Shipyard Management through Android Technology

Arun Kumar. R

Abstract: ***The purpose of paper is to provide security for user. User can store all the information about ships and loads and also the incoming and outgoing transactions in entire port information. The existing system is when user want to store the data information about all ships in port user will registered in a book to maintain all the data and operations timings about load for this they are keeping one accountant he will do accounts calculation and all the information about entire port. The proposed system our application is when user want to store the data about all the information, calculation of accounts, weighment operations user can store all accounts and also include calculator to calculation part. This is the advantage of the proposed system. Modules we can use Database, ui designing, Authentication, Barcode scanner. Android is a software stack for mobile devices that includes an operating system, middleware and key applications. The android sdk provides the tools and API necessary to begin developing applications on the android platform using the java programming language. In this application user can store all the information about ships and loads and also the incoming and outgoing transactions in entire port information.***

Keywords: ***ANDROID, API, AUTHENTICATION, BARCODE, SHIP, SECURITY.JAVA.***

I. INTRODUCTION

The over view of the application is While exporting product details can be store in mobile. We have to scan the barcode of the products and this code to be sent to the receiver through the sms or mail. While importing we can check the details of products from the server. One of the fastest growing industries now a day is mobile industry. There are many competitors in this area who are doing research and development on new platforms & user experience. One such technology is Android from Google which is supported for Google phones. These phones are described as next Generation mobiles As described by Google. Developing application for such mobile phones using the open source android SDK for storing the information. Application that interfaces with this application and extends certain use cases to Android client. The Weighment operations, Remote reporting of events, yard and storage area planning and more and also store the data of all operations application aims at remote data collection into the this application and covered the use cases like Gate Operations. Android is a software stack for mobile devices that includes an operating system, middleware and key applications. The android sdk provides the tools and API necessary to begin developing applications on the android platform using the java programming language. In this application user can store all the information about ships and loads and also the incoming and outgoing transactions in entire port information.

Before going to develop my paper I have briefly studied about android technology. Android technology is one of the latest technologies which are used for developing many mobile applications.

This we can able to develop different types of mobile applications which are portable and secure. Many of the applications can easily run on this technology. My paper is quite different from others I have used some sophisticated technologies for designing my paper.

Android is a software stack for mobile devices that includes an operating system, middleware and key applications. The android sdk provides the tools and API necessary to begin developing applications on the android platform using the java programming language. In this application user can store all the information about ships and loads and also the incoming and outgoing transactions in entire port information.

II. REQUIREMENTS ELICITATION

2.1 EXISTING SYSTEM:

The existing system Is when user want to store the data information about aii ships in port user will registered in a book to maintain all the data and operations timings about load for this they are keeping one accountant he will do accounts calculation and all the information about entire port.

2.2 PROPOSED SYSTEM:

The proposed system our application is when user want to store the data about all the information , calculation of accounts , weighment operations user can store all accounts and also include calculator to calculation part. We can provide the security. This is the advantage of the proposed system.

III. LITERATURE SURVEY

3.1. Technology Description:

The BlackBerry and phones, which have appealing and high-volume mobile platforms, are addressing opposite ends of a spectrum. The BlackBerry is rock-solid for the enterprise business user. For a consumer device, it's hard to compete with the iPhone for ease of use and the "cool factor." Android, a young and yet-unproven platform, has the potential to play at both ends of the mobile-phone spectrum and perhaps even bridge the gulf between work and play. Today, many network-based or network-capable appliances run a flavor of the Linux kernel. It's a solid platform: cost-effective to deploy and support and readily accepted as a good design approach for deployment.

Manuscript received on January, 2013

Arunkumar.R, CSE, Vidya Jyothi Institute of Technology, Hyderabad, AP, india.

The UI for such devices is often HTML-based and viewable with a PC or Mac browser. But not every appliance needs to be controlled by a general computing device. Consider a conventional appliance, such as a stove, microwave or bread maker. What if your household appliances were controlled by Android and boasted a color touch screen? With an Android UI on the stove-top, the author might even be able to cook something.

The Android platform is the product of the Open Handset Alliance, a group of organizations collaborating to build a better mobile phone. The group, led by Google, includes mobile operators, device handset manufacturers, component manufacturers, software solution and platform providers, and marketing companies. From a software development standpoint, Android sits smack in the middle of the open source world.

The first Android-capable handset on the market was the G1 device manufactured by HTC and provisioned on T-Mobile. The device became available after almost a year of speculation, where the only software development tools available were some incrementally improving SDK releases. As the G1 release date neared, the Android team released SDK V1.0 and applications began surfacing for the new platform.

To spur innovation, Google sponsored two rounds of "Android Developer Challenges," where millions of dollars were given to top contest submissions. A few months after the G1, the Android Market was released, allowing users to browse and download applications directly to their phones. Over about 18 months, a new mobile platform entered the public arena.

With Android's breadth of capabilities, it would be easy to confuse it with a desktop operating system. Android is a layered environment built upon a foundation of the Linux kernel, and it includes rich functions.

The UI subsystem includes: a) Windows b) Views

Android includes an embeddable browser built upon Weskit, the same open source browser engine powering the phone's Mobile Safari browser.

Android boasts a healthy array of connectivity options, including Wife, Bluetooth, and wireless data over a cellular connection, for example, GPRS, EDGE, and 3G. A popular technique in Android applications is to link to Google Maps to display an address directly within an application. Support for location-based services such as GPS and accelerometers is also available in the Android software stack, though not all Android devices are equipped with the required hardware. There is also camera support.

An Android application consists of one or more of the following classifications:

- Activities
- Services
- Content Providers
- Broadcast Receivers

3.2 Activities: An application that has a visible UI is implemented with an activity. When a user selects an application from the home screen or application launcher, an activity is started.

An activity is a single, focused thing that the user can do. Almost all activities interact with the user, so the Activity class takes care of creating a window for you in which you can place your UI with set content View (View). While activities are often presented to the user as full-screen windows, they can also be used in other ways: as floating windows (via a theme with window is Floating set) or embedded inside of another activity (using Activity Group). An activity has essentially four states:

1) If an activity in the foreground of the screen (at the top of the stack), it is active or running.
2) If an activity has lost focus but is still visible (that is, a new non-full-sized or transparent activity has focus on top of your activity), it is paused. A paused activity is completely alive (it maintains all state and member information and remains attached to the window manager), but can be killed by the system in extreme low memory situations.
3) If an activity is completely obscured by another activity, it is stopped. It still retains all state and member information, however, it is no longer visible to the user so its window is hidden and it will often be killed by the system when memory is needed elsewhere
4) If an activity is paused or stopped, the system can drop the activity from memory by either asking it to finish, or simply killing its process. When it is displayed again to the user, it must be completely restarted and restored to its previous state.

The following diagram shows the important state paths of an Activity. The square rectangles represent callback methods you can implement to perform operations when the Activity moves between states. The colored ovals are major states the Activity can be in.

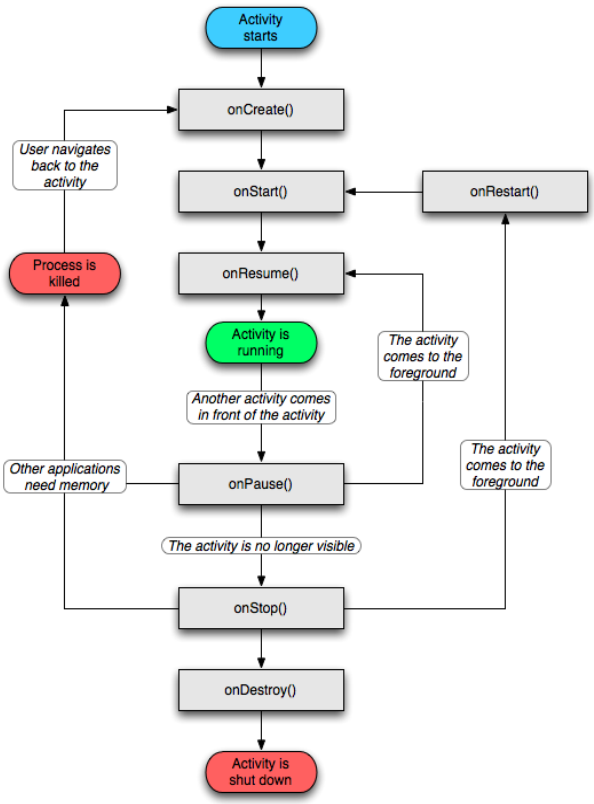

Fig 1 : Activity Life Cycle

3.3 Services: A service should be used for any application that needs to persist for a long time, such as a network monitor or update-checking application. Most confusion about the Service class actually revolves around what it is not. A Service is not a separate process.

The Service object itself does not imply it is running in its own process; unless otherwise specified, it runs in the same process as the application it is part of A Service is not a thread. It is not a means itself to do work off of the main thread (to avoid Application Not Responding errors). Thus a Service itself is actually very simple, providing two main features: A facility for the application to tell the system about something it wants to be doing in the background (even when the user is not directly interacting with the application). This corresponds to calls to Context.startService (), which ask the system to schedule work for the service, to be run until the service or someone else explicitly stop it.

A facility for an application to expose some of its functionality to other applications. This corresponds to calls to Context. Bind Service (), which allows a long-standing connection to be made to the service in order to interact with it. When a Service component is actually created, for either of these reasons, all that the system actually does is instantiate the component and call it's on Create () and any other appropriate callbacks on the main thread. It is up to the Service to implement these with the appropriate behavior, such as creating a secondary thread in which it does its work.

3.4 Content providers: Content providers are one of the primary building blocks of Android applications, providing content to applications. They encapsulate data and provide it to applications through the single Content Resolver interface. A content provider is only required if you need to share data between multiple applications. For example, the contacts data is used by multiple applications and must be stored in a content provider. If you don't need to share data amongst multiple applications you can use a database directly via SQLiteDatabase.

Content providers store and retrieve data and make it accessible to all applications. They're the only way to share data across applications; there's no common storage area that all Android packages can access. Android ships with a number of content providers for common data types (audio, video, images, personal contact information, and so on). You can see some of them listed in the android. Provider package. You can query these providers for the data they contain (although, for some, you must acquire the proper permission to read the data).

3.5 Broadcast receivers: An Android application may be launched to process a element of data or respond to an event, such as the receipt of a text message. An Android application, along with a file called AndroidManifest.xml, is deployed to a device. AndroidManifest.xml contains the necessary configuration information to properly install it to the device.

It includes the required class names and types of events the application is able to process, and the required permissions the application needs to run. Many applications may have this specific permission enabled. Such declarative security helps reduce the likelihood that a rogue application can cause damage on your device.

4. Develop a basic application:

1. Download the java jdk from below URL and follow the below important steps
   http://www.oracle.com/technetwork/java/javase/downloads/index.html.
   Select the Java button, you don't need the Java EE.
2. Install the Java JDK.
3. Download the SDK (not the installer exe, but the zip file) from the above link. Currently you want the android-sdk_r08-windows.zip file.
4. Unpack the zip file to the location you want to have the SDK installed (doesn't much matter where, just remember where you put it). For example
   C:\Users\ which will be used here.
5. Add the ANDROID_SWT user variables:

5.1. Open start menu, enter sysdm.cpl into the text field and press enter.
5.2. In the opened window, select Advanced System Settings on the left.
5.3. Select Environment variables in the lower right corner.
5.4. Press the New button under the User variables windows (the top one).
5.5. Enter ANDROID_SWT in the variable name field and the path to the swt 64-bit SWT file into the variable value field. For example
C:\Users\<YOUR USERNAME HERE>\android-sdk-windows\tools\lib\x86_64.
5.6. Press OK in all the windows to close them.

6. Download Eclipse from http://www.eclipse.org/downloads/, you want Eclipse IDE for Java Developers for Windows 64 Bit. Unpack the eclipse folder to wherever you want to have eclipse installed. Go into the folder and drag the eclipse.exe file to your desktop or start menu while holding the SHIFT key to make a shortcut to Eclipse.
7. Follow the instructions at http://developer.android.com/sdk/eclipse-adt.html to install the Eclipse ADT Plugin. Here is the list from that page:

7.1. In Eclipse go to Help > Install New Software.....
7.2. Press Add in the top right hand corner.
7.3. Write ADT Plugin in the name field and process with SDK https://dl-ssl.google.com/android/eclipse/ in the Location URL. If you get any errors, try changing from https to http.
7.4. In the Available Software dialog, select the checkbox next to Developer Tools, press Next.
7.5. In the next window, you'll see a list of the tools to be downloaded. Click Next.
7.6. Read and accept the license agreements, then click Finish.
7.7. When the installation completes, restart Eclipse.
7.8. After restaing Eclipse, go to Window > Preferences....
7.9. Select Android from the list on the left.
7.10. For the SDK Location in the main panel, click Browse... and locate your downloaded SDK directory.
7.11. Click Apply, then OK.

8. Add the SDK components needed to code for Android (not just the manager which is what you downloaded):

8.1. In Eclipse select Window > Android SDK and AVD Manager.
8.2. Select Available Packages on the left.
8.3. Check the checkbox next to Android Repository to install all versions of the SDK (so you can develop for all versions of Android 1.5 - 2.3).
8.4. Click Install Selected.
8.5. Accept the licenses, press Accept All and then Install.

9. Follow

http://developer.android.com/guide/developing/device.html to set up device (Nexus One):

9.1. Connect you Nexus One to the PC using an USB cable.9.2. When/If Windows says it has failed to install drivers for a device, tell it you have drivers. If windows don't ask you for drivers go to the start Menu and type devmgmt.msc into the field and press Enter. You will get a list of all your devices, find the one with a red mark and right click it. Select Update Driver Software.

9.3. Select browse my computer for driver software.

9.4. Select Browse and select the google-usb_driver folder in the SDK folder you unpacked in step 3.

10. Turn on "USB Debugging" on the Nexus One.

10.1. Press MENU, select Applications > Development, then enable USB debugging.

11. Try to see if your Nexus One is connected by opening command line (go to start menu, enter cmd and press enter). Navigate to the location of your SDK and the platform-tools directory. Run adb devices or just run it directly by using the full path: C:\Users\<YOUR USERNAME HERE>\android-sdk-windows\platform-tools\adb.exe devices. You should see something like:
List of devices attached
HT9DSP500000 device

12. If you already have a project written in Eclipse open the document and check AndroidManifest.xml file, open the Application tab at the bottom and then select true in the Debug gable drop-down box on the left.

12.1. If you do not have a project, follow below useful Url finding
http://developer.android.com/resources/tutorials/hello-world.html to set one up.

13. To run your In Eclipse select Run > Run Configurations...

13.1. Select Android Application on the left and press the New icon.

13.2. Enter whatever you want in the Name field and press browse and select the project you want to run.

13.3. In the target tab. Select Manual.

13.4. Press run. In the dialog select you device and press OK.

This section provides a whirlwind tour of building an Android application. The example application is about as simple as you can imagine: a modified "Hello Android" application. You'll add a minor modification to make the screen background color all white so you can use the phone as a flashlight. Not very original, but it will be useful as an example. Download the full source code.

To create an application in Eclipse, select **File > New > Android project**, which starts the New Android Project wizard.

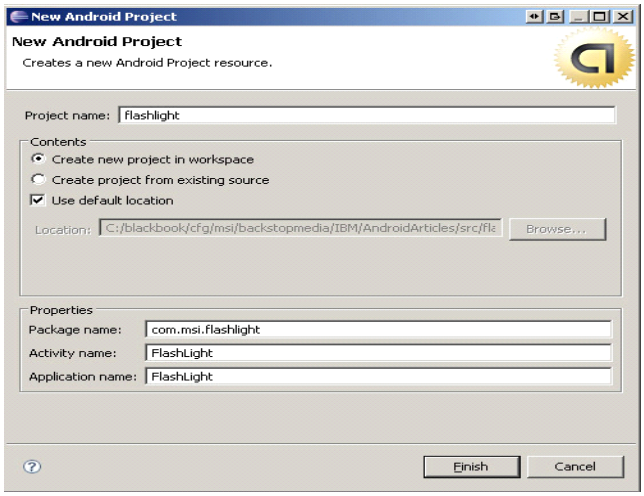

Fig 2 : Creating New Android project

V. REQUIREMENTS SPECIFICATIONS

5.1 FUNCTIONAL REQUIREMENTS:

1. UI Designing
2. Authentication
3. Database
4. Barcode Scanner

1. *UI Designing:* By using this module we can provide awesome UI designing to the user for look and feel purpose. We can use animations also to attract the users. Android provides a variety of pre-build UI components such as structured layout objects and UI controls that allow you to build the graphical user interface for your app. Android also provides other UI modules for special interfaces such as dialogs, notifications, and menus. All user interface elements in an Android app are built using View and View Group objects. A View is an object that draws something on the screen that the user can interact with. A View Group is an object that holds other View (and View Group) objects in order to define the layout of the interface. Android provides a collection of both View and View Group subclasses that offer you common input controls (such as buttons and text fields) and various layout models (such as a linear or relative layout)
2. *Authentication:* From the **Authentication** point of view the user has to be authenticating to make use of application. Every user must be signup by giving their details before using applications to provide security and identification. We can provide the security of the user. We can use only single user we cannot multiple user.
3. *Database:* From the **Database** point of view the user has to be it will store all the data of your records and files user can retrieve the data at any time .We can store the export details and barcode numbers.
4. *Barcode Scanner:* From the **Barcode Scanner** point of view the user has to be scan the barcode number to enter the price of the in to the database. Scan barcodes on CDs, books, and other products, then look up prices and reviews, or search for a word in a book and find where it occurs. You can also scan QR Codes containing URLs, contact info, calendar events, and more. Features: contact sharing, improved QR Code decoding, and added Chinese translation.

5.2 PERFORMANCE REQUIREMENTS

Performance is measured in terms of the output provided by the application. Requirement specification plays an important part in the analysis of a system. Only when the requirement specifications are properly given, it is possible to design a system, which will fit into required environment.

It rests largely with the users of the existing system to give the requirement specifications because they are the people who finally use the system. This is because the requirements have to be known during the initial stages so that the system can be designed according to those requirements. Ti is very difficult to change the system once it has been designed and on the other hand designing a system, which does not cater to the requirements of the user, is of no use.

IV. SUPPORTED DEVELOPMENT ENVIRONMENTS

For developing Android applications, we recommend that you install one of these packages:

- Eclipse IDE for Java Developers
- Eclipse Classic
- Eclipse IDE for Java EE Developers
- Android SDK
- A.D.T Plug-in

6.1 TECHNOLOGIES USED:

- User Interface : LAYOUT-XML
- Programming Language : JDK & SDK
- Debug Tool : DDMS

6.2 SOFTWARE REQUIREMENTS

Operating System : Android, windows7, windowsXP
Programming Language : Java
IDE/Workbench : My Eclipse 3.6
Google App Inventor
Android SDK
Adobe Fireworks/Photoshop
Adobe Flash
Adobe Sound Booth

6.3 HARDWARE REQUIREMENTS:

Pentium processor : 233 MHZ or above

RAM Capacity : 256MB

Hard Disk : 40GB

Device : Android supported mobile

6.4 MOBILE REQUIREMENTS:

Connectivity : EDGE, 3G, And GPS Phone
Memory : Minimum 10MB

Technology : Java(MIDP2.0) supported
mobiles, BlutoothCamera, Audio Rec

VII. TEST CASES

7.1 TEST CASE 1:

Test	Inputs	Actual Output	Obtained Output	Description
Valid Login	Mobile no & pwd	Success	Success	Test passed. Passes the control to the Other module
InValid Login	Mobile no & pwd	Failed	Failed	Test Passed. Displays the alert message

7.2 TEST CASE 2:

Test	Inputs	Actual Output	Obtained Output	Description
Valid Login	Mobile no & pwd	Success	Success	Test passed. Passes the control to the Other Module
InValid Login	Mobile no & pwd	Failed	Failed	Test Passed. Displays the alert message
InValid Mobile no	Mobile no	Failed	Failed	Test Passed. Displays the alert message
InValid Pwd	Pwd	Failed	Failed	Test Passed. Displays the alert message

VIII. RESULTS ANALYSIS

8.1 Create virtual device

Fig 3: create virtual device

8.2 Use the adb shell command

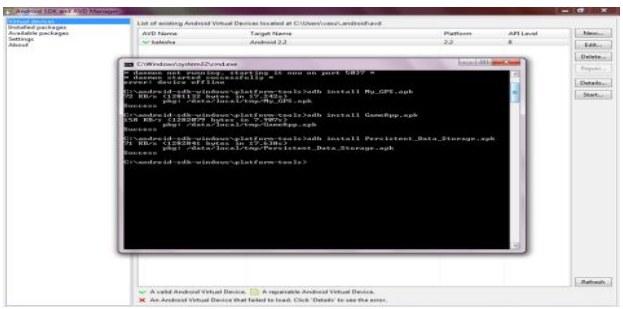

Fig 4: Use the adb shell command

Fig 5: Applications screen shots

8.3 SPLASH SCREEN

Fig 6: splash screen

Fig 7: User name and pwd

Fig 8: Import and scan

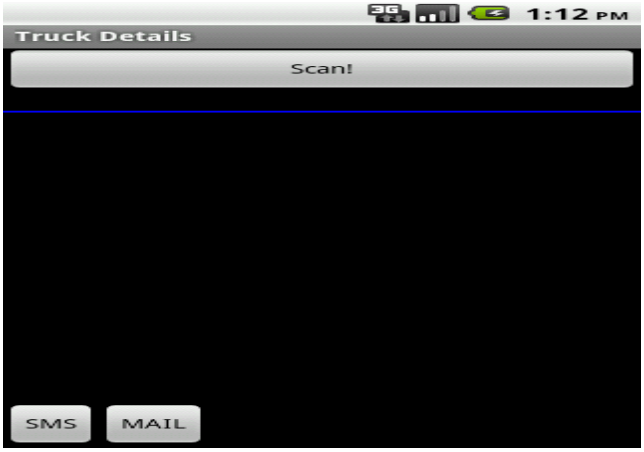

Fig 9: Truck details

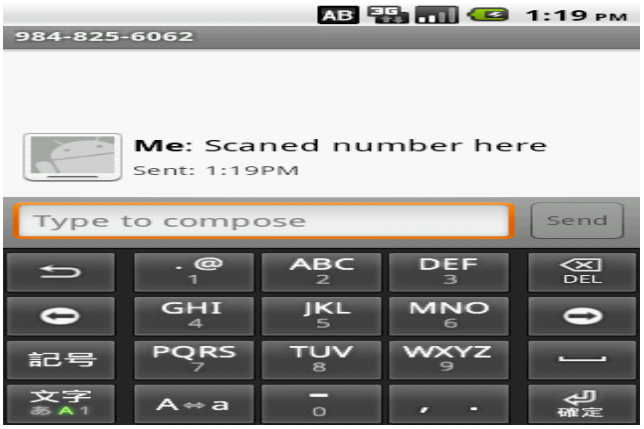

Fig 10: Scaned number

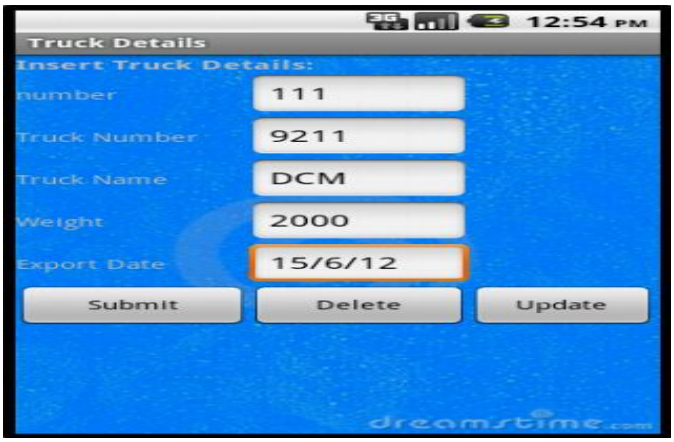

Fig 11: Truck details

Fig 11.1: Truck details

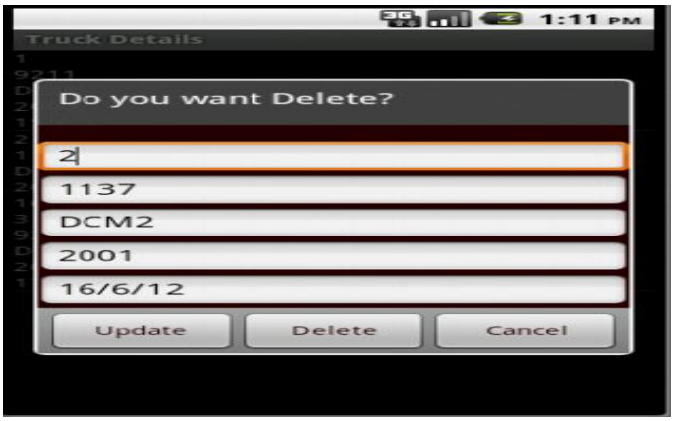

Fig 11.2: Truck details

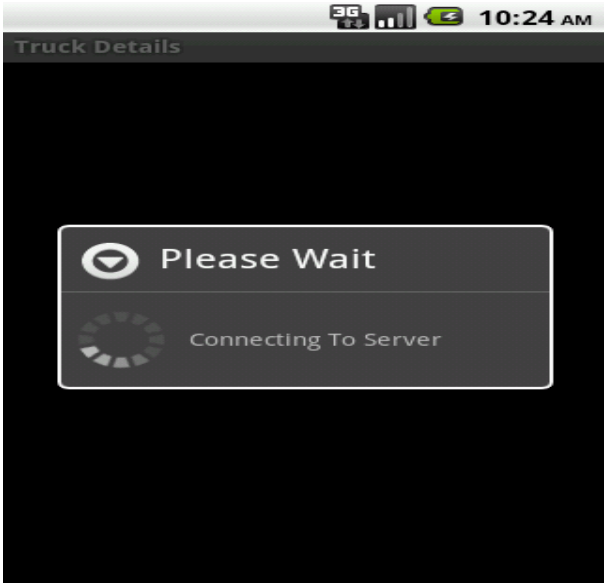

Fig 11.3: Truck details

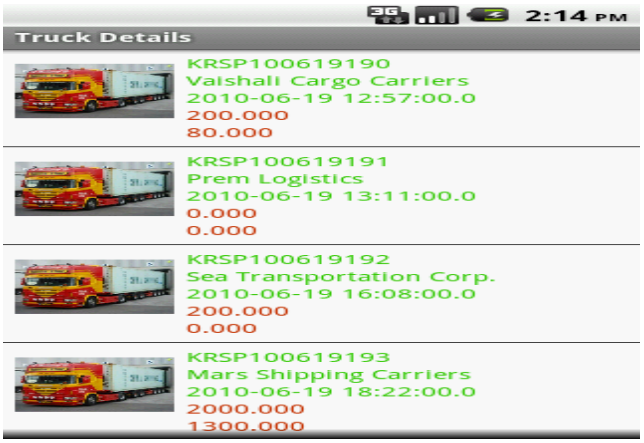

Fig 11.4: Truck details

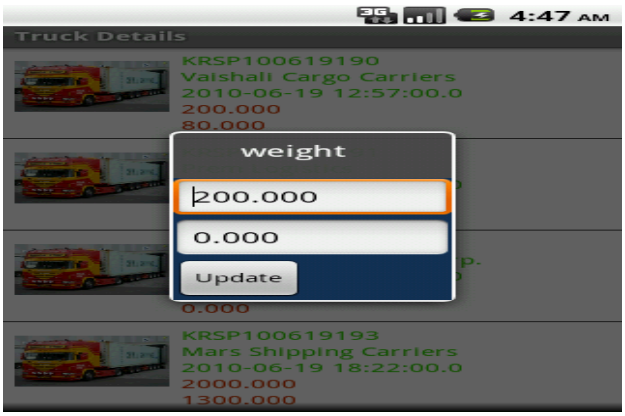

Fig 11.5: Truck details

IX. CONCLUSION AND FUTURE SCOPE

Organizations intending to deploy mobile applications must plan their testing strategy across manual and automation testing approaches for efficient and error-free delivery. In addition to actual device-based testing, emulators should be included as an integral part of the testing program. Enterprise applications require special testing techniques. Outsourcing to vendors who are operating an independent testing practice may be a viable option to manage the expertise, scalability, and quality assurance requirements for mobile application delivery.when entering in to project we will firstly open the login then we will do using barcode scanner scan the barcode number. After exporting the products.

It is not possible to develop a system that makes all the requirements of the user. User requirements keep changing as the system is being used. Some of the future enhancements that can be done to this system are: As the technology emerges, it is possible to upgrade the system and can be adaptable to desired environment. Because it is based on object-oriented design, any further changes can be easily adaptable. Based on the future security issues, security can be improved using emerging technologies.

- This software can run in above 4.0.5 versions
- It can run in Portable as well as Landscape mode
- Based on the future security issues, security can be improved using emerging technologies.
- Based on the user requirement, we can add or upgrade the system to meet user requirement
- If any new technology will come then this system can easily adaptable to that environment.

REFERENCES

1. Hello, Android, E. Burnette, the Pragmatic Programmers (2009).
2. Professional Android 2 Application Development, R. Meier, Wiley (2010).
3. Beginning Android 2, M. Murphy, Apress (2010).
4. Android Wireless Application Development, S. Conder and L. Darcey, Addison-Wesley (2010).
5. Android Application Development in 24 Hours, L. Darcey and S. Conder, Sams (2010).
6. The Android Developer's Cookbook, J. Steele, N. To, Addison-Wesley (2011).

AUTHORS PROFILE

ARUN KUMAR.R, Completed M.TECH in CSE with specialization (COMPUTER SCIENCE ENGINEERING) from CVSR college of engineering affiliated by JNTUH in 2012. Currently he is working as an Assistant professor in CSE department at Vidya Jyothi Institute of Technology, Hyderabad. he is having more than 4 years experience as an assistant professor. His areas of research interests include NETWORKS, ANALYSIS OF ALGORITHM, and COMPILER AND LANGUAGE PROCESSING.