Chapter 1: Introduction and Requirement Analysis
Introduction

Java technology has evolved from a programming language designed to create machine-independent embedded systems into a robust, vendor-independent, machine-independent, server-side technology, enabling the corporate community to realize the full potential of web-centric applications.

Java began with the release of the Java Development Kit (JDK). It was obvious from the start that Java was on a fast track to becoming a solution to the problems of many corporate systems. More interface and libraries were extended in the JDK as the corporate world demanded—and received—application programming interfaces (API) that addressed real-world situations.
JDK API extensions fully integrated into the JDK with the release of the Java 2 Standard Edition (J2SE) of the JDK. J2SE contains all the APIs needed to build industrial strength Java applications. However, the corporate world felt J2SE lacked the strength required for developing enterprise-wide applications and for servicing the needs of developers of mobile and embedded systems.

Again the corporate community pushed Sun Microsystems, Inc. to revise Java technology to address needs of an enterprise. Sun Microsystems, Inc. then launched the Java Community Program (JCP) that brought together corporate users, vendors, and technologists to develop a standard for enterprise Java APIs. The result is the Java 2 Platform Enterprise Edition, commonly referred to as Java 2 Enterprise Edition (J2EE), and the Java 2 Micro Edition (J2ME).

Enterprise systems traditionally are designed using the client/server model, where client-side systems request processing from service-side systems. However, enterprise systems were undergoing their own evolution. A new model called Web services gradually replaced the client/server model in corporations.

Application programmers assembled applications from an assortment of processing components called Web services. Each Web service was independent from other Web services and independent from applications. A client-side application communicates with a middle-tier, server-side application, which in turn interacts with the necessary Web services that are also located on the server side.

With the adoption of the Web services model in corporations, the JCP realized that J2ME must also go through another evolutionary cycle. With the introduction of new specifications, the Java community has merged J2ME technology with Web services technology.

In addition to the acceptance of Web services, corporations are also seeking to merge mobile technology such as Personal Digital Assistants and cellular phones with corporate mainstream applications. J2ME, with the new PIM API, enables developers to create sophisticated, wireless applications that have direct access to native PDA databases. This enables corporate executives to use corporation’s PDA systems to interact with data maintained by PDA native applications.
The new web-centric corporation is changing the way in which it delivers highly efficient, enterprise-wide distributive systems to meet the round-the-clock instantaneous demand expected by thousands of concurrent users—anywhere at any time. The old way of building enterprise systems won’t solve today’s corporate IT requirements.

Technologists at Sun Microsystems, Inc. and the Java Community Program rewrote the way developers build large-scale, web-centric distributive systems by using Java 2 Enterprise Edition (J2EE), and Java 2 Micro Edition (J2ME). J2EE addresses complex server-side issues faced by programmers who develop these systems while J2ME addresses the need to create mobile and embedded components that makes these systems accessible from other than the desktop.

Nearly every J2ME application that you develop requires a way for a user to interact with it unless the application is an embedded closed system. An embedded closed system such as those that control an automobile’s engine doesn’t require input from the user but instead receives input from electro-mechanical devices.

A user interface for a J2ME application is similar to yet different from a user interface that you find on a desktop application. They are similar in that both display options available to the user and then receive and process the option selected by the user. However, a J2ME user interface is less sophisticated than those found on a desktop application because of the limited resources (i.e., screen size) that are available on a J2ME device (i.e., cellular phone).
At the center of nearly every J2ME application is a repository of information that is accessed and manipulated by both service-side components, such as Web services, and client-side applications. A repository is a database management system that stores, retrieves, and maintains the integrity of information stored in its databases.

A J2ME application uses Java data objects, JDBC, and other technology that is necessary to interact with a database management system to provide information to the J2ME application.

Many corporations have practically made PDAs the de facto standard as a mobile communicator, especially since PDA and cell phone technologies have merged, causing a blur between PDAs and cell phones. That is, a PDA can be used as a cell phone and cell phones have incorporated PDA applications. Until recently, J2ME applications lacked the capability to interact with native PDA databases such as those used to store calendar, to-do list, and address information.

The Java Community Process released a new Personal Information Manager (PIM) API, which is used to develop sophisticated J2ME applications. This enables J2ME applications to interact with the J2ME device’s personal information database, which is used by the device’s address book, notepad, and calendar applications.

The glue that enables J2ME applications to interact with external applications, including server-side components, is networking capabilities. Through this project I try to implement routines that take advantage of a J2ME device’s network features to open communications with other applications using a hard-wire or wireless network connection.
Web services in J2ME are services which software building blocks which are available on a network from programmer's side can efficiently create large-scale distributive systems. Web services increase the functionality of your J2ME application beyond the limited resources found on a J2ME device.
The Project requirements is having some more facilities and specifications of application that may or may not available in the market which by having a good developing on these cases, we will get a good functionality on those applications which by applying them, we will receive some more good and needed facilitation on it.

As the requirement of having a Contact/Phone Book on a mobile device on which type we can have developing a java software that we will discuss later on, except of designing and coding of this type of application, the needed of an accessing the Phone Book to the certain network address or URL or server is important that if the mobile user have some contact on the specific website or specific address, can access to it and by receiving data from server and have a telephone or connection to that number.

So one of the applications which the market will need it in future of mobile technology will be such application and nowadays, because the internet connection on mobile handset through the different technology is working on most of the country, the requirement of this application is feeling.

The other application which I came to know is the lack of the most of game producer in the market, that, is not available on the most of the games, is having a game which through that, we can inform our friends, when we are going to play a particular game, so on, our friends will come to know about the starting of it, they will start, and finally, every players, without any access to the network during their gaming, after they finish it up, their scores will send to the certain server. Server or certain website, collect all the scores and sort them, so all together according to their scores, they will find out who got the highest score.

Along with this facility, the other facility such as having a photograph of the game player and behalf of the game score, the photo of the game player upload and send to the server also in most of games in the market, currently is not available, which I think that we can have it by developing a different types of game so bring a special facility on it.
The other requirement of developing a good and innovate project is having a different game size screen, by changing the game on different appearance also is the requirement of the new generation of the games and applications.

All these requirements according to my research in the market and on the current application of different wireless device is felt and should try to fill this lacks.