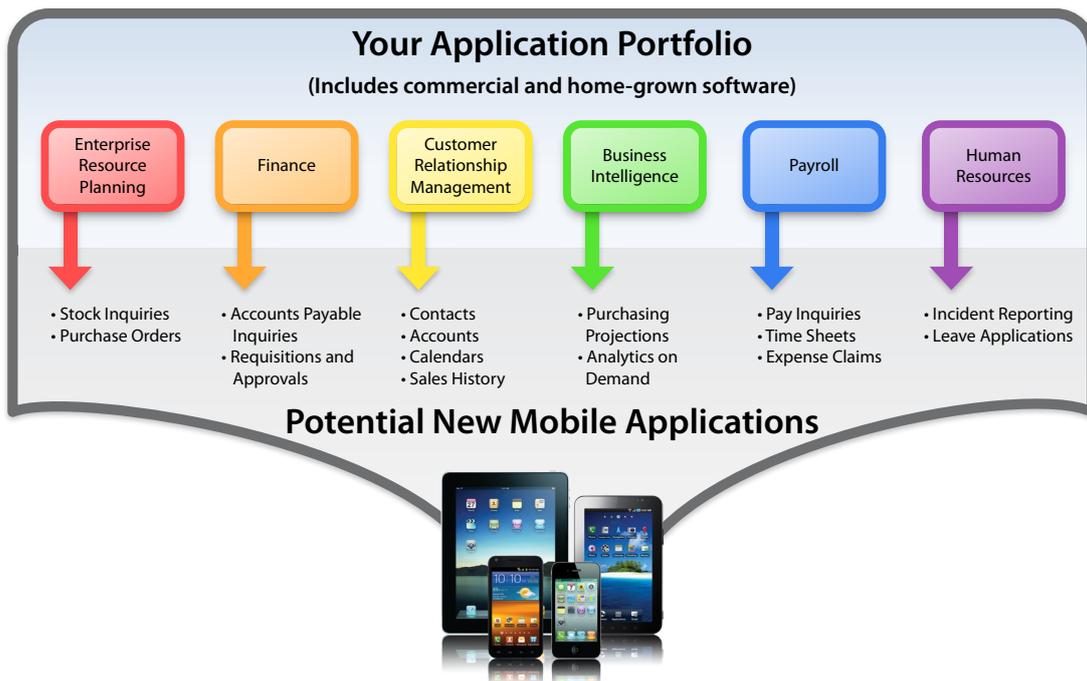
At the end of the project you must choose one of two options, either take responsibility for ongoing maintenance and enhancement, or continue to pay a third party for this work. If you take responsibility

for ongoing maintenance and enhancements you must either hire a developer who has the right skills or train your own developers, thereby incurring additional costs in salary and/or training, and if the new developer moves on you will incur the cost of hiring a replacement. By outsourcing maintenance and enhancements you will incur ongoing costs. Staff turnover at the outsourcing company will constrain their knowledge of your app and business requirements and this will also add to the ongoing cost.

Using LongRange and your existing RPG, COBOL or CL developers removes these additional costs and your developers are already aware of the business requirements and your current systems. Therefore, they understand where mobile device apps will fit into business processes and are able to get started quickly.

## How to decide what your mobile apps will do?

A mobile device app is most unlikely to replace existing ERP or other line-of-business applications, but many of these applications do not provide native apps for mobile devices. The mobile context is becoming a popular method for accessing information, and participating in business processes. Mobile workers need information wherever they are and mobile device apps allow them to access information focusing on only the information they need at a point in time.



*Figure 1: Potential New Mobile Applications*

How do you decide which business processes are the most appropriate candidates for native mobile device apps? Most companies don't build mobile device apps just because it's the fashionable thing to do. Software is a tool to help companies make and distribute products and/or provide services more efficiently. Therefore, the question is what business processes on a mobile device will make manufacturing, service provision, selling and distribution less expensive and more productive?

Business processes suitable for mobile device apps will vary across industries. One example of a business process is a stock take. Tablets can scan barcodes, are easily portable and can communicate with a server in real time. A stocktaking app on a tablet can replace dedicated stock take devices and/or paper-based stock takes. Replacing a broken tablet is easier than replacing a dedicated device, and also a tablet is a generic device that can support multiple mobile device apps. Warehouse employees can use the same tablet to perform a stock take, lodge a leave request, and record their working hours.

Examples of business processes for mobile device apps are:

- Order inquiries to view open orders, closed orders, delivery status, stock item availability, pricing and discounts.
- Purchase order tasks including create purchase orders, review purchase orders awaiting approval, add comments and instructions, and approve or reject purchase orders.
- Business intelligence inquiries based on real-time data.
- Expense claim tasks include entering expenses, submitting expense claims for approval, and managers approving or questioning a claim.
- Employee time sheets.
- Requisitions.

Let's study an example. Providing self-service requisitions to mobile employees allows them to participate in procurement workflows. They can enter a requisition and then monitor its progress through the approval and fulfillment cycle from their mobile device without returning to the office.

The questions to ask when choosing business processes for mobile apps are:

- Can a mobile device app do something new?
- Can a mobile device app automate a business process that so far has been impractical to automate?
- Can a mobile device app improve an existing business process in a demonstrably faster and/or less expensive way?
- Is the business process practical to operate in a mobile context when away from the office?

Choose narrow slivers or subsets of an application's functionality to redesign and develop for mobile devices. A high value sliver of an application's functionality is a discrete business transaction that customers or employees use regularly and/or practical to access from mobile devices. This style of application (also known as point applications) focuses on a narrow set of requirements and can provide quick answers to everyday business questions.

The mobile evolution is here and it's time to start thinking about how it can be integrated into the day-to-day operations of most businesses.

## What is LongRange?

LongRange consists of a server-side management service (LongRange Server), an app that runs natively on a mobile device (LongRange mobile app) and a developer tool (LongRange Studio).

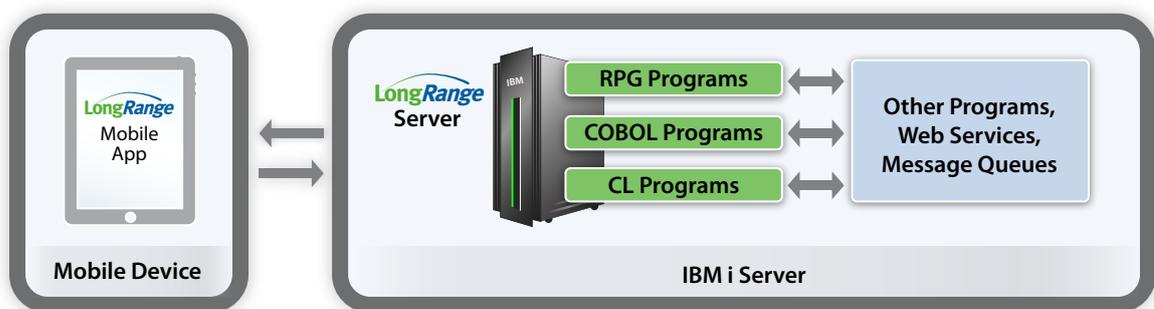
<b>LongRange server</b>	LongRange Server is the interface between IBM i programs and the LongRange mobile app. It manages requests from the app and calls the appropriate IBM i programs.
<b>LongRange mobile app</b>	LongRange mobile app is the intermediary between the mobile device user and the LongRange server. It responds to user actions and displays screens generated by IBM i programs.
<b>LongRange Studio</b>	LongRange Studio is the tool developers use to define the application model and content in an application schema.

Administrators download and install LongRange server on an IBM i server. Developers download and install LongRange Studio on a desktop or notebook computer.

Users download the LongRange mobile app to their mobile device, connect to the server and are ready to use the applications.

## How does LongRange work?

Figure 2 (below) illustrates the LongRange deployment. The LongRange mobile app runs on iOS or Android mobile devices. LongRange Server runs on an IBM i server with the programs (RPG, COBOL or CL) that are part of the application.



*Figure 2: How Does LongRange Work?*

When a user of a mobile device invokes a function in the LongRange based mobile app, the action sends a request to the LongRange Server, which calls the associated RPG, COBOL or CL program. The program performs its processing and outputs a data stream of information. The LongRange Server sends the data stream to the LongRange mobile app which then presents the information on the mobile device. The LongRange mobile app is faster than browser-based mobile apps when rendering screens and responding to user actions because the app is designed to run natively on the mobile device.

## The LongRange mobile app

The LongRange mobile app is a native mobile app and its screen layout comprises panes for navigation, tabs, commands and a content area where developers add form views (screens generated by IBM i programs), web views (HTML applications, pages and sites), and document views.

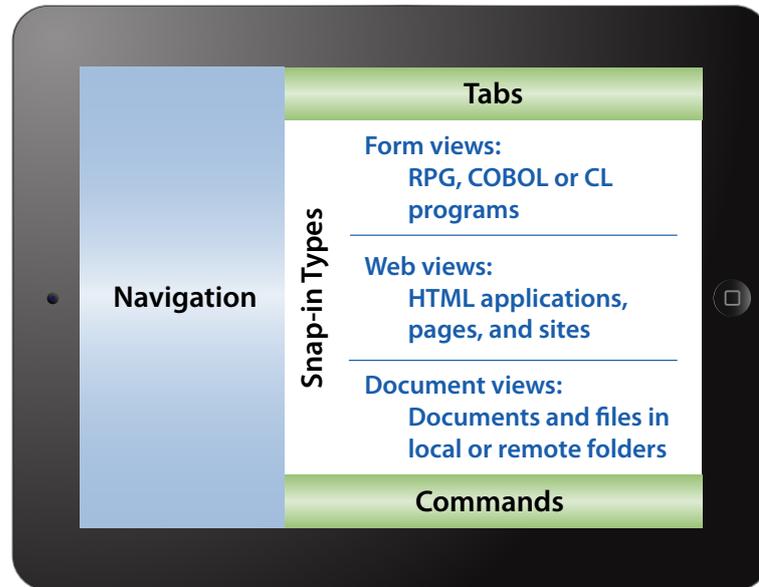


Figure 3: Typical LongRange Mobile App Screen Layout

Table 1 (below) describes the elements of the screen layout and explains their purpose.

Table 1: LongRange Mobile App Screen Layout

<b>Commands</b>	Commands are user actions; examples are save and cancel.
<b>Document views</b>	Document views show files and folders either on the mobile device or on a remote server.
<b>Form views</b>	Form views are screens generated by RPG, COBOL or CL programs that the LongRange mobile app renders on the mobile device.
<b>Navigation</b>	Navigation is the equivalent of menus and shows applications accessible from the mobile device.
<b>Tabs</b>	Tabs provide concurrent views of information from multiple IBM i programs and allow users to quickly switch between views.
<b>Web views</b>	Web views are HTML applications, Web pages or sites.

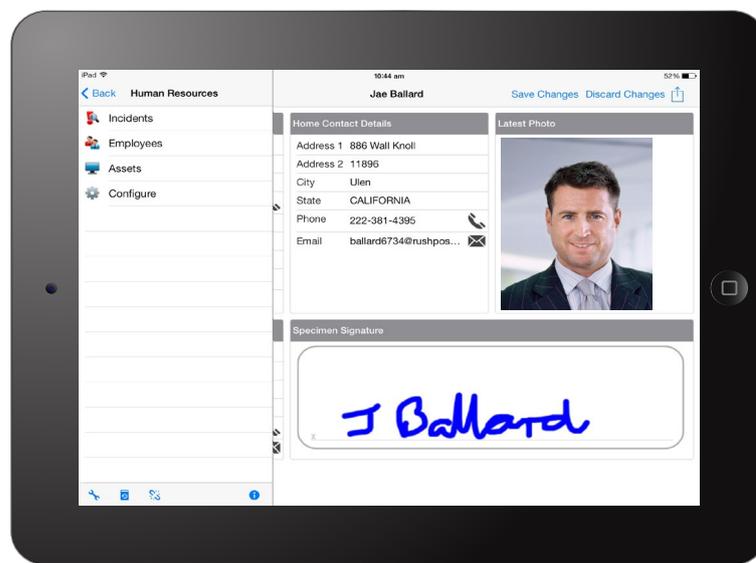
Form views are typical forms that contain data and labels (text describing the data). Developers can build applications using only RPG, COBOL or CL programs invoked via form views, or composite applications using combinations of form views, web views and document views.

Applications developed with LongRange are easily maintained and extensible. Developers can add and change programs without having to redistribute the whole application.

LongRange includes a document view element that shows documents and files in folders on a mobile device or on a server. Users can take photos and store them in folders, accept documents from other apps on the mobile device and save them in folders on the device or a server.

## What does a LongRange app look like?

The LongRange mobile app is a native mobile app that operates and looks like any other native mobile app. It processes input and output from IBM i applications with a significantly enhanced user interface on iOS and Android mobile devices.



*Figure 4: Example of an Application Screen on an iPad*

Key features of a LongRange application are that:

- The apps behave the way a mobile device user expects.
- The IBM i user interface is extended with touch and enhanced visually by mobile devices.
- Applications can send and receive files between a mobile device and a server.
- IBM i applications can use the device-side capabilities including the camera and geo-location services.

The application provides the screen layout and data content that the LongRange mobile app displays on a mobile device. Screen layouts automatically adapt from portrait to landscape and landscape to portrait as the mobile device changes orientation. Screen layouts and content automatically adjust when displayed on small or large screen sizes. Developers do not need to design screen layouts and content for each screen size and orientation.

## Developer tools

To build an app for use with LongRange developers create application content such as forms (or screens) in an IBM i program and an application schema with LongRange Studio.

### IBM i development tools

Developers can use RPG, COBOL or CL to develop applications for use with LongRange. These applications can also call other programs on IBM i servers, use data queues, interact with the database, etc. From a developer's perspective the applications are just like any other IBM i application.

LongRange complements existing developer tools. Therefore, developers can use their existing IBM i application development tools and LongRange Studio to develop applications for use with the LongRange mobile app.

Developers familiar with the Programming Development Manager (PDM) and the Source Entry Utility (SEU) require a developer workstation, LongRange Studio and terminal access to PDM and SEU.

Developers familiar with the IBM Rational Developer for Power Systems require a developer workstation, LongRange Studio and the Rational tools.

### LongRange Studio

Developers use LongRange Studio to describe apps by building application schemas. Application schemas define the static parts of an application including:

- Navigation – menu structure hierarchy and menu item icons.
- Form views – typical forms showing visual controls like labels and text boxes that present information to users.
- Web views – HTML, CSS and JavaScript pages.
- Document views – file and folder explorers showing lists of files and folders that contain documents, presentations, images photos, movies (videos) associated with the application.
- Commands – actions available to users such as save or cancel.

The application schema also includes a reference to the IBM i program to call when a user requests a form view.

### Testing and debugging applications

Developers need iOS and/or Android mobile devices to test their applications.

Developers can use their existing debugging tools to track down errors in an application. Finding and debugging errors in applications developed using LongRange is no different to finding and debugging errors in any other application.

To assist developers find errors, LongRange provides two levels of tracing, one at the application level and a second at the system level. Developers can use application tracing during testing or debugging to track activity of individual applications (or programs). System level tracing has a wider scope and produces much more trace data than application level tracing.

## **Applications can access mobile device features**

Applications that use LongRange have access to files residing on mobile devices, including photos and data. Suppose a business requirement of an insurance claim system is to capture a photo of a motor vehicle accident scene. An assessor with a mobile device can take photos then use the application to save the photos in the database on an IBM i server, or save the files in a folder in the server's file system.

## **Applications can send files to a mobile device from the server**

This feature of LongRange expands the capabilities of IBM i applications beyond the information displayable on a screen. With access to device-side features including the camera, geo-location services, and files stored on the mobile device, applications can capture, manage and display not only screens but also photos, documents, maps, and mobile device geo-location information.

## **Operate offline without a network connection**

Developers can design apps to operate online with a network connection or offline when no network is available. In offline mode, the apps use data stored in a database on the mobile device and forms that allow users to view and edit the data. When a connection is available, the apps can send data to a server and receive data from a server.

Business requirements will suggest the appropriate design for always connected or sometimes connected apps. Always connected apps need to access data on a server in real time. Sometimes connected apps can operate standalone and connect only when they have data to send to a server.

## **Sample applications and templates**

The LongRange mobile app includes sample applications and application schemas.

LongRange includes an extensive collection of sample applications and templates. The sample applications range from the introductory "Hello World" style of application to fully functional applications. When you run a sample application you can see not only the screens and user actions, but also view the program code.

## **Deployment**

Deploying the LongRange mobile app is as simple as downloading the app from an app store and configuring communications with an IBM i server.

Updating the mobile app is the same as enhancing and maintaining any other IBM i application. Once you deploy application updates to your production system they are immediately available to users of the LongRange mobile app, without having download or update anything on the mobile device.

## Prototyping and designing applications

You can use LongRange Studio to rapidly create a complete application prototype and demonstrate it to users. Building a prototype will help to define the parts that need to be built, estimate the programming effort, and also identify what's missing from the design.

Prototypes allow users to understand how an application will operate and see what the screens will look like without having to build the entire application, thereby aligning user and developer expectations of the application.

After approval of the prototype, developers build the application progressively, adding each completed program into the prototype and gradually turning the prototype into a working application.

This development methodology is agile and evolutionary. Users do not wait for months or years to work with a completed application. Instead they see an application that grows and adapts quickly to changing requirements.

## Security

Security is an integral part of LongRange and is designed to protect data and applications while making those applications available to remote users operating mobile devices. LongRange supports a number of physical implementations, each intended to provide different levels of scalability and performance without compromising security.

- Supports the IBM i security and authentication mechanisms up to and including the highest security level (level 50).
- User identification and password are encrypted before transmission.
- User identification and password cannot be cached in the browser.
- Supports log-in from specific IP addresses.
- Supports log-in from specific device names.
- Reverse proxy.

LongRange Server supports Secure Sockets Layer (SSL) and Transport Layer Security (TLS) protocols which allow for secure authentication, encryption, non-repudiation and VPN technologies. Both SSL and VPNs prevent eavesdroppers from listening and intercepting traffic between a browser and a server.

Communications traffic between a mobile device and the server is compressed which offers two major benefits. Firstly, traffic is harder to monitor with sniffer devices. Secondly, when using SSL or VPN technologies, the amount of data transmitted between a server and a mobile device is reduced before the SSL or VPN encryption or decryption processing takes place. Compressing communications traffic achieves considerable savings in bandwidth and CPU cycles.

The applications run on the server and no business logic runs on the mobile device, except when apps operate offline. Therefore, applications and the database are not exposed to the Internet or internal networks.

## Why use LongRange?

Before developing mobile device apps companies must consider issues such as which mobile platforms to support, ease of development and maintenance, and how to ensure the quality of the applications (Table 3, page 18).

**Table 2: Issues to Consider before Developing Apps for Mobile Devices**

<b>Which platform or platforms to target?</b>	Which platform or platforms will you target for development?
<b>What app architecture will you use?</b>	Developers must choose an architecture for their apps. Some of the options are native apps, Web apps and hybrid apps.
<b>How do you ensure consistent quality and management?</b>	How do you ensure consistent quality of the apps, including user interface, icons, performance, application management and deployment?
<b>How can you build apps productively?</b>	How can you deliver the apps in the shortest time and at a minimum cost? Can you absorb developer learning costs?
<b>How easy is ongoing maintenance and enhancement?</b>	How easy will it be to maintain and enhance the apps in future years? Maintenance and enhancement will consume most of the time and cost over the life of an app.

Without incurring the cost of learning new development languages and practices, with LongRange, developers can produce mobile device apps quickly and with consistent quality. Maintenance is fast and less costly than it would normally be with other mobile development tools. LongRange delivers the platform needed to achieve this with your existing business-focused developers, who are already aware of your business requirements and current systems.

## Mobile device platforms

With LongRange, developers can build applications using only RPG, COBOL or CL and have them run as native apps on mobile devices for which a version of the LongRange mobile device app is available.

Applications built for use with LongRange will run, without change, on any mobile platform for which LongRange has a version of its mobile device app. This means you can support multiple mobile platforms simultaneously using the same application and source code.

## Application architecture

The LongRange mobile device app runs natively on each supported mobile device platform. Therefore, using LongRange means that your application architecture is a native app. Native apps perform better than Web or hybrid apps. They are compiled, closer to the hardware and avoid the additional layers of processor overhead inherent in a browser (Web apps) or a framework and a browser (hybrid apps).

It must be noted that IBM i developers can build native, Web or hybrid apps without using LongRange, but they need additional tools and skills. The languages used to develop native mobile apps are Java and Objective-C. Those used to develop Web apps and hybrid apps are HTML5, CSS3 and JavaScript. Therefore,

your developers must learn how to use these languages. Several mobile app frameworks are available that allow you to develop an app and deploy it to multiple mobile device platforms, however, your developers still have to learn HTML5, CSS3 and JavaScript.

If you chose a Web application as the architecture, your app must be compatible with the browser on each mobile device platform you intend to support. If your company allows employees to bring their own devices without constraint, your app must be compatible with the browser on all these mobile devices as well.

HTML5 has a wide variation in feature implementation among HTML5 capable browsers. Developers must learn the specific capabilities of each browser they intend to support and apply extensive testing to ensure that the apps work properly. Development and testing tasks require less effort when development is only for one browser. However, the HTML5 options are constrained by each browser's HTML5 implementation.

## Consistent quality and application management

Developers are not always the best designers of user interfaces and even the ones who can produce an acceptable user interface have differing ideas about how to layout a screen.

The LongRange mobile device app is built to the operational and user interface design standards of mobile device platforms. It manages the user interface for the developer. IBM i developers using LongRange will always produce a consistent user interface that conforms to the quality standards of the mobile device platform.

Application management is a simple task for administrators and developers. The LongRange mobile device app is a download from an app store. Owners of mobile devices can go to an app store, then download and install the LongRange mobile device app. This reduces administrator workload as they do not need to schedule and roll out app upgrades. Applications built for use with LongRange need no special management. Their deployment is the same as any other IBM i application and developers and administrators are already familiar with these processes.

## Productive development

LongRange allows companies to make better use of existing developer skills by developing applications for mobile devices. Using existing knowledge produces applications quickly (development time is short), the applications are maintainable (developers are using existing skills and familiar development practices) and the result is an app with a viable total-cost-of-ownership.

Using LongRange does not change, replace, supplant and/or supersede current development tools and practices. LongRange is a complimentary extension to what developers already do.

## Maintenance and enhancement

Developers maintain and enhance IBM i applications built for use with LongRange using the same tools and methods they use with other IBM i applications.

## Choosing development tools

The reasons for choosing development tools and technologies are varied and not always objective decisions. Most developers prefer to work with familiar tools and a few like to try multiple tools and development languages. In some cases, corporate policy mandates certain technologies and computing platforms and therefore developers do not consider other options. Often the personal preference of developers determines the development tools and technologies.

Table 3 (below) describes the programming languages and development tools that are used to build native, Web and hybrid mobile apps.

**Table 3: Programming Languages and Tools needed to Develop Native, Web and/or Hybrid Apps**

Architecture	Languages	Development Tools
Native app	Java or Objective-C	Mobile device manufacturer tools and SDK for native apps, a framework and/or a code generator.
Web app	HTML5, CSS3, JavaScript	Web development tools and one or more browsers for testing compatibility.
Hybrid app	Java or Objective-C, HTML5, CSS3, JavaScript	Web development tools, one or more browsers for testing compatibility, mobile device manufacturer tools and SDK for native apps, a framework and/or a code generator.

How do you choose an appropriate architecture for mobile app development? This is not a question to ask in isolation as the purpose of the architecture is to satisfy business requirements and corporate objectives. In some cases, a native app is the right choice and in other cases a Web app is a suitable option. Any of these architectures are appropriate when building a mobile device app to order pizza. The user needs to view a menu and select a pizza. The user interface can be either a natively rendered screen or a HTML page. The app needs a communications transport mechanism to send the order to the pizza company, and this can be an Internet connection, Wi-Fi or mobile network. However, a native or hybrid app with access to geo-location services on the mobile device is a better choice than a Web app. Native and hybrid apps can detect the location of the mobile device and can send the person to the nearest store by providing turn-by-turn directions or deliver the pizza to a map location.

Using LongRange, IBM i developers can use the features of mobile devices without worrying about which architecture to use, which development tools to choose and which new skills to learn. They can build an app to order pizza using LongRange and RPG, COBOL or CL. They already have the development tools and skills and LongRange provides the architecture and infrastructure. The benefit for companies is faster mobile device app development and faster development means reduced cost and a quicker response to business needs.

## Is the same user interface appropriate for every audience?

Have you tried using a rich client or Web application designed for a desktop monitor on a smart phone or a tablet? It is not a productive experience as you spend most of the time moving the screen around just to see its content. One solution is to design a new user interface for each device size. This makes the user interface more approachable as users no longer have to move the view continually. However,

the application navigation becomes more complex when users view two or three screens to see the required information. The development effort also increases when maintaining multiple user interfaces of differing sizes.

The productivity question is whether all user audiences need the same user interface. Suppose employees provide paper expense claims to the accounts department and administrative staff transfer the details into the financial system. One way to streamline the process and reduce the accounts department workload is to provide employees with an electronic expense claim presented in a native or Web app. Just reproducing the paper expense claim form in the app will help only the accounts department staff as they work with many of the forms from a fixed location with desktop computers. Individual employees may fill in one form a month so the business process is different for employees claiming expenses. Filling in an expense claim form on a smart phone or tablet using an interface designed for use on a desktop monitor will be problematic.

Designing user interfaces requires an understanding of the audience, the business process and the nature of the device that people will use. Mobile device apps have the advantage of being able to adapt to the size of the screen. An employee can fill out the expense claim form on a smart phone, a tablet or a desktop and experience an appropriate user interface for the device.

## Q&A

Table 4 (below) presents questions and answers about aspect of LongRange.

**Table 4: LongRange Questions and Answers**

Question	Answer
Can I code RPG applications for LongRange using RPGIII?	LongRange supports RPGIV not RPGIII, however, RPGIV applications can call RPGIII programs.
Can I reuse existing 5250 screens?	No. Mobile device users will not know how to use 5250 screens. Modern touch devices do not support 5250 features like function keys and the field-exit key. The LongRange mobile app uses the 5250 interface in a manner that does not support the device specific 5250 and DDS screen handling features.
Can I reuse existing RPG or COBOL programs?	Existing RPG, COBOL and CL programs expect a 5250 context with function keys, unique keys (e.g. field exit) and a user interface designed without a touch interface. This paradigm does not fit into the mobile device context. Applications written for use with LongRange use RPGIV, COBOL or CL and can call any existing IBM i programs, including RPG, COBOL and CL.
Can I use CL programs?	Yes, and also RPG or COBOL.
Can I use COBOL programs?	Yes, and also RPG or CL.
Can I use RPGIV fixed format (columnar) coding?	Yes; however some parts of the code require the use of free format coding.

Question	Answer
Do I have to use RPGIV free format coding?	For some of the coding free format is necessary. You can use fixed format in other parts of the code.
Does LongRange replace RPG, COBOL or CL?	No. It's complementary to your development tools.
How do I obtain the LongRange mobile app?	For iOS devices, download the app from the Apple App Store or your company can distribute the app to its employees. For Android devices, download the app from the Google app store (Google Play).
Should I use LongRange to mobilize and/or modernize existing IBM i applications?	LongRange is not designed as a tool to modernize or mobilize existing IBM i applications. Use LongRange to mobilize business function slivers to new business audiences.
What infrastructure do I need to use LongRange?	An IBM i server plus iOS and/or Android mobile device(s).
What is a form view?	Form views are typical forms that contain data and labels (text describing the data).
What is a web view?	Using web views developers can include content in the application from HTML pages and/or Web sites.
What programming languages do I need?	RPG, COBOL or CL.
What RPG, COBOL or CL development tools do I need?	SEU/PDM (green screen) development environment, or IBM Rational Developer for Power Systems. Plus LongRange studio and mobile device(s) for testing.
What software do I need to use LongRange?	LongRange mobile app installed on a mobile device. LongRange server installed on an IBM i server. LongRange Studio installed on a developer workstation.

Download a free trial from: [www.longrangemobile.com](http://www.longrangemobile.com)