UNIT V -NOTES

UNIT V MOBILE PLATFORMS AND APPLICATIONS

Mobile Device Operating Systems – Special Constrains & Requirements – Commercial Mobile Operating Systems – Software Development Kit: iOS, Android, BlackBerry, Windows Phone – M-Commerce – Structure – Pros & Cons – Mobile Payment System – Security Issues

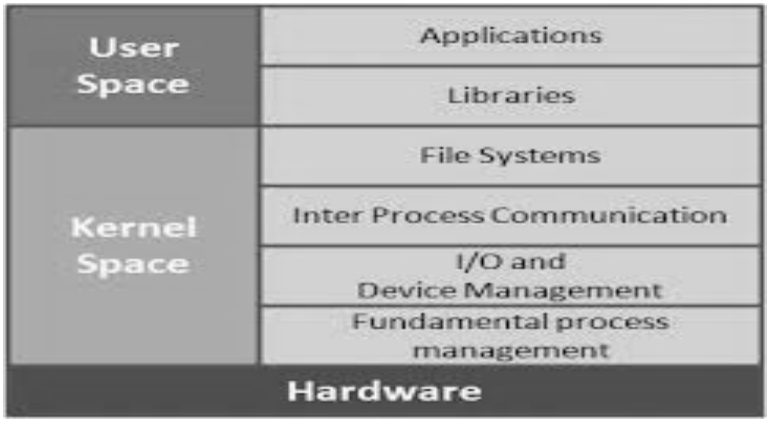
MONOLITHIC DESIGN VERSUS MICROKERNEL DESIGN OF AN OPERATING SYSTEM:

Basic OS services such as process management, memory management, interrupt handling, IO communication, file system, device drivers, networking, etc all run in kernel space. Entire services are loaded on boot up and reside in memory and work is done using system calls. Linux is an example on a monolithic kernel based OS.

Microkernel

The idea behind microkernel OS is to reduce the kernel to only basic process communication and IO control and let other system services run in user space just like any other normal processes. These services are called servers and kept separate and run in different address spaces. Contrary to monolithic OS where services are directly invoked, communication in a microkernel is done via message passing (inter process communication IPC). Mac OS and WinNT are two examples on microkernel OS architecture.

Microkernel Advantages



Monolithic kernel is a single large process running entirely in a single address space. It is a single static binary file. All kernel services exist and execute in the kernel address space. The kernel can invoke functions directly. Examples of monolithic kernel based OSs: Unix, Linux.

In microkernels, the kernel is broken down into separate processes, known as servers. Some of the servers run in kernel space and some run in user-space. All servers are kept separate and run in different address spaces. Servers invoke "services" from each other by sending messages via IPC (Interprocess Communication). This separation has the advantage that if one server fails, other servers can still work efficiently. Examples of microkernel based OSs: Mac OS X and Windows NT.

REQUIREMENTS OF MOBILE OS:

Enables an application to run by simply abstracting the mobile system hardware • Enables the programmer to abstract the devices such that the application need not know full details of the font and font size of the mobile device display • Application need not know how the message will be displayed by the LCD hardware

PalmOS • Windows CE • Symbian • Android─ released in 2008 by Google

Facilitates execution of software components on diversified mobile device hardware • Application need not be aware of the details of the LCD driver and memory at which the CPU will send the message for display

Provides interfaces for communication between processes, threads, and ISRs at the application and middleware layers • Provides middleware for the system hardware • Provides management functions (such as creation, activation, deletion, suspension, and delay) for tasks

Provides memory management • Enables running of processes • Helps the processes in obtaining access to system resources Application tasks • The OS provides the functions used for scheduling the multiple tasks in a system • Synchronization of the tasks by using semaphores (tokens) • A task may have multiple threads Provides for synchronization of the threads and their priority allocation • Accomplishes real-time execution of the application tasks and threads

Mobile OS– An OS which enables running of application tasks taking into account mobile system constraints of hardware and network • Enables a programmer to develop application without considering the specifications, drivers, and functionalities of the hardware of the system

COMMERCIAL MOBILE OPERATING SYSTEM:

WINDOWS OS

A few important features of the Windows mobile OS are the following:

- The Graphics/Window/Event manager (GWE) component handles all input and output.
- Provides a virtual memory management.
- Supports security through the provision of a cryptographic library.
- Application development is similar to that in the Win32 environment. This is considered advantageous since many programmers have knowledge of Win 32-based application development.
- At present, it does not provide true multitasking. An application in the background goes into hibernation and gets active only when it comes to foreground. However, it is expected that Microsoft may support true multitasking in the future versions of the Windows Phone operating system.

SYMBIAN OS

A few other important features supported by the Symbian operating system are given below:

- It supports a number of communication and networking protocols including TCP, UDP, PPP, DNS, FTP, WAP, etc. For personal area networking, it supports Bluetooth, InfraRed and USB connectivity.
- It supports pre-emptive multitasking scheduling and memory protection. Symbian is a microkernel-based operating system.
- CPU is switched into a low-power mode when the application is not responding to an event.
- It is optimized for low-power and memory requirements. Applications, and the OS itself, follow an object-oriented design paradigm.
- All Symbian programming is event-based, and the CPU is switched into a low-power mode when the applications are not directly dealing with an event. This is achieved through a programming idiom called active objects.
- Carbide is an Integrated Development Environment (IDE) toolkit that is available for C++ application development on Symbian OS. It essentially works as an Eclipse plug-in and contains editor, compiler, emulator, libraries and header files required for Symbian OS development. Development kits are available at Nokia and the Symbian Foundation websites.

IOS

iOS is a closed and proprietary operating system fully owned and controlled by Apple and not designed to be used by various mobile phone vendors on their systems. Apple does not license iOS for installation on third-party hardware. However, the overwhelming popularity of iPhone has given iOS a significant market presence. It provided several innovative features that grabbed the market attention. For example, user interactions with OS include gestures such as *swipe*, *tap*, *pinch*, and *reverse pinch*, all of which have specific definitions within the context of the iOS operating system. The other innovative user interactions that are supported by the iOS include internal accelerometers used by some applications for shaking the device as the undo command, rotating the device in three dimensions to switch the display mode from portrait to landscape, etc.

WAP ARCHITECTURE:

- WAP stands for Wireless Application Protocol ,WAP is an application communication protocol
- WAP is used to access services and information ,WAP is inherited from Internet standards
- WAP is for handheld devices such as mobile phones ,WAP is a protocol designed for micro browsers ,WAP enables the creating of web applications for mobile devices.
- WAP uses the mark-up language WML (not HTML) WML is defined as an XML 1.0 application

Figure gives an overview of the WAP architecture, its protocols and components, and compares this architecture with the typical internet architecture when using the World Wide Web. This comparison is often cited by the WAP Forum and it helps to understand the architecture (WAP Forum, 2000a). This comparison can be misleading as not all components and protocols shown at the same layer are comparable For consistency reasons with the existing specification, the following stays with the model as shown in Figure.

The basis for transmission of data is formed by different bearer services WAP does not specify bearer services, but uses existing data services and will integrate further services. Examples are message services, such as short message service (SMS) of GSM, circuit-switched data, such as high-speed circuit switched data (HSCSD) in GSM, or packet switched data, such

as general packet radio service (GPRS) in GSM. Many other bearers are supported, such as CDPD, IS-136, PHS. No special interface has been specified between the bearer service and the next higher layer, the transport layer with its wireless datagram protocol (WDP) and the additional wireless control message protocol (WCMP), because the adaptation of these protocols are bearer specific (WAP Forum, 2000u). The transport layer offers a bearer independent, consistent datagram-oriented service to the higher layers of the WAP architecture. Communication is done transparently over one of the avail-able bearer services. The transport layer service access point (T-SAP) is the common interface to be used by higher layers independent of the underlying net-work.

The next higher layer, the security layer with its wireless transport layer security protocol WTLS offers its service at the security SAP (SEC-SAP).WTLS is based on the transport layer security (TLS, formerly SSL, secure sockets layer) already known from the www. WTLS has been optimized for use in wireless net-works with narrow-band channels. It can offer data integrity, privacy, authentication, and (some) denial-of-service protection

The WAP transaction layer with its wireless transaction protocol (WTP) offers a lightweight transaction service at the transaction SAP (TR-SAP) This service efficiently provides reliable or unreliable requests and asynchronous transactions as explained in section 10.3.4. Tightly coupled to this layer is the next higher layer, if used for connection-oriented service The session layer with the wireless session protocol (WSP) currently offers two services at the session-SAP (S-SAP), one connection-oriented and one connectionless if used directly on top of WDP. A special service for browsing the web (WSP/B) has been defined that offers HTTP/1.1 functionality, long-lived session state, session suspend and resume, session migration and other features needed for wireless mobile access to the web.

Finally the application layer with the wireless application environment (WAE) offers a framework for the integration of different www and mobile telephony applications. It offers many protocols and services with special service access points. The main issues here are scripting languages, special markup languages, interfaces to telephony applications, and many content formats adapted to the special requirements of small, handheld, wireless devices.

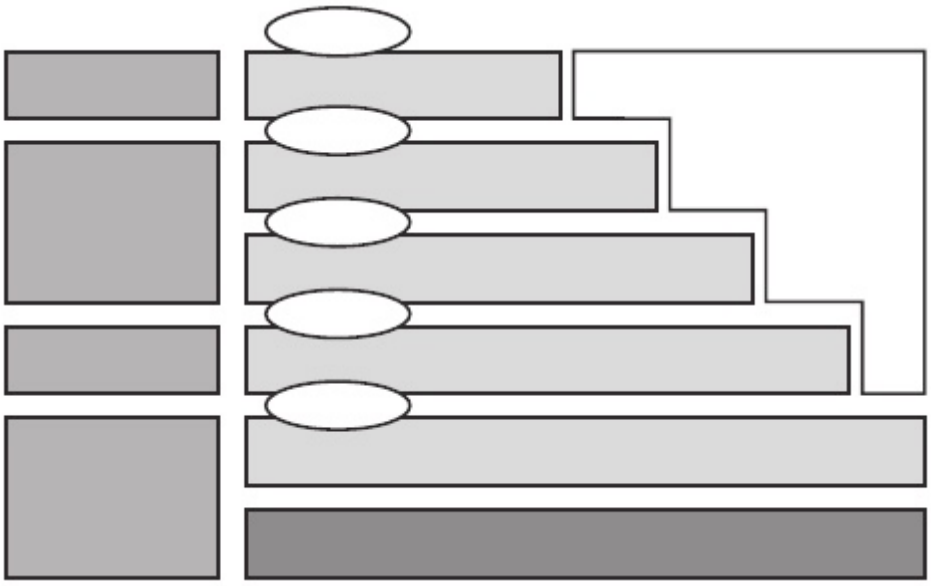
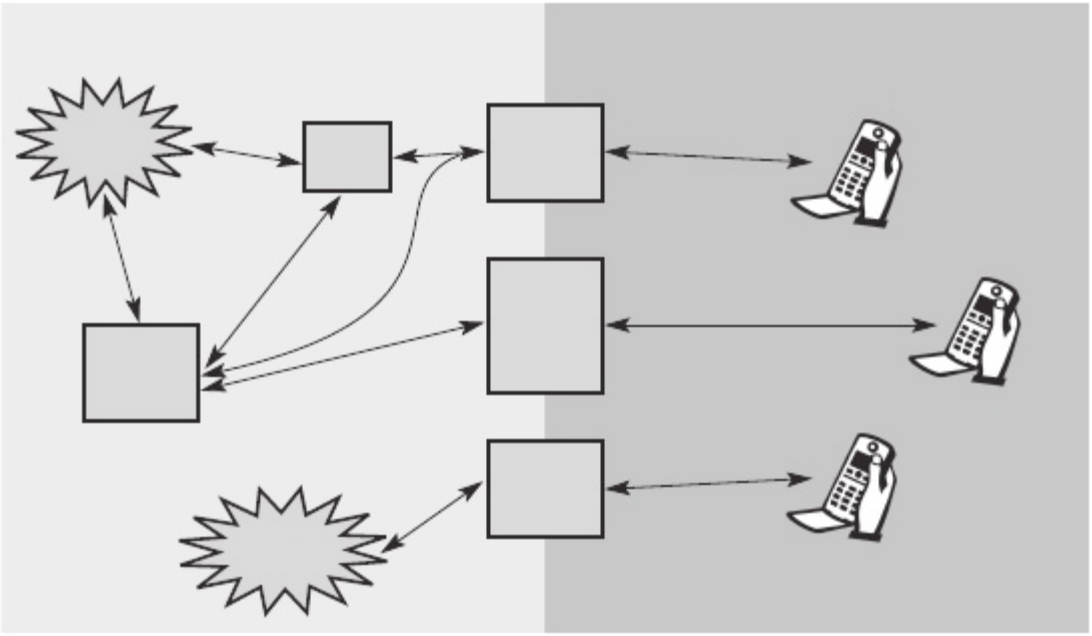


Figure not only shows the overall WAP architecture, but also its relation to the traditional internet architecture for www applications. The WAP transport layer together with the bearers can be (roughly) compared to the ser-vices offered by TCP or UDP over IP and different media in the internet. If a bearer in the WAP architecture already offers IP services (e.g., GPRS, CDPD) then UDP is used as WDP. The TLS/SSL layer of the internet has also been adopted for the WAP architecture with some changes required for optimization. The functionality of the session and transaction layer can roughly be compared with the role of HTTP in the web architecture. However, HTTP does not offer all the additional mechanisms needed for efficient wireless, mobile access (e.g., session migration, suspend/resume). Finally, the application layer offers similar features as HTML and Java. Again, special formats and features optimized for the wireless scenario have been defined and telephony access has been added.

WAP does not always force all applications to use the whole protocol architecture. Applications can use only a part of the architecture as shown in Figure10.9. For example, this means that, if an application does not require security but needs the reliable transport of data, it can directly use a service of the trans-action layer. Simple applications can directly use WDP.

Different scenarios are possible for the integration of WAP components into existing wireless and fixed networks On the left side, different fixed networks, such as the traditional internet and the public switched telephone network (PSTN), are shown. One cannot change protocols and services of these existing networks so several new elements will be implemented between these networks and the WAP-enabled wireless, mobile devices in a wireless net-work on the right-hand side.



The current www in the internet offers web pages with the help of HTML and web servers. To be able to browse these pages or additional pages with hand-held devices, a wireless markup language (WML) has been defined in WAP. Special filters within the fixed network can now translate HTML into WML, web server scan already provide pages in WML, or the gateways between the fixed and wireless network can translate HTML into WML. These gateways not only filter pages but also act as proxies for web access, as explained in the following sections.WML is additionally converted into binary WML for more efficient transmission.

In a similar way, a special gateway can be implemented to access traditional telephony services via binary WML. This wireless telephony application (WTA) server translates, e.g., signaling of the telephone network (incoming call etc.) into WML events displayed at the handheld device. It is important to notice the integrated view for the wireless client of all different services; telephony and web, via the WAE

J2ME:

To make it applicable devices with a specific range of capabilities. Parameters

1.Availability of memory space and memory type

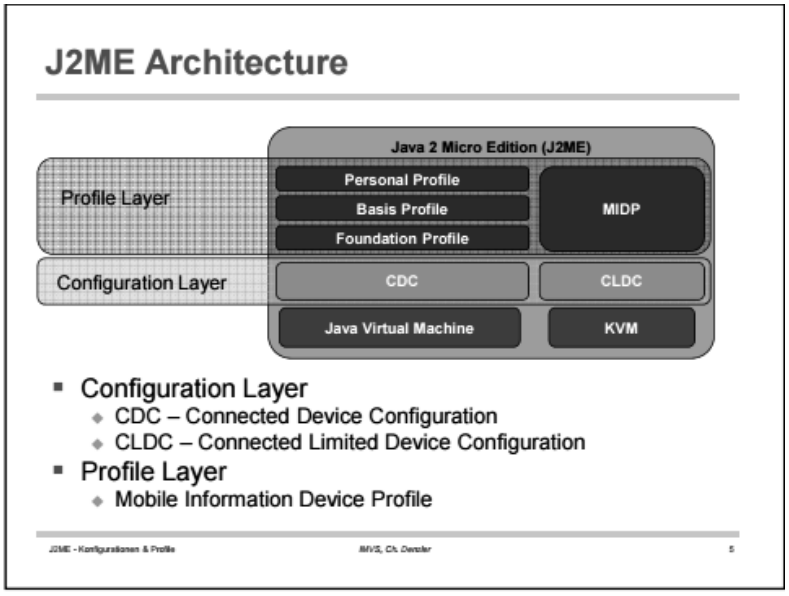
2.Specification of processor in terms of speed and type.

3.Network connectivity of the device

Two configurations

1. Connected limited device configuration for hand held device: It usually have low memory powered by battery, low bandwidth, Network wireless connection. It includes pagers, PDA,Cellphones,Dedicated Terminal, Handheld Devices.
2. Connected Device configuration foe plug-in device

These are high end device used in 32 bit processor ,2 GB memory, implemented by JVM.It included Digital Set top boxes, Home Appliance, Navigation System, Smart Phones



J2ME Connected, Limited Device Configuration (CLDC) Specifies Java environment for mobile phone, pager, and PDA class devices CLDC devices are usually wireless J2ME Connected Device Configuration (CDC) Specifies Java environment for digital television settop boxes, high end wireless devices, and automotive telematics systems. CDC devices may be wired (DTV cable, etc.)

A set of Java based APIs which supplements a Configuration to provide capabilities for a specific vertical market or device type Subject to compatibility tests The specification may not be completely implemented UI and presentation capabilities are generally defined at this layer EX: CLDC => MIDP (Mobile Information Device Profile) CDC => Personal Profile CDC => Foundation Profile

CLDC provides the lowest common denominator for small and resource-constrained devices.

Mobile Information Device Profile (MIDP) Profile for wireless devices implementing CLDC J2ME Foundation Profile Profile for non-GUI networked devices implementing CDC J2ME Personal Basis, Personal, RMI Profiles Basic graphics, next generation PersonalJava environment, and RMI support for CDC & Foundation Profile based devices

ANDROID SDK:

The Android SDK (software development kit) is a set of development tools used to develop applications for Android platform. The Android SDK includes the following:

- Required libraries
- Debugger
- An emulator

Relevant documentation for the Android application program interfaces (APIs)

Sample source code

Tutorials for the Android OS

A software development kit that enables developers to create applications for the Android platform. The Android SDK includes sample projects with source code, development tools, an emulator, and required libraries to build Android applications.

PROCEDURE TO DEVELOP AN APPLICATION USING THE ANDROID SDK:

Android is an Operating System for mobile devices developed by Google, which is built upon Linux kernel. Android competes with Apple's iOS (for iPhone/iPad), RIM's Blackberry, Microsoft's Windows Phone, Symbian OS, and many other proprietary mobile OSes.

setup a working environment using the Android SDK, Eclipse IDE, and ADT Plugin.

To compile your application against a particular version of Android, you must use the SDK Manager to download and install the SDK Platform for that release. Launch Android Studio.

1. Choose "Start a new Android Studio Project".
2. In "Create New Project" dialog □ Set "Application Name" to "Hello Android" (which is the title in your app menu) □ Set your "Company Domain" to "example.com" □ Take note of your "Project Location" □ Next.
3. In "Target Android Devices" dialog □ Check "Phone and Tablet" □ For minimum SDK, choose "API 14: Android 4.0 (IceCreamSandwich)" □ Next.
4. In "Add an activity to Mobile" dialog □ Select "Blank Activity" □ Next.
5. In "Customize the Activity" dialog □ Set "Activity Name" to "MainActivity" □ Set "Layout Name" to "activity_main" □ Set "Title" to "MainActivity" □ Set "Menu Resource Name" to "menu_main" (These are actually the default values) □ Finish.
6. Take a while to set up the "first" app. The app appears after "Indexing..." completes. By default, a hello-world app is created.

ANDROID SOFTWARE STACK AND ANDROID APPLICATION COMPONENTS:

They are four components

- Activity
- Content providers
- Service
- Broadcast receiver

Components	Description
Activities	They dictate the UI and handle the user interaction to the smart phone screen
Services	They handle background processing associated with an application.
Broadcast Receivers	They handle communication between Android OS and applications.
Content Providers	They handle data and database management issues.

ACTIVITIES

An activity represents a single screen with a user interface,in-short Activity performs actions on the screen. For example, an email application might have one activity that shows a list of new emails, another activity to compose an email, and another activity for reading emails. If an application has more than one activity, then one of them should be marked as the activity that is presented when the application is launched.

An activity is implemented as a subclass of **Activity** class as follows −

- public class Main Activity extends Activity {}

SERVICES

A service is a component that runs in the background to perform long-running operations. For example, a service might play music in the background while the user is in a different application, or it might fetch data over the network without blocking user interaction with an activity.

A service is implemented as a subclass of **Service** class as follows −

```
public class MyService extends Service {}
```

BROADCAST RECEIVERS

Broadcast Receivers simply respond to broadcast messages from other applications or from the system. For example, applications can also initiate broadcasts to let other applications know that some data has been downloaded to the device and is available for them to use, so this is broadcast receiver who will intercept this communication and will initiate appropriate action.

A broadcast receiver is implemented as a subclass of BroadcastReceiverclass and each message is broadcaster as an Intent object.

```
public class MyReceiver  extends  BroadcastReceiver {

  public void onReceive(context,intent){}}
```

CONTENT PROVIDERS

A content provider component supplies data from one application to others on request. Such requests are handled by the methods of the ContentResolverclass. The data may be stored in the file system, the database or somewhere else entirely.

A content provider is implemented as a subclass of ContentProvider class and must implement a standard set of APIs that enable other applications to perform transactions.

```
public class MyContentProvider extends  Cont
```

M-COMMERCE:

- A content provider implements on application by providing two sets of programs (i) server side(II) client side
- The client side programs run on the micro Brower installed on the user mobile device
- The server side programs performing data base accesses and computation reside on the host computers

ADVANTAGES AND DISADVANTAGES OF M-COMMERCE:

ADVANTAGES

- For the business organization the benefits of using M commerce includes customer convenient, cost saving and new business opportunities
- It provides flexibility, Anywhere any time shopping using a light weight device
- Mobile device can be highly personalized, They provide additional level of convenience to the customer

DISADVANTAGES:

- Do not generously offer graphics or processing power of the PC
- The small screen of the mobile devices limit the complexity of applications
- Underlying network may impose severer types of restrictions
- Security

ii) APPLICATIONS OF M-COMMERCE

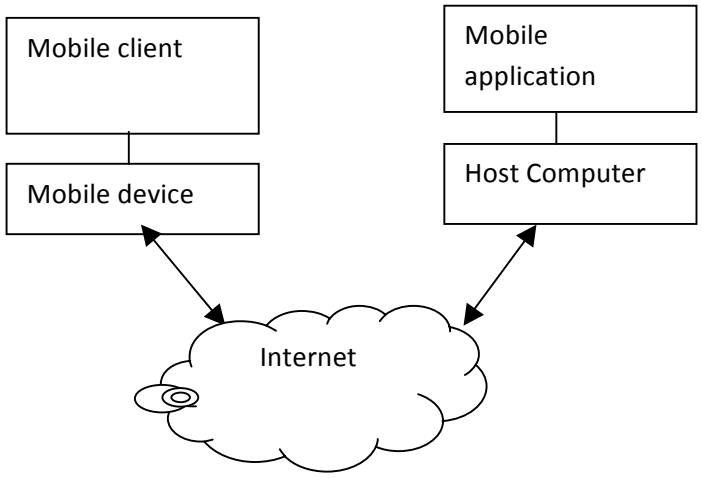
BUSINESS TO CONSUMER APPLICATION:

- Advertisement,
- Comparison Shopping,
- Info about the product,
- Mobile ticketing,
- Loyalty and payment services,
- Interactive advertisement

BUSINESS TO BUSINESS APPLICATION:

- Ordering and delivery conformation
- Stock track-in and control
- Supply chain management
- Mobile Inventory Management

STRUCTURE OF MOBILE COMMERCE



- **MOBILE DEVICES**:

  The users specify the request through appropriate interfaces, which are then transmitted through the internet to the mobile commerce application the internet

- **MOBILE MIDDLEWARE**:

  The main purpose of is to seamlessly and transparently map the internet content to mobile phone they may spot wide variety of OS markup languages, Micro Browsers, Protocols

- **NETWORK**:

  Wireless network are at the core of every M- commerce. User request are delivered either to the closest AP or to a base station

- **HOST COMPUTER**:

  They are powerful servers that process and store all the information's need for mobile commerce application

II) MOBILE PAYMENT SYSTEM:

It is defined as initiation, authorization, and confirmation of a financial transaction using a mobile device.

Payment scheme:

1. Bank account based
2. Credit card Based
3. Micro payment Based

- **Bank account based:**
  - In this scheme the bank account of the customer is linked to his mobile phone numbers. When the customers makes on mobile payment to a vendor through blue tooth or wireless LAN connectivity with a vendor Machine, a bank account of the customer is debited and the value is credited to the vendor account
- **Credit card Based:**
  - In the payment system the credit card number is linked to the mobile phone of the customer. When the customer makes on M payment transaction with the merchant, the credit card is charged and the value is credited to the merchant account
- **Micro payment Based:**
  - It is indented for payment for small purchases such as items from vending machines, the mobile device can communicate with the vending machine directly using Bluetooth or WLAN connection to negotiate the payment and the micropayment is carry out

PROPERTIES OF MOBILE PAYMENT SYSTEM:

- Easy to use
- General purpose
- Interoperability
- Trust
- Cost
- Swiftness
- Global payments

PAYMENT SOLUTIONS:

- SMS Based
- POS based (Physical point of scale)
- Barcode Based
- NFC (Near Field Communication)
- Mobile Wallet