UDC 004.724

Nooh Taha Nasif

ALNKHBA University College, Republic of Iraq-Baghdad

DEVELOPMENT OF A HYBRID MOBILE APPLICATION FOR ANDROID WITH THE FUNCTION OF PROCESSING AND SCANNING OF BAR-CODES

In this work the process of hybrid mobile application for Android operation system with new function of processing and scanning of bar-codes development is presented. Theoretical fundamentals are considered, WebSphere Commerce architecture of corporate application providing a high level of scalability, productivity, safety reliability and controllability are analyzed. WebSphere Commerce is briefly presented with the help of programming model, mobile platform, bar-code specification and used libraries. WebSphere Commerce usage for electronic commerce applications creation significantly shortens time of developing. Creation of new pages with usage of JSP technologies is simple. However, it is difficult to tweak these pages. Tweak tool, provided by Rational Application Developer does not support JSP pages tweak. It is strongly recommended to realize only imagination level but not business-level on the pages of JSP. That is why tweak of these pages is nit required because all realization code should be in Java files.

Usage of libraries with opened initial code, such as ZXing, Simple JSON or Apache POI were also useful. Instead of bar-code scanner developing from scratch, ready and well tested library was used. It has not only shrunk the time for development, but also provided the best safety and productiveness. Development, realization and testing of hybrid Android application with bar-code generator are described in the article.

Keywords: *IBM WebSphere, WebSphere Commerce, mobile application, Android, JEE, JSP, bar-codes.*

Introduction

One of the most popular aims of Java language is creation of corporative web-applications. In the beginning of simple technologies, such as servlet it was enough for meeting the needs of developers. However it became obvious, that servlet-technology is not enough for creation of corporative applications.

The response for increasing demand for Java language became an introduction of Java Platform Enterprise Edition (JEE) [1]. JEE determines the standard of application development in Java language on the basis of multicomponent architecture. This platform determines the set of programming interfaces, which should be represented by application server. These components are usually built into the server of applications, supporting Java Platform Enterprise Edition.

There is a multiplicity of server apps, capable of running enterprise applications. One of them is IBM WebSphere Application Server (WAS) [2]. It is multiplatform apps server, corresponding to the JEE certifications. WAS is based on the opened technologies, such as XML and Web-services. WAS presents a wide spectrum of service. For example, connection to a database, protocol management, load distribution, safety model. All these services might be used for enhancement of applications by the developers.

IBM WebSphere Commerce [3, 4] presents a set of ready-to-use tools, which helps the developers to emerge from the efforts of development the applications from scratch. IBM supplies a start store under the name of Madison, which can be used as start application for the development of specialized shop. Delivery of the start shop allows the developers to concentrate on the setting of a product to customers' needs. Madison is ready to use web-application with basic functions. It is also supplied with mobile version of the shop, which was used in this article. The main goal of the article is creation of mobile shop for Android operation system with the help of bar-codes on the base of search systems.

1. Model and technology

1.1. Model-View-Controller Projecting Template

WebSphere Commerce programming model consists of technologies combination, including pages of JavaServer, Enterprise JavaBeans, JavaServer Faces and Struts (fig. 1).

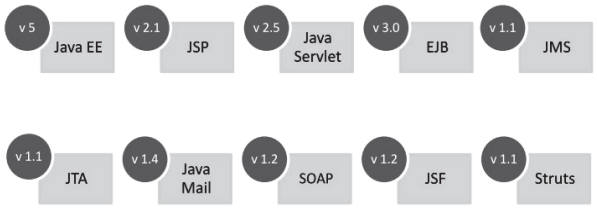

Fig. 1. JEE maintained technologies in WebSphere Application Server 7.0

One of the most important technologies being used during corporative applications creation are Enterprise JavaBeans (EJB), which are used for providing of business-logic and data access. Also it is possible to integrate exchange systems with the messages and clients of web-services with the help of application via EJB. Two types of EJB exist:

- EJB session of two types – sessions excluded and included the condition;

- EJB object of two types – constant container management and constant bean-components management.

The most significant ability to process upcoming requests from different drives among many requirements to corporative applications. Model-View-Controller pattern application presents to JEE-applications the possibility to work independent from the user's interface. MVC supports business logic and presentations separately one from each other.

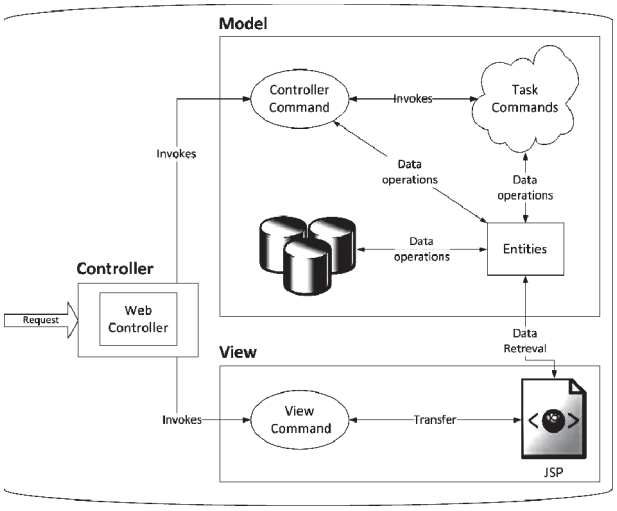

Fig. 2. WebSphere Commerce MVC coding pattern application

1.2. Model-View-Controller coding application

Model-View-Controller divides the problem of the users' interfaces into three units, namely: model, presentation and controller (fig. 2). Model contains condition of the application. Presentation is responsible for data interpretation obtained from the model and directs a report to the user. The controller processes user input and depending on this decides should the model be updated or new presentation reflected [5].

1.3. Struts framework

Apache Struts is a web-environment with opened initial code for Java web-applictaions creation intended for web-applications. Struts is intended for presentation of environment for web-applications creation provision using MVC architecture (fig. 3).

Struts can gather data from HTTP-requests. The main goal is to transfer the data between presentation and controller. Besides, Struts includes users' libraries of JSP tags, which help the developers in form creation. ActionServlet is provided by the Struts platform. This controller's servlet completes a form of actions from JPS logging. After form executing it delegates work into the category of action, where developer realizes the logic. Struts does not provide model categories. The developers should present these classes by themselves using EJB or JavaBeans [6].

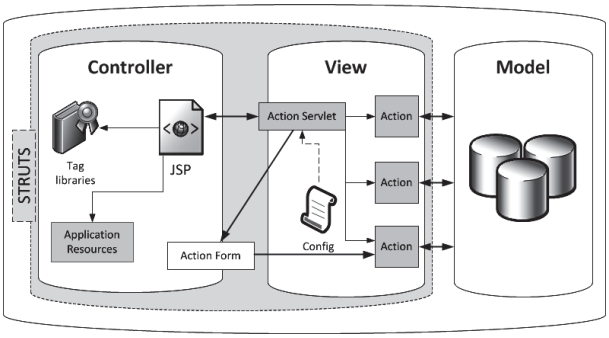

Fig. 3. Struts as a part of MVC architecture

Struts configuration at least can be placed into two files: struts-config.xml and web.xml. Web.xml is a an application descriptor, which presents by itself a base of Java. Struts web.xml determines servlet filters. It also determines remaining configuration files. It is possible to determine files with few struts-config.xml. In order to add another configuration file just indicate it in configuration field.

Struts-config.xml configuration file presents the connection between View and Model MVC. This is one of the most important Struts framework components.

Struts configuration file contains three main elements:

- <form-bean>;
- <global-forward>;
- <action-mapping>.

Form-bean element contains the definition of bean forms. Information of each form components is located in <form-bean>element. Detailed information is provided in <form-property> elements. These elements contain names properties and form types.

Global-forward element contains global forward definitions. Forward name is the name using for the JPS concrete page matching. <forward> element also contains meaning of the way, providing relative resource way.

Action-mapping element contains action definitions. Each of matching actions is defines in the <action> element.

Besides realization of MVC application, Struts also speeds up the development of web-application, providing extensive library of JSP tags. It also maintains input internationalization and users errors processing [7].

Instead of Struts usage the developers of JSP tags library can use JSTL library, which in general provides the same functions as the library provided by Status. Choose of the library depends on the developers habits.

1.4. Rational Application Developer

IBM Rational Application Developer (RAD) [8] for WebSphere Software V7 is an integrated habit of the development and platform for Java Platform Standard Edition (Java SE) and Java Platform Enterprise Edition (Java EE). RAD is based on Eclipse IDE.

RAD contains the set of tools and functions, for example, full Java EE support, UML editors, static and executing time analysis, as well as enhanced debugging and profiling. It also supports modern Java EE technologies. (fig. 4).

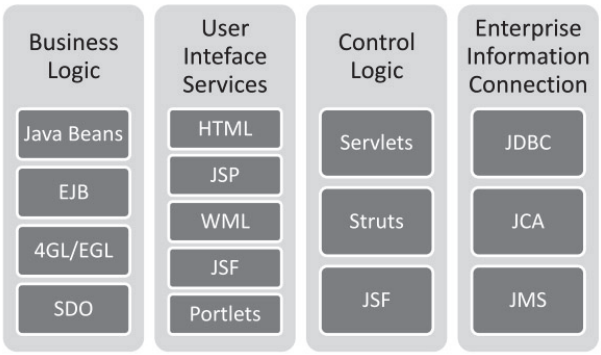

Fig. 4. Technologies, maintained with RAD

RAD helps to the Java developers quickly to develop and test Java applications. It is generated with WebSphere Application Server.

1.5. Mobile platform

IBM WebSphere Commerce storages maintain storages for many mobile devices [9]. These stores are half-functional applications of electronic commerce, which can be used with smartphones, tablets and web-browsers. Mobile stores are projected to be comfortable for finite users of unattached devices.

Mobile store can be developed in three different ways:

- mobile web-apps;
- native mobile applications;
- hybrid mobile applications.

1.6. Bar-code specification

Bar is an optical image of data, which can be read with the help of optical scanner presentation (for example, laser scanner or reader unit on the basis of camera). These codes contain data describing objects which they are attached to. Two main types of bar-codes: one-dimensional and two-dimensional (fig. 5).

Fig. 5. Bar-codes samples

One-dimensional bar-codes present combination of light and dark elements, which are marked with a symbol of quantitative code. Majority of one-dimensional bar-codes might contain only numeral data. QR-bar-codes are built of quadrates, which can be light or dark. These codes might contain alphanumerical data.

Aim of this article is to realize a function of bar-code scanner in mobile store. For arching of this aim the application uses a scanner of code on the basis of camera. Firstly, the application takes a shot of a bar-code and then implement the work of picture processing for data decoding, which contains a bar-code. The application might process both types of bar-codes. Depending on which code is used the application implements different kinds of operation. The problem is in the setting of the store presenting and business-logic for information usage, obtained from these bar-codes.

1.7. Auxiliary libraries

ZXing is the library with opened initial code for recycling of the bar-code images, written in Java. This library is included into the application for mobile app. ZXing supports a set of bar-codes format, one-dimensional and two-dimensional. It offers a wide functionality: starting from bar-codes scanning and to their creation. In order to start using this library in Java application it is necessary firstly to create it with the help of ant command line tool [10]. Initially ZXing was realized in Java, however there are ports for another languages, such as C++, Objective C, Ruby and partial C# and ActionScript.

Apache POI is the library, which provides API for many Microsoft documents. POI can open and process Excel, Word, PowerPoint, OpenXML4J, OLE2, Outlook, Visio and Publisher documents. This library proposes to the developers of simple API for reading and Microsoft documents recording in Java [11].

Simple JSON is light library, which realizes JSON-format for Java. It provides a wide function spectrum, such as coding, decoding and JSON text screening. Simple JSON does not depend of any external libraries. It is completely compliable with JSON specification [12].

2. Mobile application development

2.1. Medium test setting

Native and hybrid applications are used by the functions, inbuilt into Android operation system. Android SDK provides an emulator platform for testing; However this emulator does not support:

- telephone calls;
- USB connections;
- camera / video record;
- headphone processing;
- battery level control;
- processing SD-cards;
- bluetooth.

These limitations can be removed, using special mobile device with Android operation system. However, for local host server admission via mobile device the possibility of connection to the server via VPN is necessary.

QR-code scanner requires admission to the device camera. Emulator does not support capture; that is why it is necessary to use real device.

Madison mobile start store contains its settings in system_settings.xml file. In order to run a store on a device it is necessary to set up these settings. Below the fragment when system_settings.xml file with definite settings.

```
<string name="storeName">Madisons</string>
<string name="hostName">192.168.32.50</string>
<string name="storeId">10051</string>
<string name="storeIdentifier"></string>
<string name="catalogId">10051</string>
<string name="map_api_key"></string>
<string name="languageId">-1</string>
```

2.2. Bar-code activity realization

Mobile starter IBM store realizes basic realization of bar-code scane. Code scanning is processed with the application, set on a mobile device by the finite user. The user choose bar-code scanner from the menu. When finishing of scanning the intention is directed to a bar-code action. The aim contains the data with scanned bar-code. On the basis of this data the application implements finite actions and generates stated request for web-application. Business-logic processes this request and prepares JSP page. When the page is ready it will be directed back to the mobile device. Fig. 6 illustrates all the process.

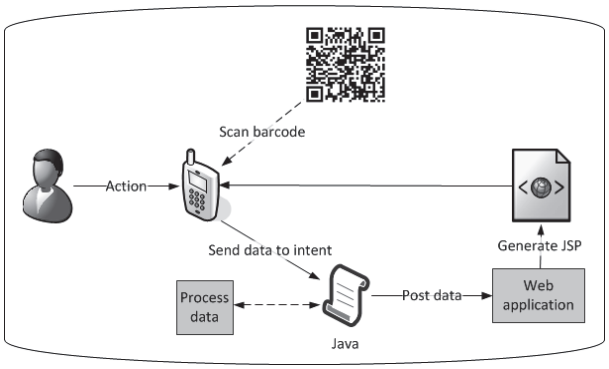

Fig. 6. Bar-codes samples

The main purpose of bar-code activity is to show up the result of JSP page. Hybrid application provides special representation under the name of StoreWebViewFlipper. In below mentioned fragment the code of Android StoreWebViewFlipper presentation is showed.

```
<com.ibm.commerce.android.hybrid.mobile.web.StoreWebViewFlipper
android:id="@+id/barcodeSearchWebViewFlipper"
android:layout_height="wrap_content"
android:layout_width="match_parent"
android:layout_weight="1.0" />
```

StoreWebViewFlipper in general is extended Android ViewFlipper, which is intended to the web-pages reflectionIn order to use all the advantages of StoreWebViewFlipper activity method class should inherit the following classes and realize one interface:

- AbstractStoreWebViewActivity abstract class ;
- IStoreWebViewActivity interface;
- AbstractHybridMobileActivity abstract class.

IStoreWebViewActivity interface contains the following abstract methods:

```
public StoreWebViewController getStoreWebViewController();
public String getDefaultUrl();
public int getContentViewId();
public int getStoreWebViewFlipperId()
```

GetStoreWebViewController method is realized in AbstractStoreWebViewActivity, and there is no need to predetermine this realization in BarcodeActivity. The remaining methods are realized BarcodeActivity.

2.3. Proxessing of the data from bar-codes

When a bar-code is scanned, intention with the data returns to activity (pic.6). AbstractHybridMobileActivity contains onActivityResult method of. This method processes all the intentions, returning to actions. The method verifies if the name of action corresponds to the SCAN_INTENT. Then it verifies a code of an intention result and sends them to BarcodeActivity.

Depending on the format of scanned bar-code different actions are performed. Basic one-dimensional bar-codes contain only the product number. On the basis of this number web-address in BarcodeActivity is created and transferred. A map with URL-address parameters is filled with storage identificators, catalogue and language. In the end of content is placed on a map. When all the data is placed, a map is delivered to the StoreWebUtils.buildUrl static method. This method based on delivered parameters returns URL-address of JSP search page. Finally, a new action with additional data is run. The main goal of this action is to show the search results of the users. In the fragment given below the process of recycling of one-dimensional bar-code.

```
Intent outGoingIntent = new Intent();
Map<String, String> queryParameters = new HashMap<String, String>();
queryParameters.put(StoreWebConstants.STORE_ID, get-
String(R.string.storeId));
queryParameters.put(StoreWebConstants.CATALOG_ID, get-
String(R.string.catalogId));
queryParameters.put(StoreWebConstants.LANGUAGE_ID,
getString(R.string.languageId)); queryParameters.put("sku", contents);
String path = getString(R.string.search_barcode_url_path);
Uri httpUrl = StoreWebUtils.buildUrl(lastHostName, path, queryParame-
ters, true);
outGoingIntent.setCategory(this, BarcodeActivity.category); outGoingIn-
tent.setData(httpUrl); startActivity(outGoingIntent)
```

QR-codes contain simple JSON object. Object, presented by JSON has as minimum two fields: action and content. The field of action contains identifier of action type, which can posess the following meanings:

- URL;
- Search;
- PROMOBUY.

Contents field contains basic data, obtained from a bar-code (for example, web-address for PROMOBUY

action type, object will contain additional field, named productId.

During scanning of two-dimensional bar code the content turns into JSON object. The following actions depend on the action field content. The fragment, which demonstrates realization of this process in given below.

```
if (format.equals("QR_CODE")) { try {
JSONObject jObject = new JSONObject(contents);
String action = jObject.getString("action"); String content = jOb-
ject.getString("content");
// Handling data code was omitted
} catch (JSONException e) {
Toast.makeText(this, R.string.ERROR_READ_QR_CODE,
Toast.LENGTH_LONG).show();
e.printStackTrace(); } }
```

URL action redirects the user of mobile device to the web-address containing in the contents field.

2.4. Struts Configuration

IBM WebSphere Commerce uses Struts infrastructure in order to help the developers to create web-applications. Struts configuration file is located in, Stores/WebContent/WEB-INF/struts-config-ext.xml file.

In order to add a new JSP page, it is necessary to set up actions and redirect it to the file of Struts configuration. Before mentioned fragment code shows configuration for BarcodeSearchResultDisplay JSP page.

```
<forward categoryName="com.ibm.commerce.struts.ECActionForward"
    name="m20BarcodeSearchResultView/10051"
    path="/mobile20/ShoppingArea/BarcodeSearchResultDisplay.jsp">
    </forward>
<action path="/m20BarcodeSearchResultView"
    type="com.ibm.commerce.struts.BaseAction">
    <set-property property="credentialsAccepted" value="0:P,0:P"/>
</action>
```

After adding of new records into configuration file, Struts should update configuration register through a console of administration. WebSphere Commerce configuration automatically allows to look through new JSP pages only administrators. In order to change this, it is necessary to add new admission policy. IBM provides a special tool for admission to policy changes, which are named acpload. This console package script, which accepts XML-file as parameter. In before mentioned XML code new policies, added to the configuration are presented.

```
<?xml version="1.0" encoding="ISO-8859-1" standalone="no" ?>
<!DOCTYPE Policies SYSTEM "../dtd/accesscontrolpolicies.dtd"> <Poli-
cies>
    <Action Name="m20BarcodeSearchResultView" Command-
    Name="m20BarcodeSearchResultView">
    </Action>
    <ActionGroup Name="AllSiteUsersViews" Own-
    erID="RootOrganization"> <ActionGroupAction
    Name="m20BarcodeSearchResultView"/>
    </ActionGroup>
</Policies>
```

2.5. JSP search page setting

One-dimensional bar-code contains only a product number. On the basis of this number search system includes only number of the product. On the basis of this number search system is trying to find a product. Firstly, it is searching only equal number. Is there are no results, it searches one more time, less strict this time. During the second search it is conducted with the help of LIKE operator EQUAL.

During default settings the data search component searches the products, elements or package. The setting of correspondent meanings in URL-address can set up these parameters.

For the search of database for correspondence to the goods CatEntrySearchListDataBean is used. This component is provided by IBM WebSphere and is used for extracting of the data about the product from the database on the basis of provided search parameters.

WebSphere Commerce presents two search variants for different cases of usage. These search forms are based on the component of data search in catalogue. Search in the catalogue proposes basic and extended search functions. Compare of these two methods is presented in the tabl. 1.

Table 1

Search methods compare

Simple search	Advanced search
A lightweight search feature	General searching feature
Can be used on any JSP page	Can be used on any JSP page
Search based on single term	Search based on multiple terms
Searching executed on base tables	Searching executed on base tables
Support Boolean values	Support Boolean values
	Advanced sorting criteria available

CatEntrySearchListDataBean is a data component, which is a Java-component, main aim of which is providing an access to the data from JSP-pages. This data component presents realization of abstract SearchBaseDataBean class. In the code mentioned above a part of JSP page realization of JSP page for one-dimensional search of bar-codes with CatEntrySearchListDataBean data component is shown.

```
<wcbase:useBean id="catEntSearchListBean" scope="page" category-
name="com.ibm.commerce.search.beans.CatEntrySearchListDataBean">
    <%-- Some less important code was omitted --%>
    <%-- Set the SKU number as a search term --%> <c:set prop-
    erty="sku" value="${WCParam.sku}"
      target="${catEntSearchListBean}" /> <c:set prop-
      erty="skuCaseSensitive" value="no"
      target="${catEntSearchListBean}" /> <c:set prop-
      erty="skuOperator" value="EQUAL"
      target="${catEntSearchListBean}" />
</wcbase:useBean>
    <c:if test="${catEntSearchListBean.resultCount == 0}">
    <%-- remove old result bean. Need new one --%> <c:remove
    var="catEntSearchListBean" />
    <wcbase:useBean id="catEntSearchListBean" scope="page"
category-
name="com.ibm.commerce.search.beans.CatEntrySearchListDataBean">
    <%-- Some less important code was omitted --%>
    <%-- Set the SKU number as a search term --%> <c:set prop-
    erty="sku" value="${WCParam.sku}"
    target="${catEntSearchListBean}" /> <c:set prop-
    erty="skuCaseSensitive" value="no"
    target="${catEntSearchListBean}" />
    <%-- We don't have any results, make query less strict --%> <c:set
    property="skuOperator" value="LIKE"
    target="${catEntSearchListBean}" />
</wcbase:useBean>
</c:if>
```

Data component performs search of coinciding lined in the database. It can return null, one or few lines. If it returns null lines, it tries again to use more extended spectrum, searching correspondent lines with LIKE operator instead of EQUAL. When on the JSP page returns only one line, automatic redirection on the product page is performed. From the beginning a catalogue record is derived from the list of results. Then the type (product, element or package) is verified. Finally, URL-address generates and transferred to the script of redirection JavaScript.

If data component returns many products, on the JSP page all are specifies as a list of products.

2.6. Setting of JSP order page

Two-dimensional bar-code contains action identifier for performing and content. QR-codes contain the object, modified into JSON format. This object contains necessary information for users tasks performing. One of these tasks can be an order of element with advertising codes.

IBM WebSphere supports extended campaigns and advertising action system. Administrator can decide the terms when customers obtain a discount for order, delivery or free present. Actions can concern each customer or just customers, defined according to several criteria. It is possible to set up public actions for each customer or limit it only for constant clients. These actions can be controlled through IBM Management Center.

The user of mobile store, which scanned QR-code, containing action in the right order obtains additional access to advertising code. This promo-code is intended only to clients, who order the product with the help of an application for mobile store. Action is spread automatically without any user knowledge. Customers can delete this advertisement if they want.

ProductDisplay JSP page contains a form with OrderChangeServiceItemAdd action, which is responsible for adding of a product to the cart. Android application generates special URL-address, which contains scan2buy parameter.

This parameter indicates on the element should be added to the cart. On the basis of this parameter JSP page adds additional input into a form and after loading of JavaScript page script presents a form. In the code given below the custom form and JavaScript case are presented.

```
<form id="AddToCartForm" method="post"
action="OrderChangeServiceItemAdd"> <input type="hidden"
name="catEntryId" value="${product.productID}" />
<input type="hidden" name="catalogId" value="${WCParam.catalogId}"/>
<input type="hidden" name="storeId" value="${WCParam.storeId}"/>
<input type="hidden" name="URL" value="${OrderItemDisplayURL}"/>
<input type="hidden" name="langId" value="${langId}"/>
<input type="hidden" name="quantity_1" value="1"/>
<input type="hidden" name="productId" value="${product.productID}"/>
<input type="hidden" name="errorViewName"
value="m20ProductDisplayView" />
<c:if test="${!empty(WCParam.scan2buy)}">
<input type="hidden" name="customPromoCode"
value="${WCParam.customPromoCode}" /> </c:if>
</form>
<script type="text/javascript"> //<![CDATA[
<c:if test="${!empty(WCParam.customPromoCode)}"> submitAddTo-
Cart();
</c:if>
//]]>
</script>
```

Page OrderItemDisplay JSP processes the order process. If user wants to order a product, it is necessary to add it to a cart. Users can add products in a cart from the page of product – on the ProductDisplay JSP page.

Action OrderChangeServiceItemAdd is a service, determined in Struts configuration file. This service is one of many services of the order, available via Struts and can be performed as URL command. Each Struts action name is reflected on correspondent facade client method. These methods are realized in OrderFacadeClient.

After upload of form users redirect to their cart, where foster is applied to the order. All the process is performed automatically and users see only the cart page.

2.7. JSP purchase page set up

After sending of the form on the ProductDisplay JSP page the users are automatically redirected to the OrderItemDisplay page. Promotion code is taken from WCParam. This map contains a pair of request parameter name and correspondent single-line meaning.

If promotion code reflects on the WCParam map, the process of verification is conducted. All active promotion codes are taken through PromoCodeListDataBean. This data component takes a list of all codes, connected with current order. If the list of promo-codes already includes a code, which was taken from WCParam, nothing happens. However, if this code is unique, additional is generated after the input. This input contains promotion code as meaning. After successful page creation with additional access the cart updates.

PromoCodeListDataBean is a data component, which can fill itself. There is no need to provide another data component for it. This component returns the list of all advertise codes, connected with the order, transmitted as parameter. Before filling the list it is necessary to establish order identifier. Basic code, responsible for promotion code application order is shown below.

```
<wcbase:useBean id="promoCodeListBean" category-
name="com.ibm.commerce.marketing.databeans.PromoCodeListDataBe
an" scope="page">
<c:set property="orderId" value="${order.orderIdentifier.uniqueID}" tar-
get="${promoCodeListBean}" />
</wcbase:useBean>
<c:forEach items="${promoCodeListBean.codes}" var="currentCode"
varStatus="status">
<c:if test="${currentCode.code == WCParam.customPromoCode}">
<c:set var="duplicateCustomPromoCode" value="true" />
</c:if>
</c:forEach>
<c:choose>
<c:when test="${!empty(WCParam.customPromoCode)
&& duplicateCustomPromoCode != true }">
```

```
<input value="<c:out value="${WCParam.customPromoCode}" />"
type="hidden" name="promotion_code" />
</c:when> <c:otherwise>
<input type="text" name="promotion_code" /> </c:otherwise>
</c:choose>
<c:if test="${!empty(WCParam.customPromoCode)
&& duplicateCustomPromoCode != true}"> <script type="text/javascript">
//<![CDATA[ updateShoppingCart(document.ShopCartForm);
//]]> </script>
</c:if>
```

2.8. Bar-codes generator

The application for reading of bar-codes is not enough for providing the client with half-functional product. In order to make a scanner function of bar-code useful, it is also important to deliver a bar-code generator. This generator should be simple for client in usage.

Client provides data for generator of bar-code in the form of Excel page. This page can be used in many other scripts, but it contains not only the data for bar-code generator. That is why bar-code generator should be flexible. Generator is created with the usage of Java language and can be performed on any computer with installed virtual machine Java. Generator works as console application. For bar-codes creation is performed.

```
root$ java -jar BarcodeGenerator.jar Generate barcodes from XLS file
usage: BarcodeGenerator [ options ] file.xls
--format=format: Barcode format. Code39 or QRCode. Default: QRcode.
--column=number: Column of xls file, where content is. Start counting
from 0. Default: 3
--actionColumnRows=number: Column of xls file, where action type is.
Start counting from 0. Default: --column + 1
--width=pixels: Image width. Default: 400
--height=pixels: Image height. Default: 300
root$ java -jar BarcodeGenerator.jar Generate barcodes from XLS file
usage: BarcodeGenerator [ options ] file.xls
--format=format: Barcode format. Code39 or QRCode. Default: QRcode.
--column=number: Column of xls file, where content is. Start counting
from 0. Default: 3
--actionColumnRows=number: Column of xls file, where action type is.
Start counting from 0. Default: --column + 1
--width=pixels: Image width. Default: 400
--height=pixels: Image height. Default: 300
```

Bar-code generator is filled with necessary libraries as JAR file. This archive includes the following libraries:

- ZXing 2.0libtrary;
- Apache POI 3.7library;
- JSON 1.1.1 simple library.

Bar-code generator contains two classes: BarcodeGenerator и DataHandler. DataHandler realizes interface IDataHandler.

BarcodeGenerator class contains basic method. In this method parameter are set. Below the process of parameters seting in BarcodeGenerator script is shown in the code fragment:

```
for (String arg : args) {
if (arg.startsWith("--format")) {
if (arg.split("=")[1] == "code39") { dataType = BarcodeFormat.CODE_39;
} else {
dataType = BarcodeFormat.QR_CODE; }
} else if (arg.startsWith("--column")) {
rowNumber = Integer.parseInt(arg.split("=")[1]); } else if (arg.startsWith("--
width")) {
width = Integer.parseInt(arg.split("=")[1]); } else if (arg.startsWith("--
height")) {
height = Integer.parseInt(arg.split("=")[1]); } else if (arg.startsWith("--
actionColumnNumber")) {
actionRowNumber = Integer.parseInt(arg.split("=")[1]); } }
```

When all parameters are set and there are no exceptions, application starts reading Excel page. This file might contain big amount of information not related to the task of bar-code creation. However, when application is performed, it is possible to state columns, which should be necessary derived from the page. Tabl. 2 presents a sample of initial data file.

Table 2

Sample of data source file for bar-code generator

Electrolux ASM450-kulhovatkain	242158	SC1203E	10,5
Nikon D5100 –digitaalinen	142958	SC1202E	11,0
Peak Performance Takki	282957	SC1201E	27,00
Hugo Boss Black Solmio	341953	SC1203D	1,6

Each obtained data line is DataHandler transferred, where it is processed. Dependant on the bar-code type the data is converted into String or JSON String. Below mentioned a fragment, showing data delivery to DataHandler and realization transformation.

```
if (dataType == BarcodeFormat.QR_CODE) {
Cell actionCell = row.getCell(actionRowNumber,
Row.CREATE_NULL_AS_BLANK); dataHandler.setData(cell.toString(),
actionCell.toString());
} else { dataHandler.setData(cell.toString()); }
data = dataHandler.getData();
@Override
public String getData() { return data; }
@Override
public void setData(String data, String action) {
params.put(DataHandler.CONTENT_FIELD, data);
params.put(DataHandler.ACTION_FIELD, action);
JSONObject jObject = new JSONObject(params); set-
Data(jObject.toJSONString()); }
```

When data is analyzed and ready for transformation into a bar-code, processing of images is performed. This process is processed by ZXing library. Application creates MultiFormatWriter object and BitMatrix object.

MultiFormatWriter has the method encode(), which is responsible for data transformation from parameter into bar-code of transferred typeIt also establishes sizes of a bar-code. This method returns the object of BitMatrix type.

In order to save a bar-code in the form of file, just pass returned BitMatrix object to MatrixToImage-Writer.writeToFile static method. Code below shows the described process.

```
data = dataHandler.getData();
matrix = barcodeWriter.encode(data, dataType, width, height);
MatrixToImageWriter.writeToFile(matrix, "png",
new File(cell.toString() + ".png"));
```

Bar-codes are generated in the folder, where bar-code generator was run. File names are determined in accordance with correspondent text cell.

3. Working application testing

In order to provide a client with full function product, it is not enough to develop it. Before presenting finite product to the customer it is not enough to de-

velop it. Before presenting on finite product to the customer it is also necessary to verify it.

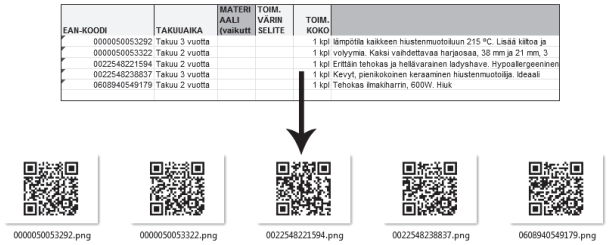

Fig. 7. QR-codes ceation on the basis of data in XLS file

On fig. 8–11 screenshots of working mobile application are shown.

Fig. 8. Main window of mobile application

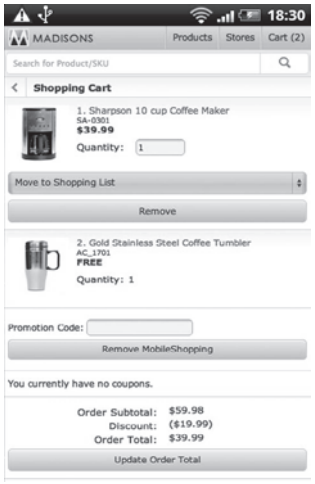

Fig. 9. Cart after QR-code with promotional code scanning

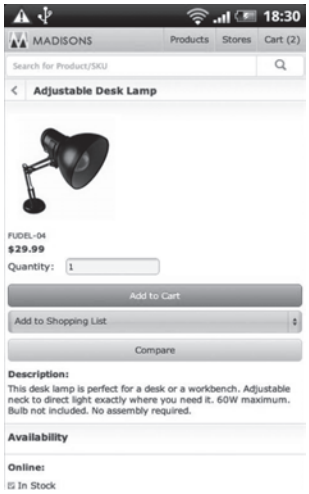

Fig. 10. Page of the product after bar-code scanning

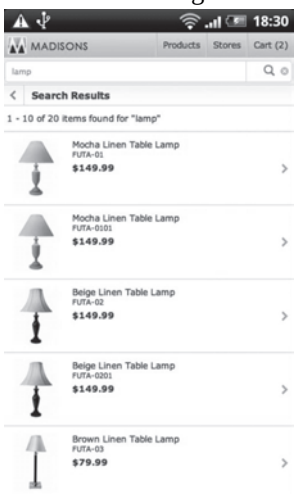

Fig. 11. Kist of products after bar-code scanning

Conclusions

In the article hybrid mobile application for Android operation system with search system on the basis of bar-codes and QR-codes is presented.

IBM provides a wide range of Madison starters store, which contains basic realization of bar-code scanner. In created application this function is extended with the help of full function and useful tool for clients and saving of the users as attractive marketing tool for improvement of sellings and increase of attractiveness for customers. For users of the store this would be practical and simple in application usage, which would be useful for daily purchases.

Bar-code scanner, presented by IBM was extended with the new functions, such as search of the products on the basis of information about bar-codes, redirection of users to the page, performance of the users search and regulation of goods with the help of special advertising actions.

Users can search a product in internet store with the help of simple bar-code scanning. They can verify presence of the good in internet store, compare the prices and take a part in marketing campaigns.

Clients can conduct interesting and modern marketing campaigns with the usage of different bar-codes types. With several choose variants they can plan and start campaigns of the new type.

IBM provides wide mobile start store, which does not require complicated configurations for its running. Improvement of this start store and developing of new functions is really relatively simple.

WebSphere Commerce usage for electronic commerce applications creation significantly shortens time of developing. Creation of new pages with usage of JSP technologies is simple. However, it is difficult to tweak these pages. Tweak tool, provided by Rational Application Developer does not support JSP pages tweak. It is strongly recommended to realize only imagination level but not business-level on the pages of JSP. That is why tweak of these pages is nit required because all realization code should be in Java files.

Usage of libraries with opened initial code, such as ZXing, Simple JSON or Apache POI were also useful. Instead of bar-code scanner developing from scratch, ready and well tested library was used. It has not only shrunk the time for development, but also provided the best safety and productiveness.

References

1. Gupta, Arun (2014), Java EE 7. Basic = Java EE 7 Essentials. - Moskow, - 336 p.

2. IBM WebSphere Application Server (2017), [Електронний ресурс]. – Режим доступу: http://www-01.ibm.com/software/webservers/appserv/was/.

3. IBM - WebSphere Commerce (2017), [Електронний ресурс]. – Режим доступу: http://www.ibm.com/software/genservers/commerceproductline/compare.html

4. Baila, B., Bleibtrey, N., Nagineni, C. and Pesic, B. (2011), WebSphere Commerce V7.0 Feature Pack 2 Search Solution Overview and Deployment. - 296 p.

5. Crawford, W. and Kaplan, J. (2003), J2EE Design Patterns. O'Reilly.

6. Struts (2017), The Apache Software Foundation [Електронний ресурс]. – Режим доступу: http://struts.apache.org/.

7. Siggelkow, B. (2005), Jakarta Struts Cookbook. O'Reilly.

8. Wahli, U., Cui, H., Fleming, C., Mehta, M., Rohr, M. and Ugurlu, P. (2007), Rational Application Developer V7 Programming Guide. IBM.

9. IBM Info Center. WebSphere Commerce Version 7.0.0.5 (2017), [Електронний ресурс]. – Режим доступу: http://publib.boulder.ibm.com/infocenter/wchelp/v7r0m0/topic /com.ibm.commerc e.developer.doc/concepts/ csdapplication.htm

10. ZXing (2017), [Електронний ресурс]. – Режим доступу: http://code.google.com/zxing/.

11. The Apache POI Project (2017), [Електронний ресурс]. – Режим доступу: http://poi.apache.org.

12. Fang, Y. (2017), JSON Simple [Електронний ресурс]. – Режим доступу: http://code.google.com/json-simple.

Надійшла до редколегії 8.06.2017

Рецензент: д-р техн. наук проф. Г.А. Кучук, Національний технічний університет «Харківський політехнічних інститут», Харків.

РОЗРОБКА ГІБРИДНОГО МОБІЛЬНОГО ДОДАТКУ ДЛЯ ANDROID З ФУНКЦІЄЮ ОБРОБКИ ТА СКАНУВАННЯ ШТРИХ-КОДІВ

Нух Таха Насіф

У цій роботі представлений процес гібридного мобільного додатка для операційної системи Android з новою функцією обробки і сканування розробки штрих-кодів. Розглянуто теоретичні основи, проаналізована архітектура корпоративного додатка WebSphere Commerce, що забезпечує високий рівень масштабованості, продуктивності, надійності безпеки і керованості. WebSphere Commerce коротко представлена за допомогою моделі програмування, мобільної платформи, специфікації штрих-коду і використовуваних бібліотек. У статті описані розробка, реалізація і тестування гібридного додатки Android з генератором штрих-коду.

Ключові слова: *IBM WebSphere, WebSphere Commerce, мобільний додаток, Android, JEE, JSP, штрих-коди.*

РАЗРАБОТКА ГИБРИДНОГО МОБИЛЬНОГО ПРИЛОЖЕНИЯ ДЛЯ ANDROID С ФУНКЦИЕЙ ОБРАБОТКИ И СКАНИРОВАНИЯ ШТРИХ-КОДОВ

Нух Таха Насиф

В настоящей работе представлен процесс разработки гибридного мобильного приложения для операционной системы Android с новой функцией обработки и сканирования штрих-кодов. Рассмотрены теоретические основы, проанализирована архитектура WebSphere Commerce - корпоративного приложения, обеспечивающего высокий уровень масштабируемости, производительности, безопасности, надежности и управляемости. WebSphere Commerce кратко представлено с помощью модели программирования, мобильной платформы, спецификации штрих-кода и используемых библиотек. В статье описаны разработка, реализация и тестирование гибридного приложения Android с генератором штрих-кода.

Ключевые слова: *IBM WebSphere, WebSphere Commerce, мобильное приложение, Android, JEE, JSP, штрих-коды.*