

Modular Mobile Application Design

This article describes the development of the Minrva library app for Android phones. The decisions to build a native application with Java and use a modular design are discussed. The application includes five modules: catalog search, in-building navigation, a barcode scanning feature, and up to date notifications of circulating technology availability. A sixth module, Amazon recommendations, that is not included in the version of the app that was released is also discussed. The article also reports on the findings of two rounds of usability testing and the plans for future development of the app.

by Jim Hahn and Nathaniel Ryckman

Minrva Modules and Future-Proofing Mobile Apps

Minrva is designed as a modular mobile software application. This conceptualization is novel to the field of mobile computing in library settings, which generally has developed one-offs that run the risk of obsolescence over time. Native app development supports a module system that would not be a one-off but rather would grow in usefulness and services over the life of the project. The Minrva app we are developing is a test case in the modular design approach where each function of the application is a module that can be refreshed or added to as needed and in a collaborative manner. In this paper we suggest advantages to the native developer environment, particularly Java modules, while reporting through qualitative user study results how native components of mobile apps are useful in library settings.



Figure 1 - Minrva Conceptual Model

The app is comprised of modules that are configurable to local library user needs. Features of the app are not possible with traditional desktop computing. These services include data integration among the library wayfinding (GPS-like functionality), location-based recommendation, camera data and library OPAC data. The 1.0 release of Minrva, available on the Google Play website (<http://goo.gl/r7pyY>) includes five interactive modules. There is a sixth module (the suggest module) that is not available on Google Play at this time. The functions we studied include a barcode scanner module, a suggest module, a home module (main state), a search module, a wayfinding module, and a technology equipment availability module.

Scanner

The barcode scanner allows a user to scan a book with her phone and get a book summary. The scanner can also load book

summaries into the home module.

Suggest

The suggest module sends a request to Amazon for a list of ISBNs for items in the collection that are relevant to the scanned library item. The suggest module incorporates Amazon recommendation data into the user's stacks location and subject interest.

Home

The home module is the main state of the app, and lists the title, author, and call number location of the item. The home module will also present an image of the book cover with book summary. The home module will show the user current availability and display what departmental library the item is held within.

Search

The search module supports searching by title, subject, or author in the library catalog, and supports a list view of search results. Selecting a search result item will take the user back to the home module where they can get the detailed book item information.

Wayfinding

The wayfinding module presents the user with a map of the Undergraduate Library and also draws a red dot over the section of the library stacks where the desired item is located.

Technology Equipment Availability

The technology equipment availability module presents a list of the circulating technology equipment that is currently available for checkout in the library. The technology equipment availability module is organized by popularity (based on circulation history) and then lists the number of remaining items for checkout.



Figure 2 - Technology Equipment Availability Module

Consider the following use case in any library setting. The user searches for an item and is then guided to the book using the wayfinding module, which has displayed a map that shows the location of the item in the library book stacks. Once in the library book stacks, at her desired item, the student can access more information about nearby books, such as a book summary by taking a picture of the book's library barcode. The student can also learn the current availability for checkout of any circulating technology equipment by tapping the technology availability tab.

Design of Library Apps

There are three commonly used approaches to design of mobile apps: native, web based, or a hybrid approach of the previous two.

The native approach includes developing in compiled languages like Java in the Android platform, and Objective C in the iOS platform. Web based mobile apps make use of HTML5 and CSS and JavaScript to achieve an app-like interface and experience. jQuery mobile is a popular framework for developing web based mobile apps <http://jquerymobile.com/>. Hybrid approaches rely on using mobile frameworks like PhoneGap <http://phonegap.com/>.

We developed Minrva as a native app to future-proof the design. We chose to develop in Java in order to make use of Java's modular feature set. Each of the components and functions of the Minrva app was designed as its own module. These modules are designed for customization across other library settings, meaning that other library systems outside of the University of Illinois could use the architecture and web services for their environment. While such an initiative is not without local implementation work, we designed for sharing, and as such systems librarians and other professionals working in library systems should be able to adapt the framework for local needs.

Each of the Android modules is lightweight. They receive all data through a Wi-Fi, or cellular data connection such as 3G. Modules request data from RESTful Web services and receive JSON feeds in response. A typical RESTful request to one of the services that we have set up will look like this:

```
http://localhost:8080/MinrvaServices/LoanableTech
```

The JSON response returned from the request is as follows (abbreviated):

```
1 [ { "bibId" : "6399958",
2
3   "count" : "2",
4
5   "name" : "MacBook Pro ",
6
7   "rank" : "1"
8
9   },
10
11  { "bibId" : "6426672",
12
13   "count" : "4",
14
15   "name" : "Macbook Charger ",
16
17   "rank" : "2"
18
19   },
20
21  { "bibId" : "5241959",
22
23   "count" : "12",
24
25   "name" : "Laptop ",
26
27   "rank" : "3"
28
29   }
30
31  ...
```

In order to provide the JSON feeds, we have set up a Web application called Minrva Services. Minrva Services is a RESTful Web app written in Java that runs on Apache Tomcat 7 (see: Huang, Wu, Guo, 2009 for an example of a project using these tools [1]). Minrva Services is able to map HTTP requests to Java code and automatically serializes its response to a variety of formats by utilizing a dependency called Jersey (<http://jersey.java.net>). The format that the data is serialized to depends on the request header; thus, Minrva Services is an API that can be used to provide data to multiple types of software including iPhone apps, Android apps, and websites.

Each Minrva module receives data from a corresponding Minrva Web service. Each Web service is composed of a data access object, a data model, and a resource class. The resource class is responsible for mapping RESTful requests to the proper data access object. The data access object is responsible for retrieving relevant data from a variety of sources and packaging the retrieved data into data model classes. The data model classes are responsible for serializing the Web app's response into a variety of formats including: HTML, XML, and JSON. Minrva Services format responses into JSON when responding to the Minrva app requests because Minrva app modules request JSON in their request headers.

The data model used by Minrva Services to serialize the technology equipment availability module's data to JSON (abbreviated):

```
1 package edu.illinois.ugl.minrva.model.loanabletech;
2
```

```

3  import javax.xml.bind.annotation.XmlRootElement;
4
5  @XmlRootElement
6
7  public class Model {
8
9  private String bibId;
10
11 private String thumbnail;
12
13 private String name;
14
15 private String count;
16
17 private String rank;
18
19 public Model() {
20
21 super();
22
23 this.bibId = "";
24
25 this.thumbnail = "";
26
27 this.name = "";
28
29 this.count = "";
30
31 this.rank = "";
32
33 }
34
35 ...

```

With the aid of a Java library called GSON, the very same data model can be used by the Android Minrva app in order to parse the JSON feed created by Minrva Services.

The data model used by the Minrva app to parse the technology equipment availability module's data feed:

```

1  package edu.illinois.ugl.minrva.loanabletech;
2
3  public class Model {
4
5  private String bibId;
6
7  private String thumbnail;
8
9  private String name;
10
11 private String count;
12
13 private String rank;
14
15 public Model() {
16
17 super();
18
19 this.bibId = "";
20
21 this.thumbnail = "";
22
23 this.name = "";
24
25 this.count = "";
26
27 this.rank = "";
28
29 }
30
31 ...

```

Note from the preceding two examples of code that the Java data models used in the Android code are the same models used in the Java servlet running in Tomcat 7. Each of the modules in the Minrva application was designed to gather or provide unique information about an item in the collection (book, media, circulating loanable technology) by using a unique identification number

(e.g. internal bibliographic identification number). Only one unique identification number is loaded into the Minrva app at any given time. When the module runs, it will send the unique identification number to the RESTful Web service. The RESTful Web Service will then send back information to the module that will then be displayed on the phone. In order to avoid dependencies between individual modules, all of the modules communicate with one another indirectly through the unique identification number. In addition, the data retrieval process is completely independent of the Android app because the RESTful Web Service is responsible for retrieving the data. As an instructive primer on RESTful services see Richardson & Ruby, 2007 [2].

Methods for User Study

Rapid prototyping studies are shown to be an effective method to gather user preferences throughout the design phase such that products can be delivered in a shorter development time as well as encompassing desired functionality of end users. The rapid prototyping method described and tested by Jones & Richey (2000) is the basis for the use study results presented here.[3]

This study encompasses qualitative data collection that used detailed interview and observation. This methodology allowed us to understand areas of use and fail points for each module, and to iterate designs before the initial 1.0 release of the app. The use study took place in the University of Illinois library, in both the Main building and the Undergraduate Library building. The study had a total of 10 test users. See the appendix for interview questions and observation log.

Test users were given a \$20 gift card for pizza if they agreed to participate in the user study. We observed the following actions of all users: students as they scanned an item in the collection, searched for an item with the search module, located an item in the stacks using the wayfinding module, received Amazon suggestions from the suggest module, and discovered what circulating technology equipment was available in the library from the technology equipment availability module.

Thematic Use Results

Thematic use results are the aggregate usability trends over all test users of the app. Themes of use represent preferences for desired functionality and overall Minrva app performance. These include:

- Intuitive interactions among the modules,
- Library-wide functionality,
- A deeper integration and emulation of the library catalog.

These themes can be applied across all modules since each participant made use of each of the functions of the Minrva app.

Intuitive Interactions

Users of the Minrva module made many comments that showed us the need for intuitive interactions. This manifests in two modules especially. The scanner and wayfinding modules require further information to make them easier to understand. Since the map may not display exactly in the position the user is facing, it requires a compass-like function to help the user get to the location in the stacks. Secondly, if the scanner is unfamiliar to the user, it will require helpful graphics which illustrate a proper scan and an example library barcode (see Figure 3). Users expected seamless integration of functionality among the modules. Students expected that tapping an element in one module, like the search module, should have the effect of loading this item (book data) into the home module.

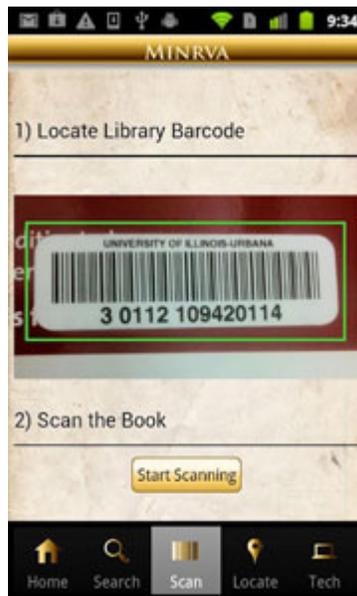


Figure 3 - Scanner Module with usage indicator

Library-wide functionality

Library wide functionality was added to the Minrva app based on the use studies. We added the ability to search library collections within the disparate library units at the University of Illinois. The users we spoke with also desired a consortia search. Before we uploaded the Minrva app to Google Play app store, we incorporated a consortia search (to search multiple departmental holdings) into the VuFind module – this necessitated a reworking of the unique identifier we used to pass data among the application modules. We re-worked the data architecture while still maintaining adherence to the conceptualization of one unique identifier passed among the component modules of the application (see Elmasri & Navathe, 2006, p. 704, regarding the importance of unique object identity) [4]. Figure 4, below, shows the Home module with a library location outside of the Undergrad Book Stacks, which uses the re-designed consortia search.



Figure 4 - Home Module

Another library-wide functionality request was the ability to use maps for multiple library locations, not just the Undergrad Library. Directional help in other highly circulating collections such as the Grainger Engineering Library, Main Stacks, the Music and Performing Arts Library and others is planned for integration into the wayfinding module.

OPAC Integration

The thematic use result of deeper integration and emulation of the online library catalog is indicative of the desire for the Minrva app

to have a similar element set to that of the catalog. This includes showing item availability, call number, and library location. User preference for deeper emulation of the online catalog is manifested in the desire among students to be able to login to accounts, renew items, as well as request items from the library catalog. Test users want everything the catalog can do, plus all of the features that a mobile device is capable of – e.g., indoor positioning systems akin to GPS functionality. There were also requests for the Amazon suggestions to then be pushed into the catalog search such that the catalog would filter out any Amazon recommendations that were not in the library collection.

While a rapid prototyping methodology is not statistically significant in a quantitative sense, we were able to collect a set of rich qualitative use data on the modules' functionality and interaction among the features studied. These use results informed the 1.0 release now available on the Google Play app store. During our study we also uncovered new modules to develop.

Future Design Iteration

The use studies informed the design iterations planned for the next release of Minrva. Based on user feedback we will design a patron login feature so that students are able to renew items and view their library account. Test users also requested a favorites module; they asked that we be able to retain their favorites in the app and work on implementing a wish list of items inside of the mobile app. At the time of this article our staff has investigated posting to a user's Facebook timeline from the app, much like the popular Goodreads (goodreads.com) mobile application. This feature may be implemented in the 1.1 release of Minrva.

Students asked that we design a module that could function similarly to the technology equipment availability module, where the Minrva app can show the user how many group study rooms are immediately available in the library. Students also made a feature request to generate citations of the book data from the home module, and then send these citations to their student email account. Further, students would also like to have additional library information included in the Minrva app, like library hours.

Our campus office of technology management did not allow the Minrva app to be uploaded with the suggestions module. The suggestion module was retrieving information from Amazon's website and re-using this information inside of the app, a use that currently is not an allowed according to the campus interpretation of Amazon's copyright claim. We are exploring the possibility of getting written consent from Amazon to make use of their website for ISBN referrals and this may be incorporated again into a future version of Minrva.

Conclusion

The value of native application development is shown by our thematic use results, which indicate that users do find that a modularly designed native application delivers helpful and useful assistance at point of need in library settings. Our plans to integrate desired functionality demonstrate the value of using this modular concept to future-proof the app. Our design approach can accommodate students' ideas as we build out additional modules to the Minrva mobile app.

We hope other library staff (coders like you) would make use of the existing architecture to implement any of the core Minrva services in their local setting. We designed for this purpose. It is our additional hope that library staff might contribute new modules that they have the resources and time to implement. In sharing these modules with the Minrva project they may be implemented in any library using the Minrva app. This is the key advantage of the app concept: modular collaborative design can pool collective advances from the community of Minrva developers. The modules are available at Github: <https://github.com/minrva>

At the University of Illinois, through a variety of internal and national grants we are actively researching the following features as project next steps:

- Port the Minrva app to iOS systems
- Implement Optical Character Recognition and Augmented Reality services for “augmented book stacks” browsing in the scanner module
- Implement an Indoor Positioning System with WiFi data in the wayfinding module
- Incorporate article search into the search modules, along with catalog requests and a favorites feature that posts to a Facebook timeline
- Make available additional campus library maps in the wayfinding module
- Design a group room availability and scheduler as a new module
- Design a due date reminder push notification feature

We anticipate securing additional grant funding so that we can place Minrva services into public and academic libraries across the

United States at no cost to any individual library. We are especially interested in working with traditionally under-served areas such that we could place next generation technologies into communities that may not traditionally have been able to design library mobile apps. A Pew Internet and American Life report, *Smartphone Adoption and Usage* [5] underscores the point that smartphone adoption is highest among non-whites, noting that “44% of blacks and Latinos are smartphone users,” indicating that mobile first initiatives can make inroads to serving traditionally under-served populations. We believe that as a special advantage to mobile apps, these tools may increase library use among user groups that have not been engaged with or interested in traditional library services.

To move this initiative forward we now require the collaborative efforts of library coders. The Minrva Project’s Github space supports collaborative coding. While such a call to action for librarians has antecedents in open source projects, the library world has yet to see a collaboratively maintained and enhanced mobile software application. The true value for this project is that all libraries and library users devoting time to this effort will see their work power an app that can become ever more useful and feature rich as the project grows.

Acknowledgements

The authors wish to acknowledge the Research and Publication Committee of the University of Illinois at Urbana-Champaign Library, which provided support for the completion of this research.

Notes

[1] Huang, Y.M., Wu, D.F., Guo, Q. (2009), “Build a RESTful Web service using Jersey and Apache Tomcat,” <http://www.ibm.com/developerworks/web/library/wa-aj-tomcat/>.

[2] Richardson, L. & Ruby, S. (2007), *RESTful Web Services*, O’Reilly, Sebastopol, CA.

[3] Jones, T. & Richey, R. (2000), “Rapid Prototyping methodology in action: a developmental study,” *Educational Technology Research and Development* 48, 63-80.

[4] Elmasri, R. & Navathe, S. (2006), *Fundamentals of Database Systems*, Addison Wesley, Boston, MA

[5] Smith, A. (2011), “Smartphone Adoption and Usage,” *Pew Internet and American Life*, <http://pewinternet.org/Reports/2011/Smartphones.aspx>

See also

The Minrva Project: library technology prototyping initiative

<http://minrvaproject.org/>

Appendix

Investigator Log

Please describe any previous experience finding items in the Undergraduate Library?

Scanner Module

Does the scanner work in the undergrad library bookstacks?

Do students scan the library barcode?

How do students react to the scanning functionality?

Do students select any of the elements displayed after scanning?

What unexpected things occur?

Search Module

How do student use the search module?

Do they type in any free text for searching?

What unexpected things occur?

Suggestions Module

How do students use the suggestions?

Do students select any of the elements displayed?

Wayfinder Module

How do students react when the wayfinding application does not work as they expect?

What unexpected things occur?

How do students make use of the recommendation features?

Are students able to locate the items that are on the shelf?

Note any additional observations of student use of the Minrva software:

Interview Questions

How easy is the application to use? What would make it easier to use?

What was hard to do with the application?

What was confusing?

What was surprising?

What do you wish you could have done with the application while you were using it?

How useful do you find the application? What would make it more useful?

Would you recommend it to friends?

What do you actually want from a library app? Is there something else that should be here that is not here?

Is this application a worthwhile tool for the library to develop?

About the Authors

Jim Hahn (jimhahn@illinois.edu) is Orientation Services and Environments Librarian, Undergraduate Library, University of Illinois.

Nathaniel Ryckman (nathanielryckman@gmail.com) is Visiting Research Programmer, Undergraduate Library, University of Illinois.

Subscribe to comments: [For this article](#) | [For all articles](#)

This work is licensed under a Creative Commons Attribution 3.0 United States License.

