

An Investigation of the Development of Mobile Business Intelligence

Sathyanath Lappasi Ramamoorthy, Jagdev Bhogal and Sharon Cox

Faculty of Technology, Engineering & the Environment,

Birmingham City University

Corresponding e-mail sharon.cox@bcu.ac.uk

Abstract: *The smartphone market is expanding rapidly as smartphones become ubiquitous in the workplace. In this paper, recent developments in mobile business intelligence are considered, including a comparison of development platforms, market trends and the future growth of mobile business intelligence. The paper also outlines the design of a mobile business intelligence application. The application retrieves data stored in an SQLite database and enables users to visualize and drill-down into the data in order to explore sales intelligence data, trends, growths and market analogy. Key decisions to be taken in the design and development of mobile business intelligence applications are identified to assist developers of future business intelligence applications.*

Keywords: *mobile business intelligence, BI, Android, iOS, native application development, cross-platform development, SQLite.*

Introduction

Mobile business Intelligence (mobile BI) is a technology providing access to business data anytime and anywhere. By getting access to business data, users can perform visual analytics to make quick decisions to support greater competitive advantage in a rapidly changing business environment. As smartphones have become more popular and affordable they allow more people to consume business intelligence applications. Mobile BI offers greater flexibility to staff in at all levels in the organization, from senior executives to sales representatives, by providing access to real-time business data whilst on the move. This means that staff can keep informed about recent trends to respond to customer queries and to draw interests of more stakeholders (Accenture, 2012).

This paper investigates the recent trends and developments in mobile business intelligence (BI) and outlines the development of a prototype mobile BI

application in a cross-platform environment with embedded database connectivity. Section two reviews the literature on understanding the technology behind mobile BI and its development trends. The review also compares development platforms and technologies such as PhoneGap and embedded database such as SQLite. Section three explains the development of an enhanced BI dashboard strategy and the methodology used, showing how the data was collected and used in an SQLite database. Section four discusses the development of the application to enable a user to better understand how mobile BI application works and how the data are stored and retrieved from the SQLite embedded database. The development of the app incorporated the use of Integrated Development Environment (IDE) tools, Android Software Development Kit (SDK) and Java Development Kit (JDK). The paper concludes by providing recommendations to be considered when developing mobile BI applications.

Mobile Business Intelligence

The term BI was first coined by the Gartner group in the mid-1990s, but the concept is much older in that it has its roots in the management information systems of the 1970s (Turban *et al.*, 2007). BI can be regarded as a decision support system that consumes and converts business data into useful information that can be used to create new business opportunities. Vercellis (2009) defines BI as the ability to exploit the available data to generate significant knowledge and information useful for complex decision-making processes. The term BI is used by software vendors to characterize a broad range of technologies, software platforms, specific applications and processes (Vitt *et al.*, 2002). BI systems combine databases, applications and analytical tools to perform analysis on data.

Business intelligence applications run on desktop machines that retrieve and interpret data into information in the form of visual charts and graphs to give a unified view of data to the user. However, work-related activities are increasingly performed outside the office. The workforce needs access to real-time data and needs to keep informed about trends in business performance to make better business decisions quicker and faster whilst on the move (Brockmann *et al.*, 2012).

Mobile BI extends desktop-based BI applications for use on a mobile device (Information Builders, 2011) and increases an organization's capability to deliver business data in a timely manner to anyone regardless of wherever they are able to access the data (Yellowfin, 2011). Mobile BI applications retrieve historical data from repositories and present the data to a user, in the form of charts and graphs, on a mobile device to highlight hidden trends.

Business Intelligence Dashboard Strategy

BI applications are used to get a deeper insight into historical data in the form of dashboards, key performance indicators (KPIs), scorecards and graphs. This analysis can also be conducted on mobile devices by creating BI applications that query data from database and produce charts and graphs to the user to perform ad hoc visual analytics (Rouhani *et al.*, 2012). Mobile BI delivers BI information in the form of dashboards, KPIs, scorecards and alerts to mobile devices. A well designed dashboard is important as with better visualization, more information can be conveyed from the available data in repositories. When designing mobile BI application, the users, style, content and context need to be considered (Pandian, 2012).

Eckerson (2011) suggests that BI applications can be deployed on mobile devices either by the development of native mobile applications or using hybrid cross platform mobile applications. These applications are discussed in the following sections.

Native Mobile Applications

A native mobile application is an application designed specifically to run on a target mobile operating system. For example, to develop an application for Apple iPhone, Objective-C programming language must be used, similarly C# is used for Windows devices and Java is used for Android mobile devices. Native applications of iOS (the operating system used on Apple devices) cannot run on Android devices and vice versa. Native applications take full advantage of the features of a mobile device such as camera, gyroscope and accelerometer,

however, the development of native applications requires a deep platform knowledge of specific operating systems (McGuirk *et al.*, 2012). Native applications are therefore complex and more difficult to develop as each application has to be adapted to run on different mobile devices with different operating systems.

Hybrid Cross-Platform Mobile Applications

Cross-platform development is the development of an application that can operate on multiple platforms. Hybrid cross-platform mobile application development connects mobile web development to native development, enabling the use of web mark-up languages, such as HTML, CSS for layout and JavaScript. It extends native device capability to the mobile web browser (McGuirk *et al.*, 2012).

Platform Selection

Three parameters need to be considered when selecting a mobile application development platform: Firstly, the market share of the operating system, as this will determine the potential market for the application. Secondly, whether the development software tools and source code are available in the open source domain. Thirdly, the predicted future trends and growth potential. Globally, half of the smartphone sales are Android-based devices (Gartner, 2013). In the year ending 2012, Android led the market with a market share of 49% followed by Apple's iOS at 19% and Research In Motion (RIM) at 13%. The Android operating system comes under the open handset alliance and offers developers freedom to develop applications in a flexible manner. It is an open source platform and is free for the developers to use.

PhoneGap Architecture

PhoneGap is an open source development framework for the development of hybrid cross-platform mobile applications. The applications are hybrid in that they use web markup languages but also access the native application programming interface (API) of the mobile device. The API is a set of standards that the software must follow to establish an access to the other software program. It makes a request to access the other resources of that particular software program (3Scale, 2011). In this research, a mobile BI was developed that accesses the native database storage API to store the data to be displayed on smartphone screen.

The PhoneGap API requests data from an SQLite database through the native operating system API to access database that is embedded in the Android operating system. SQLite is an embedded relational database management system. It is embedded inside the application it serves and its codes are wrapped as part of the program that hosts it (Allen *et al.*, 2010). It is widely applied in embedded database management environment such as mobile devices, tablet computers and other handheld devices. SQLite supports all the relational database SQL syntax. The database is available on all Android devices and does not require database setup or administration (Vogel, 2013).

The advantages of SQLite include:

- It is a light weight compatible database that meets atomicity, consistency, isolation, durability (ACID) norms for transactions.
- No server configuration, scripting and installation are required.
- Seamless create, retrieve, update and delete (CRUD) operations are possible.
- The SQLite database is contained in a file format.

- It extends support to access the content offline.
- PhoneGap accesses the SQLite database through its storage API.

SQLite can be accessed through a sample database open call as shown in Figure 1.

```
var dbShell =
window.openDatabase
(database_name,
database_version,
database_displayname,
database_size);
where
window.openDatabase - returns a new
database object
name - database name given to the
database in the application
version - represents the database version
display_name - the display name to be
shown
size - size to be assigned for the database
in bytes
```

Figure 1: Open Call to Access SQLite

Oracle Development Tools

The Oracle Application Development Framework (ADF) uses Java and HTML5 to build mobile applications. The ADF allows developers to develop single application that can run on multiple platforms through HTML, CSS and Javascript in conjunction with Java as application logic to communicate to native device functionalities such as database, camera, GPS and more (Oracle, 2011). This enables developers to create applications for both Android and iOS with single one time code. As the code is applicable to both mobile operating system platforms it eliminates the need to interpret and translate the code to a device specific operating system platform.

OBIEE is an enterprise BI tool from Oracle which is built extensively on OBIEE 11g platform (Oracle, 2011). The data repository of OBIEE applications

resides in various levels such as the client level, server level and middle tier (or an additional server level). It is optimized to meet the needs of mobile BI and enables security models such as secure sockets layer(SSL), single sign-on(SSO) and virtual private network (Oracle, 2011).

Development of Mobile Business Intelligence Application

The Android operating system was chosen as the development platform for the development of a mobile BI application as it is an open source platform. This decision was taken because native mobile application development requires prior knowledge on the native programming languages and is a time consuming process. The application developed is therefore not multi-platform; it cannot function on different mobile operating systems.

The PhoneGap mobile architecture was chosen due to its extended multi platform support. It is an open source framework that works with Eclipse IDE for Android mobile BI application development. The underlying software that drives PhoneGap is the Apache Cordova. The latest version, PhoneGap version 2.5.0 was used.

The data collected for use in the mobile BI development originates from Gartner mobile OS sales report 2013. The data refers to sales figures for operating systems over a number of years. The quantitative data was identified and consolidated using Microsoft Excel spreadsheet. Further quantitative analysis including a forecast calculation was performed using Excel functions to determine the values for future year's sales. A sample of 2010 data was used to build the BI dashboard, to study how the mechanism functions to determine the key resources required to build a simple BI dashboard.

The agile development methodology was used to develop the application. Rahimian and Ramsin (2008) suggest that the agile development methodology is the most suitable for mobile application development. The agile methodology offers greater software productivity and flexibility enabling the development process to adapt to changes in requirements. It enables the rapid development and deployment of fully functional applications in a relatively short time span, which is vital in a fast growing market. The following sections outline the design and development of the mobile BI application.

Mobile BI Architecture

A generic mobile BI architecture is shown in Figure 2. The data repository layer contains data that are stored in the database and user authentication credentials such as usernames and passwords. As with desktop-based database administration, it is possible to restrict and grant access to specific users. It offers authentication services to web services generated by the client and it will log information, deterring potential intruders from accessing data when the validation fails (Sajjad *et al.*, 2009).

Mobile BI systems actively connect through the Business Logic Layer (BLL). The BLL offers services to carry out specific sets of calculations according to the client's requirements. There are several layers of authentication and transaction processes within this layer. When the client web service raises a request, it generates the request in the form

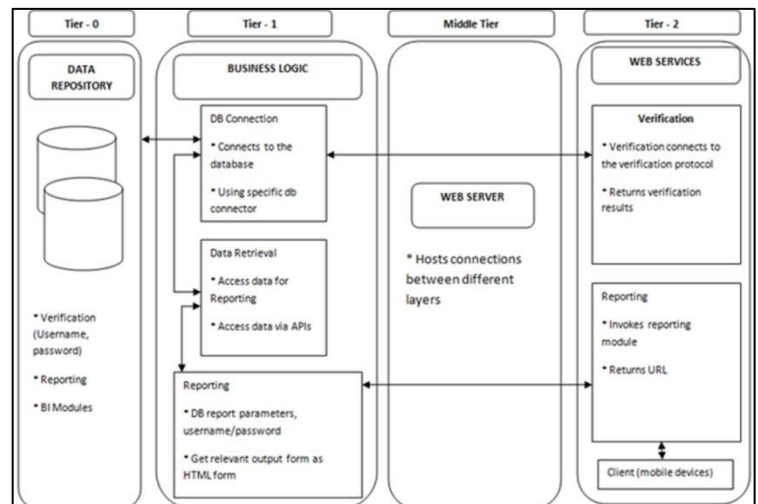


Figure 2: Generic Mobile BI Architecture (Sajjad *et al.*, 2009)

of query to send it to the database. During this request generation, a unique key is also generated to authenticate the user before accessing the data. A URL is generated for this specific output file which is then passed on to the web service. The web service shows the output file by receiving the URL and showing it in many forms such as graphs and charts on the smartphone screen (Sajjad *et al.*, 2009).

The web server (also referred to as the application server) acts as a bridge to connect web services to the other layers. It hosts connection to the different layers present within the mobile BI architecture. This layer also establishes the firewall and other security applications that will eliminate potential threats and unwanted intruders into the system.

Proposed Mobile BI Architecture

Generic frameworks are most appropriate for industrial transactions which involve large databases and complex business logic to perform calculations (Sajjad *et al.*, 2009). The application developed is intended for pure educational purpose as to learn what it takes to design a mobile BI

mechanism. There are therefore differences between the generic architecture and the application developed in this project (Figure 3). The application developed uses native resources to carry out BI tasks. The data request sent to the SQLite database links the request in the form of queries. The data are retrieved and sent to the web browser in turn the web browser executes the file.

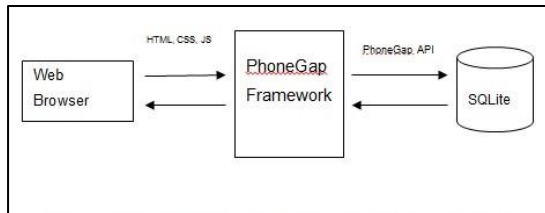


Figure 3: Mobile BI Application Web Browser Interfacing Layer

The application developed stores the mobile operating system data collected. All the transactions take place within the mobile device. Hence the application can run offline without any disruptions. The reports and dashboards generated are stored locally on the device for offline analysis. Mobile applications can leverage native storage for smaller data transactions.

SQL Query Execution Method in Mobile BI Application Developed

The mobile BI application uses the SQLite database for data transactions. The execution method happens in a sequential manner as shown in Table 1.

Step 1	The application loads up from the user action from menu.
Step 2	It shows the mobile BI screen which allows the user to select the required year and quarter from the dropdown menu.
Step 3	As an example, user selects the year '2010' and the quarter 'Q1'.
Step 4	It triggers Javascript API to interact with native storage API to

	process the request.
Step 5	The SQL statement is executed to retrieve the data for the year 2010 in relation with the quarter Q1 which was created in the database.
Step 6	Whenever the 'database' object is called, its call back method will also be called with 'SQLTransaction' object.
Step 7	The database transaction is built by calling the 'executesql' method to perform the action.
Step 8:	When called, the executesql method of SQLTransaction will invoke its callback function with 'SQLResultSet' object

Table 1: Steps in SQL Execution

Figure 4 shows the connections with the storage component. When the user submits the request, the execution of SQL query takes place that returns with the data to the front-end web application. The execution of queries takes place involving javascript APIs with PhoneGap storage plugins. It facilitates the communication to interpret and submit the request and retrieve the data from the native component

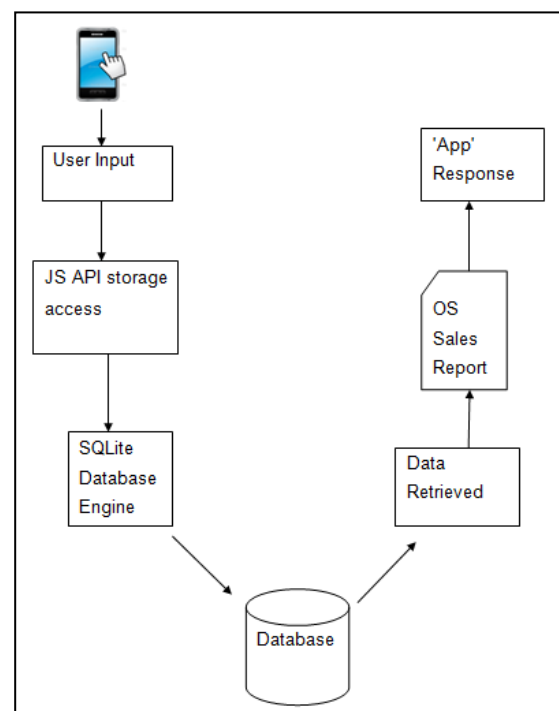


Figure 4: Mobile BI Application System Design

Android Manifest.xml

The development of the Android BI application must have an Android Manifest, an xml file in its root directory. This file contains the essential information of the application. The underlying Android core system must have access to this file before the Android device can start running the codes of the application. The core aspect of BI is its reporting functionalities to achieve data insight. The application used javascript libraries for jQuery and jQuery.flot.categories.js from JS library file to plot the data. It allows the developer to customize bar graph such as alignments and width. The application was simulated on the Android Virtual Device (AVD). Figure 5 shows the result from the 'executeSql' statement.

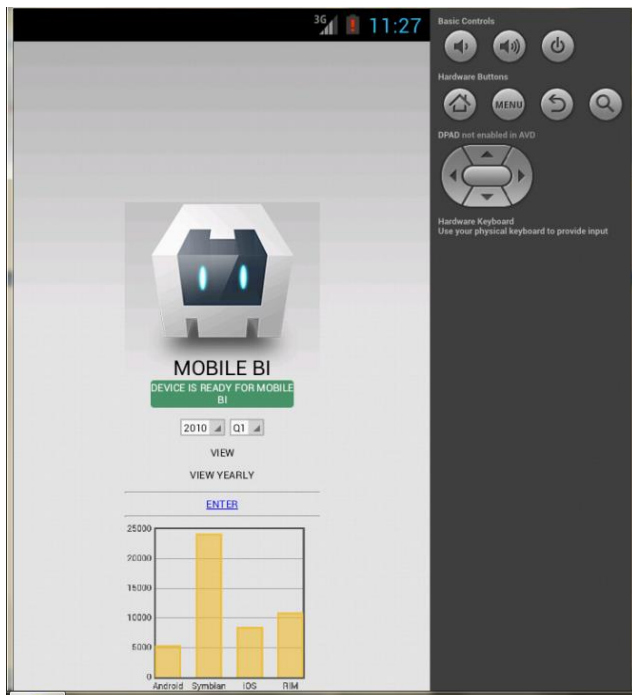


Figure 5: Mobile BI Application Simulated on Android Virtual Device

Conclusions

The mobile BI application developed is an Android-based application, using cross-platform web-based technology, SQLite storage, APIs and a mobile framework architecture. The SQLite database was sufficient to store the sample data and to populate the graph in Figure 5. More complex tables can also be created using SQLite to offer more granular level of detail to perform BI operations. Further development is needed to enable the application to perform visual analytics to understand market share of each operating system, quarterly sales, half yearly and annual sales by year.

From development of this application, key decisions to be taken in the design and development of a mobile BI application have been identified as:

- The choice of development platform: This needs to consider the customer requirements and the implementation platform (for example, iOS, Android or both).
- Native or web-based development. This will depend on the functional requirements and target audience for the application. For example, if the application is specifically built for gaming purposes, a native development approach is needed for the device specific implementation.
- Functionality and target audience. BI applications need to cater for the different needs of different audiences. For example, a senior executive would need to analyse regional sales whereas a line manager would require real-time reports from the production floor.

The mobile BI application developed demonstrates the ability to provide executives with BI on mobile devices. However, two key challenges remain: how to effectively downsize the design strategy of BI applications for mobile devices and how to choose the most appropriate

visualization method visual to display data on the small device screens.

Technological developments enable executives to gain remote access to data, however, remote access to meaningful information is now a priority. As the demand for mobile access to information continues to increase, further work is needed to improve the development of mobile BI applications to enable a seamless transition from the desktop applications to mobile devices. The development of mobile BI application offers the potential to provide decision-makers with consistent reliable information, irrespective of location or device used. Further research into the effective delivery and visualization of data on mobile devices is needed to enable mobile BI to achieve the potential of seamless information provision.

References

- 3Scale (2011) 'What is an API? Your Guide to the Internet Business (R)evolution', <http://www.3scale.net/wp-content/uploads/2012/06/What-is-an-API-1.0.pdf>, [accessed 2 April 2013].
- Accenture (2012) 'Business Intelligence is Going Mobile: Are You Ready?', <http://www.accenture.com/us-en/Pages/insight-business-intelligence-going-mobile-summary.aspx>, [accessed 18 January 2013].
- Allen, S., Graupera, V. and Lundrigan, L. (2010) *Pro Smartphone Cross-Platform Development: iPhone, Blackberry, Windows Mobile, and Android Development and Distribution*, Apress, New York.
- Brockmann, T., Stieglitz, S., Kmieciak, J. and Diederich, S. (2012) 'User Acceptance of Mobile Business Intelligence Services', *Proceedings of the 15th International Conference on Network-Based Information Systems*, 26-28 September, Melbourne, pp. 861-866.
- Eckerson, W. (2011) 'A Complete Primer to Mobile BI', *Beye Network*, <http://www.oracle.com/us/solutions/business-intelligence/oracle-mobile-bi-ebook-ds-501071.pdf>, [Accessed 8 October 2013].
- Gartner, (2013) 'Gartner Says Actionable Analytics Will Be Driven by Mobile, Social and Big Data Forces in 2013 and Beyond', <http://www.gartner.com/newsroom/id/2332515>, [accessed 20 January 2013].
- Information Builders (2011) 'Solutions for Mobile Reporting: Embrace the Mobile Channel without Complexity', <http://www.informationbuilders.co.uk/solutions/mobile>, [accessed 28 January 2013].
- McGuirk, C., Pekala, T., Petrin, J. and Renardi, E. (2012) 'Choosing the Right Mobile Development Method', *RDA Corporation*, http://www.rdacorp.com/wp-content/uploads/RDA_Mobile_Whitepaper_1.pdf, [accessed 2 March 2013].
- Oracle (2013) 'Oracle: Big Data for the Enterprise', *Oracle White Paper*, <http://www.oracle.com/us/products/database/big-data-for-enterprise-519135.pdf>, [accessed 6 October 2013].
- Oracle (2011) 'Develop Mobile Applications with Oracle ADF Mobile', *Oracle Datasheet*, <http://www.oracle.com/technetwork/developer-tools/jdev/adf-mobile-development-129800.pdf>, [accessed 12 February 2013].
- Pandian, M. C. (2012) 'Getting Started on Mobile BI, 4 Parameters to Consider', *Hexware Technologies Blog Network*, <http://blogs.hexaware.com/business-intelligence/getting-started-on-mobile-bi-4-parameters-to-consider/>, [accessed 6 October 2013].

Rahimian, V. and Ramsin, R. (2008) 'Designing an Agile Methodology for Mobile Software Development: A Hybrid Method Engineering Approach', *Proceedings of the Second International Conference on Research Challenges in Information Science*, 3-6 June, Marrakech, pp.337-342.

Rouhani, S., Asgari, S. and Mirhosseini, S. V. (2012) 'Review Study: Business Intelligence Concepts and Approaches', *American Journal of Scientific Research*, **50**, pp. 62-75.

Sajjad, B., Mir, A., Khawar, A., Bashir, F. and Tariq, A. (2009) 'An Open Source Service Oriented Mobile Business Intelligence Tool (MBIT)', *Proceedings of the International Conference on Information and Communication Technologies*, 15-16 August, Karachi, pp. 235-240.

Turban, E., Aronson, J. E., Liang, T. P. and Sharda, R. (2007) *Decision Support and Business Intelligence Systems*, Eighth Edition, Prentice Hall, London.

Vercellis, C. (2009) *Business Intelligence: Data Mining and Optimization for Decision Making*, Wiley, Chichester.

Vitt, E., Luckevich, M. and Misner, S. (2002) *Business Intelligence: Making Better Decisions Faster*, Microsoft, Washington.

Vogel, L. (2013) 'Android SQLite Database and Content Provider – Tutorial', <http://www.vogella.com/articles/AndroidSQLite/article.html>, [accessed 29 March 2013].

Yellowfin (2011) 'Mobile Business Intelligence and Analytics', <http://www.yellowfinbi.com/Document.i4?DocumentId=99556>, [accessed 29 January 2013].