

BARNES & NOBLE

BN.com

nookTM

by Barnes & Noble

QuickNav

B&N : nook ® Platform UI Widget

Lalitha Krishnamurthy
Barnes & Noble

lkrishnamurthy at book dot com

QuickNav

B&N : nook ® Platform UI Widget

Agenda

- nook ® family of tablets
- nook ® Eco-system
- Introducing QuickNav
- Operation and Demo
- Design
- Implementation
- OEM considerations
- Conclusion
- Q&A, Contact Info

nook ® family of tablets

Nook Simple Touch™



nook ® family of tablets ...

Nook Color™



nook ® family of tablets ...

Nook Tablet™



nook ® Eco-system

- nook ® Platform is OEM-customization of Android Operating Environment.
 - Nook Tablet™ is custom Android Gingerbread.
 - Nook Simple Touch™ is custom Android Eclair.
 - Nook Color™ is custom Android Froyo.
- nook ® Ecosystem comprises of nook ® devices, nook ® Cloud, nook ® digital content stores.

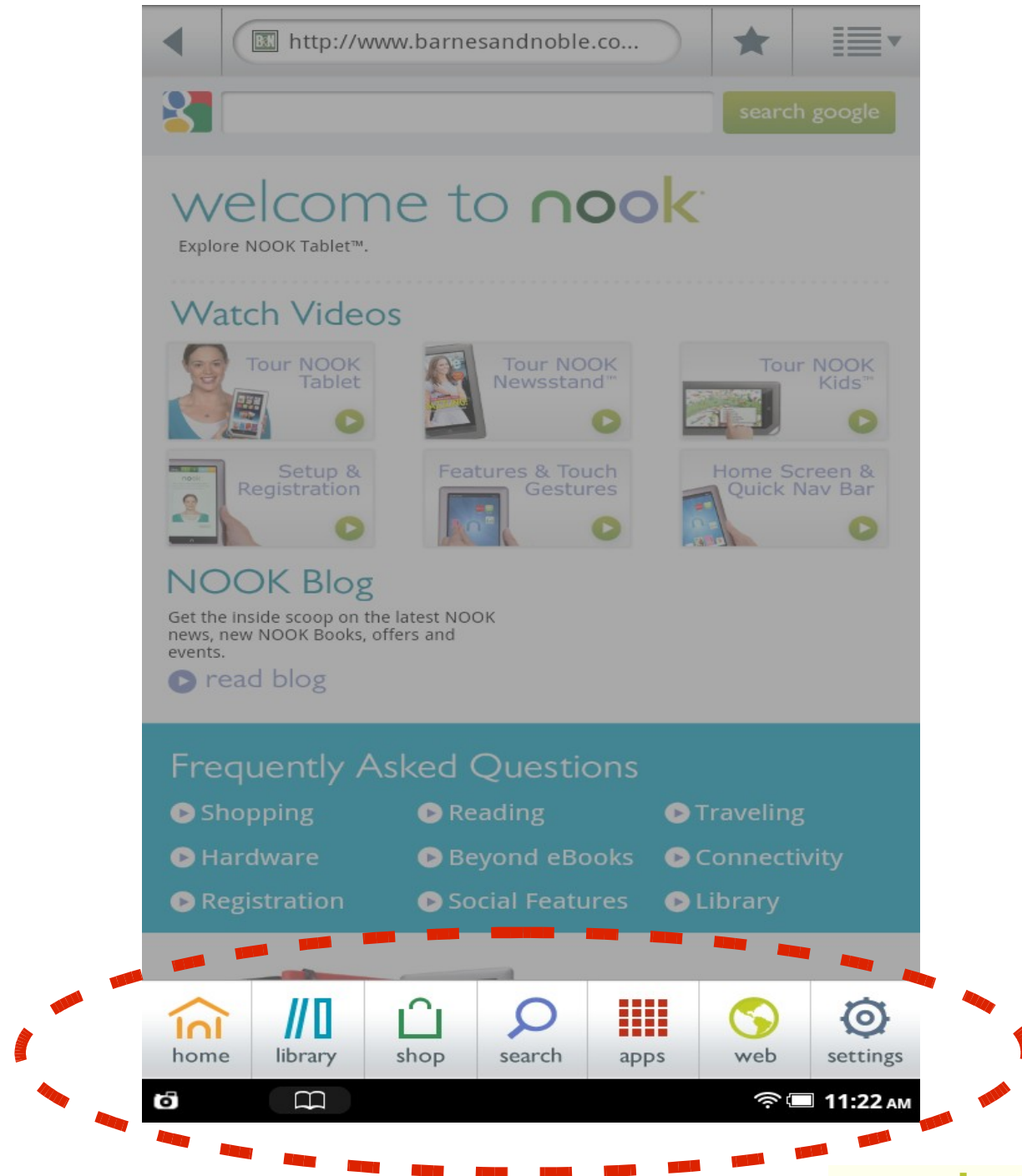
Introducing QuickNav

- Custom Platform UI widget provided for quick navigation. Also known as QuickNav bar, Navigation bar.
- Its purpose is to help a nook ® user navigate to following services and applications on the device.
 - Web
 - Shop
 - Home
 - Library
 - Settings
 - Search
 - Apps

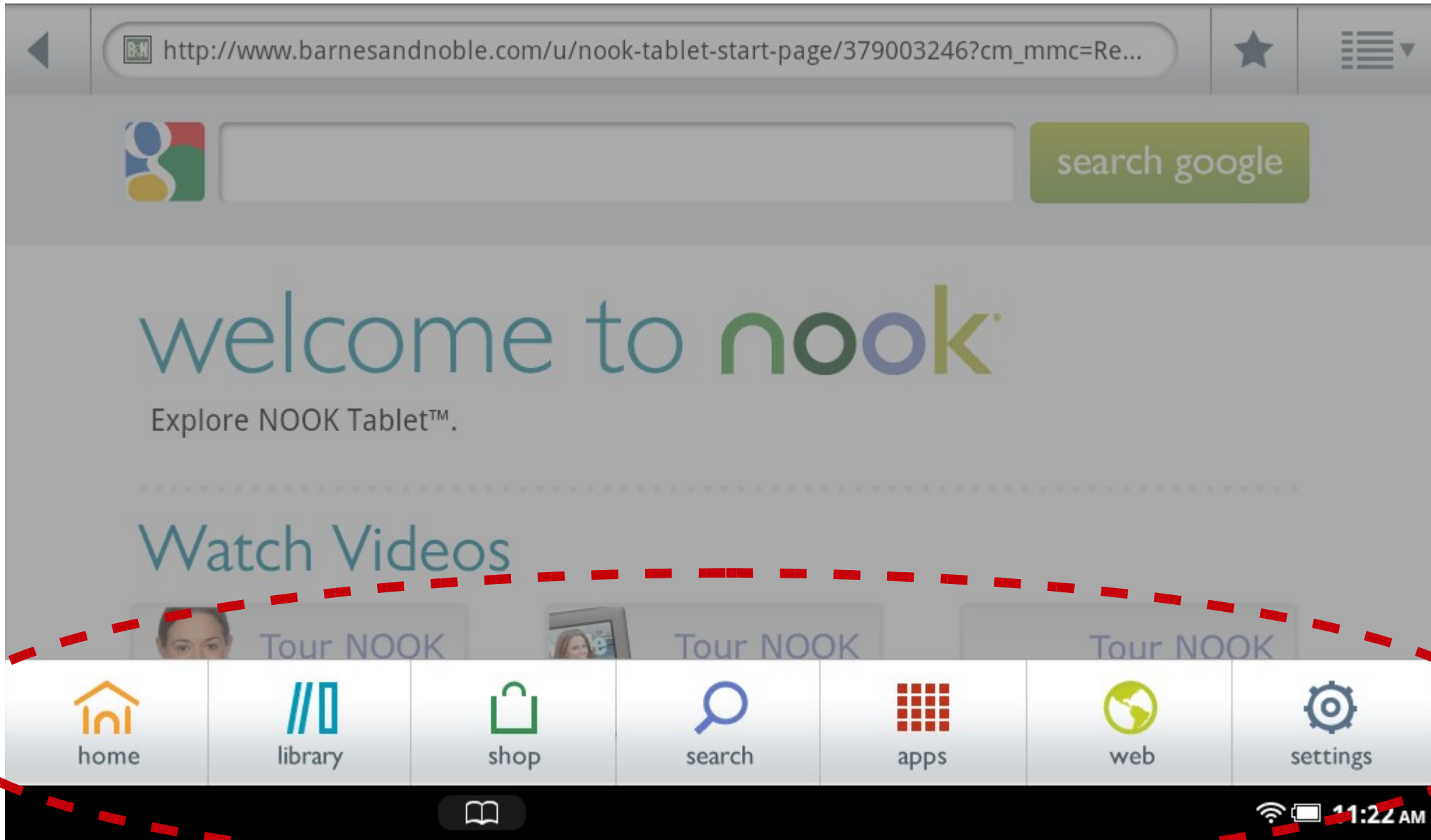
NOOK™ Tablet

Portrait mode

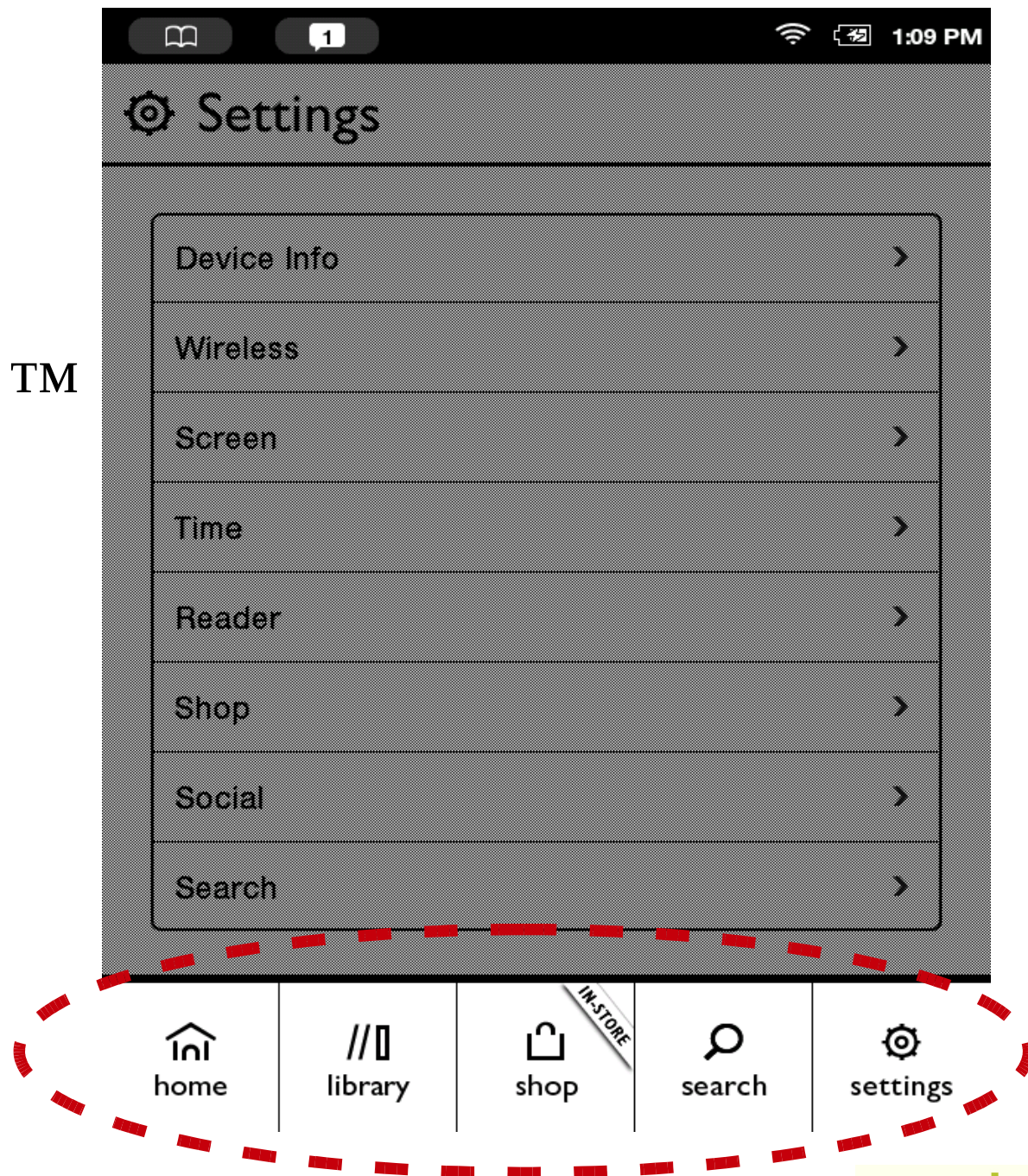
QuickNav UI



NOOK Tablet™ landscape mode QuickNav



QuickNav for NOOK Simple Touch™



Operation and Demo

- QuickNav launch and selection
 - QuickNav comes into view on click of the “n” key on the device.
 - A user can then press to select any of its tabs to launch a target application.
- QuickNav dismissal
 - If the user does not interact with QuickNav while it is visible, it fades gracefully out of view after 20 seconds.
 - A user can however click the “n” key while the QuickNav is visible to dismiss it and launch Home.
 - Another way to dismiss is to touch the part of the screen which is not in focus when the QuickNav is visible.

Design

- Motivation
 - To provide a quick, elegant, UI widget for navigation.
 - To *not* “create-a-clutter” or cause “visual-confusion” while a user interacts with it.
 - To be invoked on demand, and *not* be a permanent (visual) fixture on the screen.
 - To be “above-all” functional as provided; to be *not* user-configurable or user-programmable.
 - To dynamically adjust to device status related to services the UI provides.
 - To help be a productive widget for nook ® - user

Design ...

- QuickNav (and StatusBar)
 - QuickNav is an extension of Android StatusBar.
 - QuickNav is operational as long as StatusBar is visible and enabled.
 - When QuickNav is visible, the focus is on its View, the rest of the device's screen is veiled.
 - QuickNav's operational framework is actually pinned into Android's StatusBar framework.

Design ...

- During (nook ®) Android initialization,
 - StatusBar is constructed along with its internal components.
 - QuickNav is built, its framework callbacks are pinned using references to StatusBar internal components.
 - Intents for the tabs, methods to operate the QuickNav are configured and set; Touch event handlers and click event handlers for the tabs are established.

Design ...

- StatusBar (and its association with QuickNav)
 - StatusBarManager internal-app.
 - StatusBarManagerService is a helper module, providing a *starting point* for other components to *initiate* StatusBar operations.
 - StatusBar module is a *collection of commands* that can be enqueued and processed.
 - StatusBarService, a *Android Service*, processes the enqueued StatusBar operations ie notifications, QuickNav toggling, enabling/disabling ticker etc.

Design ...

- QuickNav
 - is designed to be a custom horizontal view, with individual tabs that are set to react to short-press and long-press touch.
 - is designed for the tab images to have state-drawables for normal, selected, enabled and disabled states.
 - is designed to redraw its view when device changes configuration, ie, its images are redrawn when the device goes into portrait/landscape mode.
 - is designed to be not operational when the StatusBar is disabled/not-visible.

Implementation

- **Files *modified***
- **PhoneWindowManager.java**
policy/src/com/android/internal/policy/impl/
- **StatusBarManagerService.java**
services/java/com/android/server/
- **StatusBarService.java, CommandQueue.java**
packages/SystemUI/src/com/android/systemui/statusbar/
- **StatusBarManager.java**
core/java/android/app/
- **IStatusBar.aidl, IStatusBarService.aidl**
core/java/com/android/internal/statusbar

Implementation ...

- **Files *added***
- **BNQuickNavbar.java**
packages/SystemUI/src/com/android/systemui/statusbar
- **common_quicknavbar_view.xml**
packages/SystemUI/res/layout
- **Support XML/asset files**
packages/SystemUI/res/

Implementation ...

- PhoneWindowManager.java
 - Has been modified to help initiate toggle QuickNav using the “n” key on the device.
 - “n” key also launches to Home screen and has other operations; these are carefully tied in to not disrupt sleep/suspend/wakeup operations of the device.
 - StatusBar and QuickNav are not visible until the device is unlocked. This implementation is also managed in this file.

Implementation ...

- **StatusBarManager.java, StatusBarManagerService.java (implements IStatusBarService)**
 - Add the operation toggleQuickNav().
 - Manage the flow of the calls from StatusBarManager application to the low level internal execution of actually toggling the QuickNav.
- **StatusBarService.java (extends Service, implements CommandQueue.Callbacks), CommandQueue.java (implements IStatusBar)**
 - Provide the bulk of implementation of constructing, operating the QuickNav.

Implementation ...

- **BNQuickNavbar.java**
 - Define the look, layout, tabs, Intents for QuickNav use.
 - Manage device configuration management details and touch based interaction to launch other applications.
- **common_quicknavbar_view.xml**
 - Define layout, image badges for QuickNav.

Thus we added/modified files to help implement the QuickNav in Android core UI framework.

OEM considerations

- NOOK Color™ and NOOK Tablet™
 - Operates in Portrait and Landscape mode.
 - QuickNav fades out of view after 20 seconds, after launch.
- NOOK Simple Touch™
 - Device operates in portrait mode; StatusBar is on top, however QuickNav on launch is visible from the bottom of the screen.
 - To release the QuickNav “n” key needs to be pressed – ie there is no timer to help it fade out of view.

Conclusion

- QuickNav is a *B&N nook* ® *widget* to help quick, convenient navigation to other applications.
- QuickNav is a StatusBar component; Its operations and design is custom fit into the StatusBar/StatusBar service framework.
- QuickNav is Platform UI framework component, with its own look and feel; Android core UI framework has been *enhanced and customized* to help launch, operate it for nook ® *tablets*.

QuickNav Trivia

- Originally we had a soft-button in the center of StatusBar to help launch/dismiss QuickNav.. but this was modified later.
- QuickNav does have hooks in its design for an external user-view/application to invoke it – not exercised at this time.
- In the very first version of nook ® (dual screen mode) the QuickNav had a scrolling view with more than seven tabs.

Q&A, Contact Info

Lalitha Krishnamurthy

lkrishnamurthy@book.com

lali.krishnamurthy@gmail.com

Linked In and Google+