

## **Chapter 1: Theoretical Framework**

### **1.1 Introduction**

Homestay [1] is a popular form of hospitality and lodging whereby visitors sleep at the residence of a local of the city to which they are traveling. The length of stay can vary from one night to over a year and can be free, in exchange for monetary compensation, in exchange for a stay at the guest's property either simultaneously or at another time (home exchange), or in exchange for housekeeping or work on the host's property. Homestays are examples of collaborative consumption and sharing. In cases where money is not exchanged in return for lodging, they are examples of a barter economy or gift economy.

The terms of the homestay are generally worked out by the host and guest in advance and can include items such as the type of lodging, length of stay, housekeeping or work required to be performed, curfews, use of utilities and household facilities, food to be provided, and rules related to smoking, drinking, and drugs.

Hosts-

Hosts may receive monetary compensation and/or housekeeping or work on their property. However, they must be comfortable with others using at least part of their home.

Guests-

Homestays can provide several benefits to guests:

- Savings on lodging costs
- Local perspective and information about the city that is not easily found in guidebooks
- A deeper understanding of the everyday life of the locals
- Opportunities to stay in areas under-served by hotels or hostels
- Opportunities to stay in unique properties such as igloos, cabins, and castles
- Compared to staying in a hotel, a homestay may result in a lower carbon footprint
- In certain cases where students that are studying abroad stay with a family, the host family may play a pseudo-parental role, giving advice and sometimes

supervising students' activities. In some homestays, families act as cross-cultural advisers, helping the students understand and adjust to their new culture.

Homestays may have disadvantages over lodging in hotels and hostels:

- May require additional planning before travel
- Inconvenience due to last minute changes or cancellations by either the host or the guest
- Lodging and sleeping surfaces may be less comfortable and/or have less privacy
- Fewer amenities, such as cleaning services, compared to hotels
- Guests may be required to adhere to a schedule or follow rules set by a host, which restrict freedom
- Lodging may not be close to tourist attractions
- If the guest and host do not get along, the homestay can make a visit to an otherwise pleasant city unbearable
- In cases where the guest must perform a service for the host, the homestay can deplete the amount of time available for sightseeing

Homestays is an android application based app. Android is now nearly nine years old but there are still plenty of people who don't know what Android is.

Android is the name of the mobile operating system owned by American company, Google. Android [2], a Linux-based operating system that is designed primarily for touch screen mobile devices such as smart phones and tablet computers. Initially developed by Android that is Google backed financially and later bought in 2005. It most commonly comes installed on a variety of smartphones and tablets.

Android has a large community of developers for writing applications (apps) and these extend the functionality of devices, written primarily in customized version of the Java Programming language. The open nature of Android has further encouraged a large community of developers and enthusiasts to use the open-source code as a foundation for community-driven projects, and this add new features for advanced users or to bring Android devices officially released running on other operating systems. Android

OS has a built in database named SQLite is an open source SQL database that stores data on a device which will be used to store student name and department name and this app will also be using online server.

Android system is a Linux-based system which use the software stack architecture design patterns. As shown in Figure 1, the android architecture consists of four layers: Linux kernel, Libraries and android runtime, Application framework and Applications. Each lower layer provides a sort of encapsulation, while providing call interface to the upper layers.

- (i) **Application:** Android gives a set of core applications including client, SMS program, calendar, maps, browser, contacts, and so on, all developed in Java.
- (ii) **Application Framework:** The developer is allowed to access all the API framework of the core programs. The application framework simplifies the reuse of its components. Any other app can release its functional components and all other apps can access and use this component by following the security aspects of the framework. It's quite helpful for the developers to substitute the program components with this reuse mechanism.
- (iii) **Libraries and Android Run Time:** The library is divided in to two components: Android Runtime and Android Library. Android Runtime is consisted of a Java Core Library and Dalvik virtual machine. The Core Library provides Java core library with many functions. Dalvik virtual machine is a register virtual machine which makes some specific improvements for mobile device. Android system library is to support the application framework; it is also an important link connecting between application framework and Linux Kernel. This system library is developed in C or C++ language. These libraries can also be utilized by the different components in the android system. They provide service for the developers through the application framework.
- (iv) **Linux Kernel:** The kernel system service provided by android inner nuclear layer is based on Linux 2.6 kernel; operations like internal storage, process management, internet protocol, bottom-drive and other core service are all based on Linux kernel.

The app in this dissertation will be developed using latest Android Studio 3.3. Android Studio is the official Integrated Development Environment (IDE) for Android app. Android Studio is specially designed specifically for android development. It is available for download on Windows, Mac and Linux OS and replaced Eclipse Android

Development Tools (ADT) as Google's primary IDE for native Android application development. Android Studio has built-in GitHub integration.

## **1.2 Android Studio**

Android Studio is the official Integrated Development Environment (IDE) for Android app development [3]. It has powerful code editor and developer tools.

Android Studio offers even more features that enhance the productivity when building Android apps, such as:

- A flexible Gradle-based build system
- A fast and feature-rich emulator
- A unified environment where We can develop for all Android devices
- Instant Run to push changes for running app without building a new APK
- Code templates and GitHub integration to help We build common app features and import sample code
- Extensive testing tools and frameworks
- Lint tools to catch performance, usability, version compatibility, and other problems
- C++ and NDK support
- Built-in support for Google Cloud Platform, making it easy to integrate Google Cloud Messaging and App Engine.

Each project in Android Studio contains one or more modules with source code files and resource files. Types of modules include:

- Android app modules
- Library modules
- Google App Engine modules

By default, Android Studio displays project files in the Android project view. This view is organized by modules to provide quick access to the project's key source files. All the build files are visible at the top level under Cradle Scripts and each app module contains the following folders:

- Manifests: Contains the AndroidManifest.xml file.

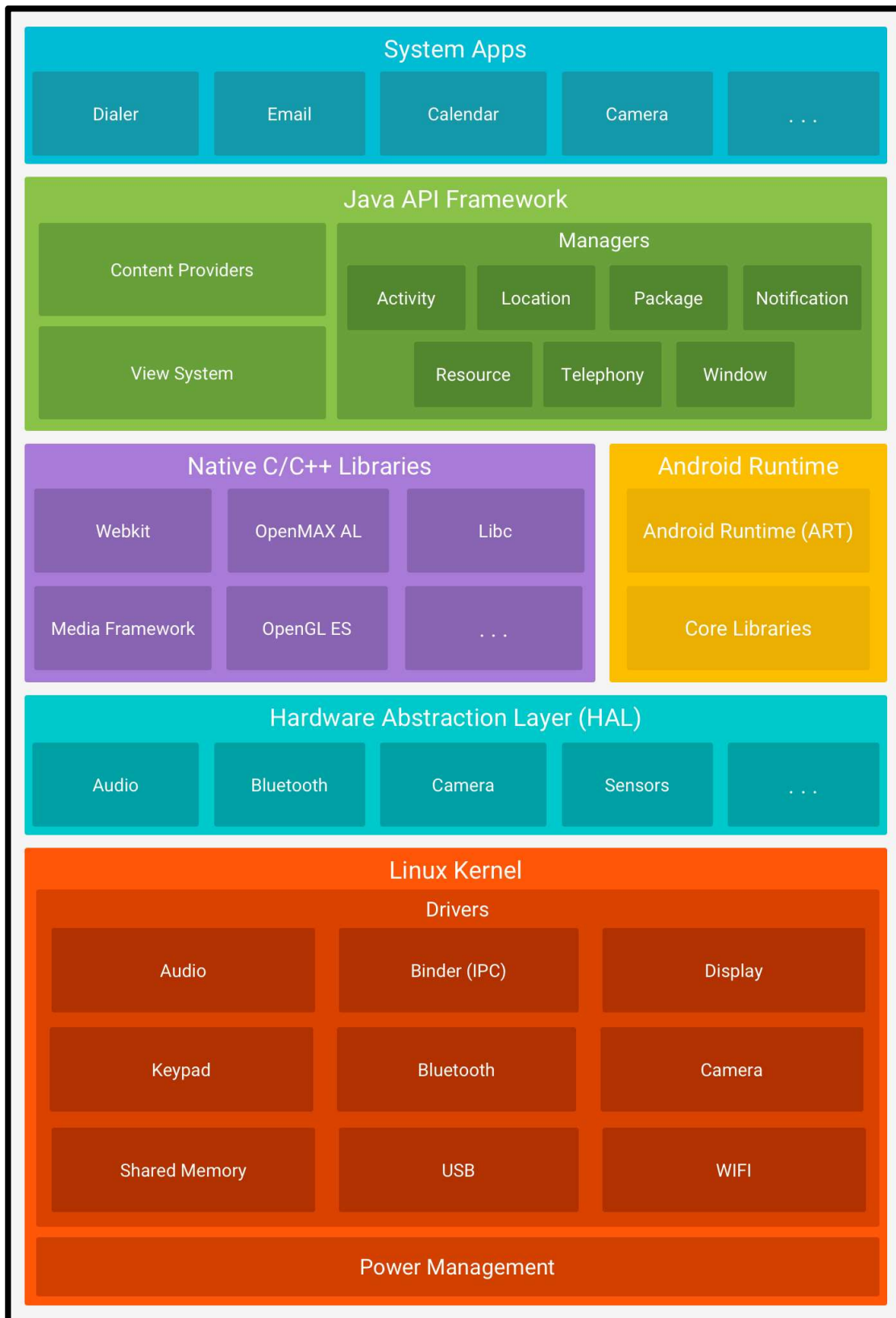


Figure 1.1: Android Architecture Diagram

- Res: Contains all non-code resources, such as XML layouts, UI strings, and bitmap images.
- Java: Contains the Java source code files, including JUnit test code.

The Android project structure on disk differs from this flattened representation. To see the actual file structure of the project, select Project from the Project dropdown.

We can also customize the view of the project files to focus on specific aspects of our app development. For example, selecting the Problems view of our project displays links to the source files containing any recognized coding and syntax errors, such as a missing XML element closing tag in a layout file.

Previous Versions of Android:

- Unnamed (1.0 + 1.1)
- Cupcake (1.5)
- Donut (1.6) –Quick Search Box
- Éclair (2.1) –High Density Displays, Traffic + Navigation
- Froyo(2.2) –Voice Control, Hotspot, Speed
- Gingerbread (2.3) –Simpler, Battery Life, More apps
- Honeycomb (3.0) –Flexible interface, tablets
- Ice Cream Sandwich (4.0) -Customization
- Jelly Bean (4.1) –Google Now, actionable notifications
- KitKat (4.4) –“Ok Google”, voice control variety
- Lollipop (5.0) –fluid tactile screens
- Marshmallow (6.0) –battery life, app permissions, UI,FingerPrint Lock.
- Nougat (7.0) - battery life, smart navigation.
- Android Oreo(8.0)- New screen divider.
- Android Pie.

### **1.3 B2C Business Model**

Business to consumer (B2C) [4] refers to the transactions conducted directly between a company and consumers who are the end-users of its products or services. The business to consumer as a business model differs significantly from the business-to-business model, which refers to commerce between two or more businesses. While most companies that sell directly to consumers can be referred to as B2C companies,

the term became immensely popular during the dotcom boom of the late 1990s, when it was used mainly to refer to online retailers, as well as other companies that sold products and services to consumers through the internet.

Business to consumer (B2C) is among the most popular and widely known of sales models. The idea of B2C was first utilized by Michael Aldrich in 1979, who used television as the primary medium to reach out to consumers. Traditionally, B2C referred to mall shopping, eating out at restaurants, pay-per-view and infomercials. However, the rise of the internet created a whole new B2C business channel in the form of e-commerce or selling goods and services over the internet. Businesses that rely on B2C sales must maintain good relations with their customers to ensure they come back. Unlike business to business (B2B), businesses that rely on B2C must make the consumer have an emotional response to your marketing. In B2B, marketing campaigns are geared to show value of the product or service.

There are typically five types of online B2C business models that most companies use online to target consumers:

- Direct sellers: The most familiar kind of model, where people buy goods from online retail sites. These can include manufacturers or small businesses or simply online versions of department stores that sell products from different manufacturers.
- Online intermediaries: These are liaisons or go-betweens who don't actually own products or services that put buyers and sellers together. Think of sites like Expedia or Trivago [5]
- Advertising-based B2C: This model uses free content to get visitors to a website. Those visitors, in turn, come across digital or online ads. Basically, large volumes of web traffic are used to sell advertising, which, therefore, sells goods and services. An example would be media sites like the Huffington Post, a high-traffic site [6] that mixes in advertising with its native content.
- Community-based: Sites like Facebook [7], which builds online communities based on shared interests, help marketers and advertisers get their products

directly to consumers. Websites will target ads based on users' demographics and geographical location.

- Fee-based: Direct-to-consumer sites like Netflix [8] will charge a fee so consumers can access their content. Sometimes, the site can also offer free, but limited content, while charging for most of it. The New York Times and other large newspapers often use a fee-based B2C business model.

#### **1.4 Firebase**

Firebase [9] is a backend of a web system and it is cloud based server designed to help developers build high-quality apps. Most databases require HTTP calls to get and sync data. Most databases give data only when we ask for it. When we connect our app to Firebase, we're not connecting through normal HTTP. We're connecting through a Web Socket. Web Sockets are much, much faster than HTTP. We don't have to make individual Web Socket calls, because one socket connection is plenty. All of our data syncs automatically through that single Web Socket as fast as client's network can carry it. Firebase sends us new data as soon as it's updated. When client saves a change to the data, all connected clients receive the updated data almost instantly.

##### **Steps for Connecting App to firebase:**

- Create an account on Firebase Login at <https://www.firebase.com/login/> using Google account.
- Create a new application on Firebase, Firebase creates a new application for us when we login for the first time. Also, in the bottom left corner, there is an option to create a new application on the Firebase server. The app URL has to be unique among all applications deployed on Firebase.
- Add Firebase as a project dependency Append the following lines to the build.gradle file, which is located in our app's project folder, and NOT the root folder. Dependencies
  - ✓ `{ compile 'com.firebase:firebase-client-android:2.5.0+' }`
  - ✓ Make sure to sync the application after adding any dependency. If We are getting a build error complaining about duplicate files We can choose to exclude those files by adding the packagingOptions directive to build.gradle file: `android { ... packagingOptions {`



```
exclude 'META-INF/LICENSE'      exclude 'META-  
INF/LICENSE-FIREBASE.txt'      exclude 'META-  
INF/NOTICE'}}
```

- Add permissions to Android application Add network permission to app, the same way we did for parse. Add the following line to AndroidManifest.xml file: `<uses-permission android:name="android.permission.INTERNET" />`

## **1.5 Java**

Android Applications are developed using Java language. Java is object oriented programming language. Java is very popular programming language owned by Oracle. Developed long after C and C++, Java incorporates many of the powerful features of those powerful languages while addressing some of their drawback. Still, programming languages are only as powerful as their libraries. These libraries exist to help developers build applications. Some of the Java's important features are-It is easy to learn and understand. It is designed to be platform-independent and secure, using virtual machines.

Android relies heavily on these java fundamentals. The Android SDK includes many standard Java libraries like data structures libraries, math libraries, graphics libraries, networking libraries and everything else a developer wants as well as special Android libraries that will help develop awesome Android Applications. However, now the official language for developing android apps is Kotlin.

## **1.6 SQLite Database**

SQLite is an Open Source database. SQLite supports standard relational database features like SQL syntax, transactions and prepared statements. The database requires limited memory at runtime (approx. 250 KB) which makes it a good candidate from being embedded into other runtimes [10]. SQLite supports the data types text (similar to String in Java), integer (similar to long in Java) and real (similar to double in Java). All other types must be converted into one of these fields before getting saved in the database. SQLite itself does not validate if the types written to the columns are actually of the defined type, e.g. we can write an integer into a string column and vice versa. SQLite is an embedded SQL database engine. Unlike most other SQL databases, SQLite does not have a separate server process. SQLite reads and writes directly to

ordinary disk files. A complete SQL database with multiple tables, indices, triggers, and views, is contained in a single disk file.

SQLite is a compact library. With all features enabled, the library size can be less than 500KiB, depending on the target platform and compiler optimization settings. (64-bit code is larger. And some compiler optimizations such as aggressive function inlining and loop unrolling can cause the object code to be much larger.) If optional features are omitted, the size of the SQLite library can be reduced below 300KiB. SQLite can also be made to run in minimal stack space (4KiB) and very little heap (100KiB), making SQLite a popular database engine choice on memory constrained gadgets such as cellphones, PDAs, and MP3 players. There is a tradeoff between memory usage and speed. SQLite generally runs faster the more memory we give it. Nevertheless, performance is usually quite good even in low-memory environments.

### **1.7 Payment Gateway**

A Payment Gateway [11] is a critical infrastructural component to ensure that such transactions occur without any hitches and in total security over electronic networks. A Payment Gateway is the access point to the national banking network. All online transactions must pass through a Payment Gateway to be processed. In effect, Payment Gateways act as a bridge between the merchant's application and the financial institutions that process the transaction.

A Payment Gateway authenticates and routes payment details in an extremely secure environment between various parties and related banks. The Payment Gateway functions in essence as an encrypted channel, which securely passes transaction details from the buyer's Personal Computer (PC) to banks for authorization and approval. On gaining the approval, the Payment Gateway sends back the information to the merchant thereby completing the order, and providing verification.

A Payment Gateway is immensely justifiable on account of the multiple benefits it offers including:

- The obvious 24x7 convenience.
- Real time authorization of credit/debit cards.
- Efficient transaction processing.
- Multiple payment options.

- Secure flow of transaction details among buyers, sellers and financial institutions.
- Flexible, powerful real-time reports generation.
- Multi-currency settlements, if the need be facility for customer refund.
- Merchants can get rid of large databases, extensive processing and complex. Software.
- Secure servers.
- Collection of bulk data in a cost-efficient manner, with the additional benefit of being checked for card validity.
- Ability to provide value-added services to merchants, acquiring and issuing banks.
- Provision for multiple host interfaces.
- High security measures to gain customer and merchant trust.

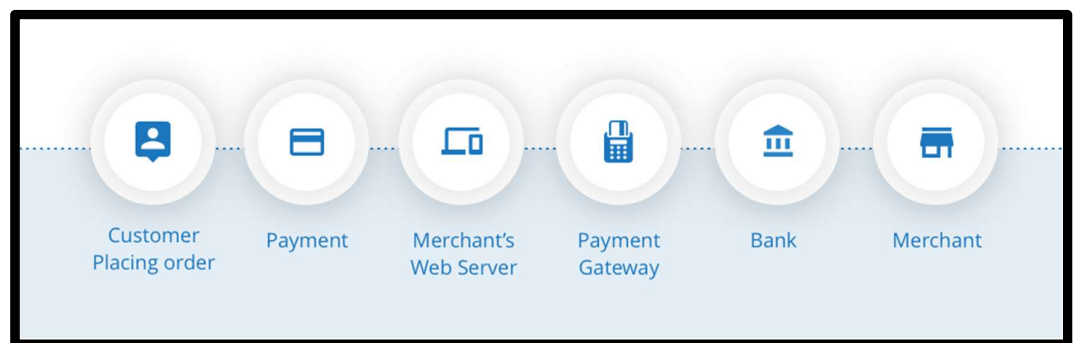


Figure 1.2 Stages in Online Payment.

Whenever a customer buys something from a virtual shopping mall, the Payment Gateway comes in the picture for the following functions:

- Authorizing – Verifying the buyer's credit/debit card details
- Clearing – Transferring the transaction to merchant's bank
- Reporting – Recording all transactions

Steps Involved in a Payment Gateway Transaction:

Step 1 – Consumer visits a shopping website and selects the goods or services and clicks on the Buy button. A message is sent to the website regarding the consumer's desire to buy and make payment.

Step 2 - The Web store's server, after receiving the message from the buyer, adds its digital certificate to identify the mall. This message is now called a Digital Order and also includes the consumer's IP address and transaction amount. The Digital Order is now sent to the Payment Gateway over a secure network. Security is ensured by data encryption.

Step 3 - Based on the Digital Certificate, the Payment Gateway authenticates the web store.

Step 4 - The Payment Gateway offers various payment options on a screen to the buyer.

Step 5 - Buyer chooses the desired payment option, which is transmitted via the secure link to the Payment Gateway.

Step 6 - The Payment Gateway sends the payment details to the acquiring bank (in case of card transactions) or seller's bank (as termed for other instruments).

Step 7 - The acquiring bank sends the information to the buyer's issuing bank (in case of card transactions) or buyer's bank (as termed for other instruments) over a secure link.

Step 8 - Based on the credit limit and the payment instrument's validity, the issuing bank either accepts or rejects the transaction. The confirmation/rejection message is transmitted to the Payment Gateway through the acquiring bank.

Step 9 - The Payment Gateway then transmits digital receipts to the shopping site as well as the buyer.

Step 10 – The web store can ship the goods/services to the buyer.

As opposed to the lengthy offline process, the online version may at the most require 30-40 seconds. It is hard for banks and financial institutions (more so microfinance organizations) with limited capital expenditure budgets to develop a Payment Gateway on their own. It makes more sense for them to look towards third-party providers. Therefore, a national-level Payment Gateway will be the ideal solution for them, with government ownership. Through a Payment Gateway, one bank can connect with another, making it easier to effect international transactions. There would be several benefits to the common man as shown below:

Benefits for the Common Man:

- Payment system within easy reach of customers, who can make payments from the comfort of their homes
- Ability to access multiple counters from one location
- Reduced time for making transactions
- Maximum security for transactions
- Ultimate fallout could be reduced prices of goods and services due to lower number of intermediaries and savings passed on by the merchant for not maintaining a shop infrastructure
- Lower prices could attract new consumers hitherto reluctant to let go of traditional transaction modes, leading to a greater population of Internet-savvy people in India
- Greater reach to the rural areas and interiors of a vast nation like India
- Greater willingness of service providers to offer their products/services in rural areas due to the cost benefit and infrastructure availability.

### **1.8 Problem Definition**

To develop an android application for Homestays of Sikkim.

### **1.9 Objective**

The objectives of this dissertation is to-

- To provide a platform to Homestays of Sikkim.
- To develop an Android Application.
- To implement every possible Online payment option (excluding cryptocurrency).
- To provide location of homestays.
- To provide all the features and offerings of homestays
- To get the No. of days out of Date selected for staying and calculating Fare.
- To give the best UI design like others Hotel Booking apps.
- To design highly efficient Admin Section in Same App.

### **1.10Scope**

Owing to the popularity of Android, Mobile Apps development industries are considering Android Application Development as one of the best remunerative business opportunities. Homestays of Sikkim will get an Android platform for growing and setting up their business. One of the main reasons for this is that software as a service is highly cost effective. It provides ease of use to users, generates revenue and android support backed by Google. International economy is being increasingly electronized, with more and more transactions occurring online and electronic payments increasing significantly to the extent that they are surpassing paper-based payment instruments. In addition, governments worldwide, particularly in India, are targeting electronic delivery of public services and banking activities. Also, the Internet is being tapped for servicing the rural populace due to significant cost benefits and the reach afforded. However, to facilitate increasing volumes, security and efficiency are essential and so is the required infrastructure.