Salesman Mobile Application (on Android)

Tanmay Grover, Asheet Makhija, Ankur Goyal , Deepak Kumar Sharma

Abstract— The rapid development and popularity of smartphones in the consumer market has given an opportunity to software engineers to facilitate business by mobile phone applications .Therefore this paper looks to resolve the challenges faced by sales industry that is examined by developing a “Salesman Application” an android mobile app that provides various hands-on services to a salesman thereby minimizing the reporting time and increasing the efficiency . The app will allow a salesman to manage his appointments with clients , submit immediate orders , generate & print receipt via bluetooth printer , track his/her performance index , maintain a products catalogue, also features different payment modes (cheque , DD, Cash) and maintains client’s history along with the feedback & a picture of the client (by creating phonegap camera plugin) on their android smartphones and tablets .

Index Terms— Salesman Mobile Application, Smart Phone, Android

1 INTRODUCTION

Smart mobile devices are the fastest growing computing platforms with 1.6 billion mobile device users in 2013 [3]. This rapid development and growth of smart phones in consumer market over the last few years has alarmed the platform that is utilized for social business, entertainment, gaming, productivity marketing using software applications involving global positioning sensors (GPS), and wireless connectivity, photo/video capabilities, built in web browsers, voice recognition and various other native capabilities of the smart phone. These features present in mobile devices present new challenges and requirements to application developers that are not found traditional mobile apps.

SALESMAN APP allows a salesman to use a mobile device to maintain a track of his targets, products and goals. The software being used for development is IBM Worklight which is a plug-into the Eclipse IDE. It provides an open, comprehensive and advance mobile application platform to build, run and manage mobile applications. The project is being managed on Websphere Application Server (WAS) Liberty Profile .The objective is to save the salesman’s time that he uses to report to the office before and after going to the field, this is achieved by providing all the required data to the salesman on his smart phone before he heads to the field thus he then only needs to travel between his home and field thereby minimizing the reporting time to the office and increasing the efficiency. A synchronize button is provided to sync the data with the server providing new appointments to the salesman on daily basis and updating offers on products etc. GPS enables the manager of the salesman to easily monitor his motion.

The application can be divided into following parts:-

1. Appointment Manager: It allows you to manage the meeting time, date and place with the client and also to store the client's contact information, ie: Client name, address, email id, telephone no.
2. Check List: It allows the salesperson to check at a glance the result and status of all its conducted appointments. The clients will be filtered on the basis of three categories Confirmed Clients (those who have placed the order), Rejected Clients (fail), Potential Clients (negotiating).
3. Order Facility: This portal will allow the salesperson to efficiently place the order through the app and can also generate its corresponding receipt that contains the print button which will allow him to print the receipt through the bluetooth printer.
4. Product Information: This will be the brochure of the entire product range offered by the company.It will contain the product specifications, overview, price, supply information.This will help the salesperson to market his product accurately.
5. Performance Index: This will contain a description of the salespersons performance for self evaluation and allow the manager to keep a tab on the salespersons work. It will have a real time updated performance index which shall show the time frame ie. Targets for the month and targets met.
6. Special Offers: It shall contain the special offers like free merchandises or any custom scheme available on any product.
7. Submit button: On one click the complete summary of the days work will be sent to the company.
8. Update button: Will bring information from the server (i.e. new client information, new product information, special offers etc.)

- *Tanmay Grover , B.Tech (Department of Computer Science and Engineering), Manav Rachna College of Engineering , MD University
  Email :tanmay_ksa@yahoo.com*
- *Asheet Makhija ,Country Head, Services, IBM India Pvt. Ltd.*
- *Ankur Goyal, Technical Analyst, IBM India Pvt. Ltd*
- *Deepak Kumar Sharma is assistant professor (Information Technology) in MRCE, M.D University, India.*

2 WHY WORKLIGHT STUDIO?

Worklight [10] is been chosen as the programming environment because it facilitates a large number of Hybrid Mobile App features. The Worklight Studio's capabilities and the software development kit (SDK) [1]of the Worklight offers the development of mobile and omni-channel applications (apps) throughout multiple mobile platforms, including iOS, Android, BlackBerry, Windows 8, Windows Phone and Java ME. The Worklight optimization framework fosters code reuse and delivers rich user experiences that match the styling requirements of each target environment. With such code reuse, Worklight software reduces costs of development, reduces time-to-market and provides strong support for your ongoing management efforts. Worklight software delivers integrated functional testing capability. For apps that are developed within Worklight software, developers or testers can record a sequence of actions on a mobile device. Once the recording has been captured, it becomes the basis of an intelligent, resilient and code-less test case that can be "played back" on demand on virtually any iOS or Android device or emulated iOS or Android device within the same OS family.

2.1 Development tools:-

1. Worklight Studio: In a mobile development platform, cross-platform portability of the code is critical for mobile device development. Various methods exist to achieve this portability. With IBM Worklight, you can develop multiplatform s by using Worklight Studio, which is a mobile development studio, to address the requirements of the organization.
2. WorklightServer: The Worklight Server is a runtime container for the mobile applications you develop in Worklight Studio. It is not an application server in the Java Platform, Enterprise Edition (JEE) sense. It acts as a container for Worklight application packages, and is in fact a collection of web applications (optionally packaged as an EAR file) that run on top of traditional application servers.
3. Worklight device runtime components: IBM Worklight provides client-side runtime code that embeds server functionality within the target environment of deployed apps. These runtime client APIs are libraries that are integrated into the locally stored app code. They complement the Worklight Server by defining a predefined interface for apps to access native device functions. Among these APIs, IBM Worklight uses the Apache Cordova development framework. This framework delivers a uniform bridge between standard web technologies (HTML5, CSS3, JavaScript).
4. Worklight Console: The Worklight Console is used for the control and management of the mobile organization, from managing deployed applications to collecting and analyzing user statistics. It assigns device-specific identifiers (IDs) to ensure secure application provisioning. Remotely disable applications by using preconfigured rules of app version and device type.
5. Application Center: With the Application Center, you can share mobile applications that are under development within your organization in a single repository of mobile applications. Development team members can use the Application Center to share applications with members of the team. This process facilitates collaboration between all the people who are involved in the development of an application.

2.2 Software Requirement

Operating Systems, Eclipse versions and SDKs supported by IBM Worklight. Prerequisites of IBMWorklight 5.0.5:-
Development tools: Eclipse 3.6.2/3.7.2/4.2
Java SDK: IBM Runtime Environment, Java Technology Edition 6.0/7.0
Javascript libraries: Dojo Toolkit, Jquery Mobile [4], Sencha Touch, jquery.
Application Servers: Websphere Application Server 7.0 / Apache Tomcat7.0
Databases: DB2 Enterprise Server Edition 9.7/Mysql 5.5/ Apache Derby

2.3 Hardware Requirement

Detailed hardware requirements by operating system family.

Table1: Hardware Requirement

OPERATING SYSTEM	HARDWARE	REQUIREMENT
1. LINUX	**Disk Space**	**Minimum : 150 MB**
	Memory	**Minimum : 2GB RAM**
2.Mac OS	**Disk Space**	**Minimun : 150 MB**
	Memory	**Minimun: 2GB RAM**
3.Mobile OS	**Disk Space**	**Minimum: 150 MB**
	Memory	**Minimun : 32MB RAM**
4.Windows	**Disk Space**	**Minimun: 150 MB**
	Memory	**Minimum: 2GB RAM**
5.AIX	**Disk Space**	**Minimum:150 MB**
	Memory	**Minimun: 2GB RAM**

2.3 Methodology [8]

This Application is Developed in HTML5 enabled Technology and JQuery Mobile [4], the native framework provides access to device native functionalities in the form of Javascript functions that can be called within the HTML5 web pages .The tool used to develop the the app is known as "IBM Worklight" , it contained the following modules :-

- Setup: This category allows to learn how to set up the development environment to work with IBM Worklight.
- Basics: This category helps to learn how to create IBM Worklight apps in different mobile operating systems.
- Server-side development: This category helps to learn how to develop the server code (adapters) that your mobile application requires to integrate with enterprise backend applications and cloud services
- Mobile-client development: This category helps to learn how to implement different features in your mobile application, such as controls, skins, offline access. You also learn how to develop your client application by using native APIs.
- Authentication: This category helps to learn how to pro-

tect sensitive assets, such as application code, user data, and corporate services, against unauthorized accesses.

- Team work: This category helps to learn how to work as a team as your company develops your apps.
- Advanced: This category helps to learn how to use some advanced features of IBM Worklight, such as push notifications and device analytics.
- Non-mobile environments: This category helps to learn how to develop desktop apps.

3 USE CASE DIAGRAMS

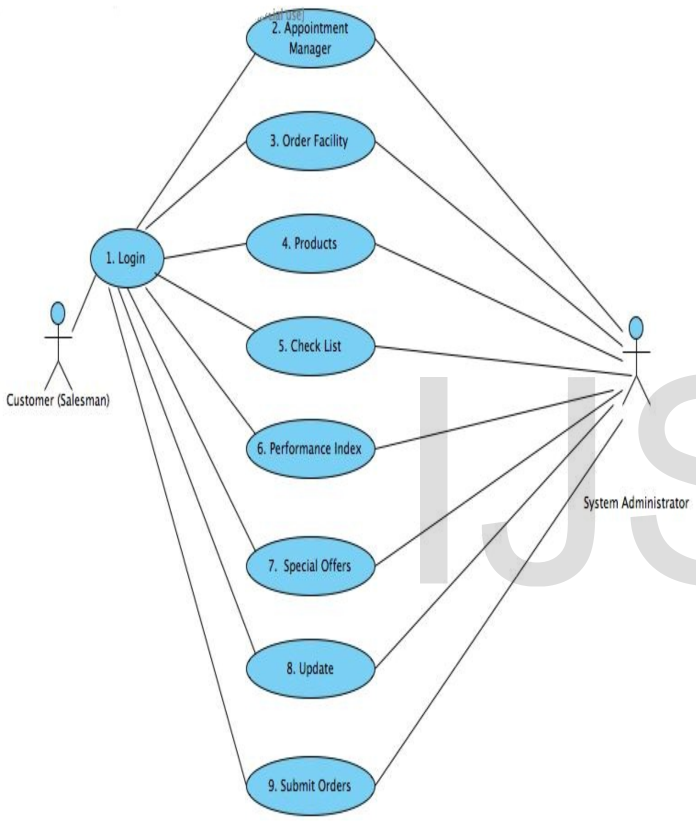


Figure 1: Usecase diagram for salesman application

USE CASE 1: Login into Salesman App.

Objective	To login into Salesman App
Primary Actor	Consumer
Pre-Conditions	The consumer has Salesman App on their mobile device
Post-Conditions	App menu displayed on mobile device
Flow of Events	Consumer enters login credentials. Consumer clicks login/Enter App. Mobile device matches data with data in LDAP directory. LDAP server accepts request. Consumer views menu.

USE CASE 2: View Appointment Manager Status

Objective	To know the status of Appointments
Primary Actors	Consumer , System administrator
Pre-Conditions	System administrator has valid login to admin console
Post-Conditions	Appointment Manager updates are displayed
Flow of Events	System administrator enters into admin console System administrator updates the appointment status status. The Consumer only views its today, upcoming and missed appointments.

USE CASE 3:ORDER FACILITY

Objective	To place Purchase Order.
Primary Actors	Consumer , System administrator
Pre-Conditions	System administrator can only view purchase orders.
Post-Conditions	Consumer can place order for client.
Flow of Events	Consumer enters order facility and places purchase order as specified by the client. Consumer also generates receipt of the order there. The System administrator only views the orders.

USE CASE 4: VIEW PRODUCTS INFORMATION

Objective	To know the products to be sold.
Primary Actors	Consumer , System administrator
Pre-Conditions	System administrator has valid login to admin console
Post-Conditions	Various Products are displayed
Flow of Events	System administrator enters into admin console System administrator updates the products details. The Consumer only views different products and their specifications.

USE CASE 5: Tracking the check list

Objective	To know the status of Clients
Primary Actors	Consumer , System administrator
Pre-Conditions	System administrator has valid login to admin console.
Post-Conditions	Client's statuses are updated.
Flow of Events	System administrator enters into admin console System administrator updates the client's status. The Consumer views, confirmed, rejected and potential clients.

USE CASE 6: Tracking Consumer's Monthly Performance

Objective	To track Consumer's performance.
Primary Actors	Consumer , System administrator
Pre-Conditions	System administrator has valid login to admin console.
Post-Conditions	Consumer's monthly targets set.
Flow of Events	System administrator enters into admin console System administrator sets the consumer's targets for that month. The Consumer views a numeric slider that displays his targets for month and no. of targets met.

USE CASE 7: Offers on products

Objective	View offers on Products.
Primary Actors	Consumer , System administrator
Pre-Conditions	System administrator has valid login to admin console.
Post-Conditions	Consumer updates about available offers to clients.
Flow of Events	System administrator enters into admin console System administrator sets the offers/discounts with corresponding products. The Consumer views and updates about products available with offers.

USE CASE 8: Update

Objective	To update appointments , products , offers, checklist .
Primary Actors	Consumer , System administrator
Pre-Conditions	System administrator has valid login to admin console.
Post-Conditions	Consumer avails updated app data.
Flow of Events	Consumer sends request to update the app data. System administrator enters into admin console System administrator sets the new offers/discounts, new appointments, new product list and updates app data wherever required. The Consumer views the updated app data.

USE CASE 9: Submit Orders

Objective	To submit consumers daily work.
Primary Actors	Consumer , System administrator
Pre-Conditions	System administrator can only view the consumer's orders.
Post-Conditions	Consumer received sent alert.
Flow of Events	Consumer updates his team about his day's work by submitting his orders. System administrator receives the data and forwards it to the manager.

4. SECURITY OF THE APPLICATION

4.1 LDAP Authentication

- Application security is ensured by LDAP authentication[5]. An LDAP directory is required to implement authentication .Apache directory is used for authentication[2].
- Authentication has a security test- LDAPSecurityTest .
- This is defined in Authenticationconfig.xml in the Server/Conf folder of the project.
- Every Security Test has a Realm and a login module.

Realm= LDAPRealm
Authenticator used = UsernamepasswordJSONAuthenticator.

Authenticators are responsible for collecting user credentials.
UsernamepasswordJSONAuthenticator.java
(Found in Server/java/com/worklight of your project folder)

Login module= LDAPLoginModule

LoginModules are responsible for verifying user credentials.

LDAPLoginModule has several parameters.
These parameters contain the settings to connect to and verify from LDAP SERVER.

Ldaphost = the hostname with port of your LDAP server.
LdapDomain = the domain name of your Ldap User Registry (where you have defined the users.)
searchBase = The Base for where the Module will

search for credentials to verify.
MemberofMustContain = A string to specify a group.

4.2 LDAPLoginModule.java:-

- (found in Server/java/com/worklight of your project)
- Contains the Java code for verifying user credentials received from JsonAuthenticator.
- It gets configuration settings from Authenticationconfig.xml as strings.
- The username and password are passed as strings.
- Then an environment is created using configuration provided earlier.
- Then the module connects to LDAP server and compares credentials.
- If credentials match then Authentication is successful and all user attributes are retrieved from LDAP server.
- Otherwise an Authentication Failure exception is thrown.

4.3 LDAPRealmChallengeHandler.js

This Javascript is responsible for API implementation of authentication.

- It checks the Authstatus (Logged in or logged out) and then decides which page to show and which page to hide.
- Its also responsible for navigating from LOGIN page to APPLICATION.
- Provides Login, Logout Functionality and triggers the login process from the API.
- Thus each file will communicate with each other to ensure only predifined users already registered in LDAP Directory will get access to the Application.

4.4 OTP and Mobile Active Code Authentication

IBM Security Access Manager for Enterprise Single Sign-On supports the use of one-time passwords (OTP) and Mobile ActiveCodes (MAC) to authenticate users that logon to corporate VPN servers, AccessAssistant, or Web Workplace.For requesting the server to synchronize the data with the app an OTP password would be generated upon salesman's request.[6]

4.4.1 One-time password

One-time password is a randomly generated password, intended only for one user for a specific time and purpose and provided to the user either through SMS or an OTP token.

OTP is used as an authentication factor for users to log on to AccessAssistant or Web Workplace.

You use AccessAdmin to:

- Assign an OTP token to a user or revoke an OTP token from a user.
- Enable or disable authentication with an OTP token for an authentication service.

IBM Security Access Manager for Enterprise Single Sign-On supports the OATH HOTP algorithm and selected vendor-specific OTP algorithms. IBM Security Access Manager for Enterprise Single Sign-On supports the following devices:

- VASCO Digipass GO 3
- Authenex A-Key OATH-only token without USB interface (OATH-based OTP)

Authentication with OTP tokens is centrally logged in the IMS Server. Administrators or Helpdesk officers can view the audit logs throughAccessAdmin, including logs reported by AccessAgent.

4.5 Mobile ActiveCodes

A Mobile ActiveCode is a randomly generated, event-based one-time password. The Mobile ActiveCode is generated on the IMS Server. The Mobile ActiveCode is delivered through a second channel, such as short message service (SMS) on mobile phones or through email.[6]

The users can use Mobile Active Code to logon to the following applications:

- Applications supporting RADIUS, such as VPN Servers
- You must configure applications that support RADIUS to redirect the authentication to the IMS Server. Use this setup to have applications grant or deny access to users based on whether the OTP is successfully verified by the IMS Server.
- Web applications
- AccessAssistant or Web Workplace

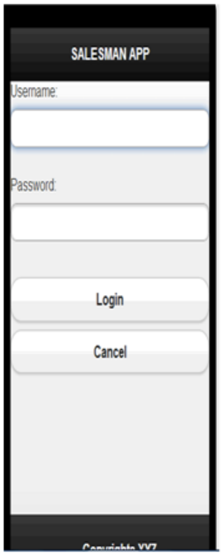
4.6 Salt Encryptioin of Data

In cryptography, a salt [9] is random data that are used as an additional input to a one-way function that hashes a password or passphrase.

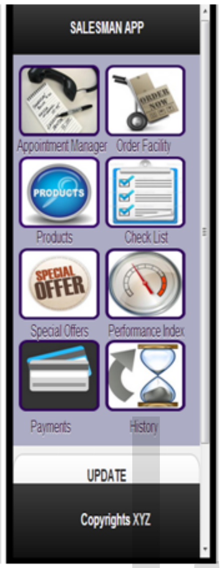
A new salt is randomly generated for each password. In a typical setting, the salt and the password are concatenated and processed with a cryptographic hash function, and the resulting output (but not the original password) is stored with the salt in a database. Hashing allows for later authentication while defending against compromise of the plaintext password in the event that the database is somehow compromised. The original intent of salting was primarily to defeat pre-computed rainbow table attacks that could otherwise be used to greatly improve the efficiency of cracking the hashed password database. A greater benefit now is to slow down parallel operations that compare the hash of a password guess against many password hashes at once.

Cryptographic salts are broadly used in many modern computer systems, from UNIX system credentials to Internet security. By salt encryption of Data if the data is lost then it becomes use less, and cannot be used hence it remains protected. It is common for a web application to store in a database the hash value of a user's password.Without a salt; a successful SQL injection attack may yield easily crackable passwords. Because many users re-use passwords for multiple sites, the use of a salt is an important component of overall web application security.

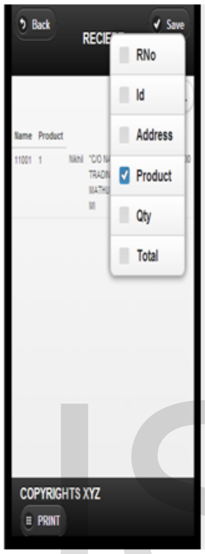
5. SCREENSHOTS



(Login Page)



(Menu Page)



(Receipt page)

6. CONCLUSION AND FUTURE SCOPE

The development of mobile application is not an easy task. There is a variety of platforms and technologies to choose from [7]. User experience with mobile devices and software is various.The Salesman App is successfully deployed on the emulator & test device and is working as per the use cases. The App is successfully compiled and has performed its required task of providing productivity and efficiency to the consumer.

6.1 Future work

- New Functionalities
- Interface Improvement
- Using Web Services for Mobile learning Applications helps the process of development by providing standardized way of communication between mobile clients and servers.
- Integrating and implementing Bluetooth printer .
- Making the user- device experience completely sublime.

Reference:

[1].Android SDK available at, http://developer.android.com/sdk/index.html

[2] Apache directory available at, http://directory.apache.org/studio/

[3]Josh Dehlinger and Jeremy Dixon,"Mobile Application Software Engineering: Challenges and Research Directions", Workshop on Mobile Software Engineering, MOBICASE, USA, 2011.

[4] JQUERY available at, http://www.Jquermobile.org/demos

[5]LDAP Authentication available at http://pic.dhe.ibm.com/infocenter/seas/v2r4m1 /index.jsp?topic=%2Fcom.ibm. help.seasfielddefs.doc%2FSEAS_LDAP_Auth.html

[6] OTP and Mobile Active Code Authentication available at,http://pic.dhe.ibm.com/infocenter/tivihelp/v2r1/index.jsp?topic=%2Fcom.ibm.itamesso.doc%2Fconcepts%2Fauthentication_types_addl_otptoken.html

[7].Paul Pocatilu,"Developing Mobile Learning Applications for Android using Web Services", Inforec Association EUROPE, 2010.

[8].PDF'S is available at http://www.ibm.com/developerworks/mobile/worklight/getting-started.html

[9] Salt Encryption of Data available at https://en.wikipedia.org/wiki/Salt_(cryptography)

[10] Technical White Paper IBM Worklight 6.0 is available at, http://public.dhe.ibm.com/common/ssi/ecm/en/wsw14181usen/WSW14181USEN.PDF