Developing a Mobile Application to be used in Mineral/Geological Exploration with Emphasis on Geochemical Soil Sampling

Albert Kwansah Ansah[1], *Member, IAENG* and Gill Windall[2]

***Abstract*—This paper presents a mobile application that could be used during geological field exploration placing emphasis on Geochemical Soil Sampling technique to capture data. Geological Soil Sampling Field data recorded digitally benefits from quick processing, hence facilitating the mineral exploration task and reducing exploration duration. This paper describes a prototype application developed with NetBeans IDE 6.7/J2ME Wireless Toolkit 2.2 emulator platform, which could run on any MIDP compatible devices.**

Index Terms—Geochemical Soil Sampling, MIDlet, Mining/Mineral Exploration, Mobile Device Application, Personal Digital Assistant (PDA), Wireless Communication.

1. Introduction

Geological exploration plays an integral part in any successful mining and mineral company. The technology that is employed in geological exploration has an important effect on the overall productivity. The employment of Mobile Information Technology (*MIT*) in an organization can have a significant effect on overall objectives and mineral industries are no different. The adoption of the use of portable mobile devices *(Personal Digital Assistant-PDAs)* and mobile communication into geological/mining exploration may project general objectives. Mobile Information Technology (*MIT*) in mining and geological Solutions cannot be over emphasized in this modern technological age and this goes a long way to facilitate the realization of business goals.

However, the adoption of mobile *IT* in the field of exploration geology could have dangerous consequences to the business strategy as discussed below in the areas such as exploration database solution. The already existing and planned exploration database may need to be adjusted to accommodate the new technology and this could lead to redesigning of the database entirely and therefore high investment with no guarantee of success in its implementation i.e. an investment risk. Other affected areas may be the industry's Local Area Network (*LAN*) and the infrastructure the *LAN* is sitting on. All of these may have to be upgraded to meet the technical challenges offered by mobile *IT*.

This paper seeks to marry mining exploration and Mobile *IT*. Mobile exploration can go a long way to increase productivity by reducing the amount of time data takes to get to processing laboratories with slightly increased production cost.

[1]Department of Computer Science and Engineering, University of Mines and Technology, Box 237 Tarkwa Ghana, afkansah@geologist.com Tel: +233 634 518866
[2]Computing and Mathematical Science Department, University of Greenwich, London SE10 9LS G.F.Windall@gre.ac.uk

The production cost may include devices and software procurement, training of personnel to use the devices and the specified application and the application running cost (purchasing of licensing, buying upgrades), not forgetting cost for maintaining and repairing these hardware devices, and disposing off worn out devices etc.

This paper looks at developing a Mobile Application (*MIDlet*), a prototype, which will be of use in mining exploration with an emphasis on geochemical soil sampling. The application accepts geochemical soil sampling data; and stores it and periodically makes network connection to upload captured data unto processing servers.

Information Technology has found its way quite recently into mining and construction. Once a prospect has been identified, and the right to explore is acquired, assessing it involves advancing through a progressive series of definable exploration stages. The application described here could be used in these exploration stages. [4]

With the widespread use of Information Technology (*IT*) and the emergence of Mobile Information Technology (*MIT*) and Communication, their adoption into Geological Solutions and Mining cannot be over emphasised. Therefore, it has become necessary that appropriate applications be developed to solve the problems encountered in the traditional way of mining exploration, hence increasing throughput.

Data captured and entered via this application is uploaded to the company's website for processing, making use of the Operating System (*O/S*) on the mobile device that the MIDlet sits on and http connection. One good thing about the employment of mobile *IT* into geological and mining exploration is that, the *ore body* is delineated just as the exploration completes because the processing centre is updated with data at the finish of work each day and as the exploration process progresses.

2. Material and Methods

Mobile Devices and Mobile Devices Software:

Mobile devices range from Pagers, Personal Digital Assistants (IPAQs, Black Berry etc), and Palmtops through to Notebooks and Laptops etc including diverse kinds of cell phones. Most devices have communications ability through Wi-Fi or the cell-phone network and can thus access the Internet, Intranet and/or Extranet. [26]

Mobile devices pose unique design and technical challenges to manufacturers since they become pervasive personal belongings. The following features are found with successful mobile devices:

- Small size
- Rich multimedia presentation capabilities
- Long battery life

ISBN: 978-988-17012-0-6
ISSN: 2078-0958 (Print); ISSN: 2078-0966 (Online)

- Large memory for data and application
- Fast response
- Fashionable and personalizable

Manufacturers have to spend a great deal of time and invest lots of money to produce competitive chips and handset designs. Devices used for mining and geological solutions such as the iPAQs have all the above capabilities. Mobile device software may be of two kinds; the underlying operating system and the applications that run on top of the operating system. Just like desktop PCs, mobile devices also require operating systems to run mobile applications. Examples include Palm OS, Symbian OS, Embedded Linux, Windows CE, Android and vendor/device specific etc. Mobile Applications are developed on platforms, which provide advanced set of development tools and features; e.g. Java Micro Edition, Microsoft.NET CF (*Compact Framework*) and WAP Micro Browser. Java ME mobile application platform is one of the most widely used and is available of a very wide range of mobile devices. The application presented in this paper could be developed using any of the platforms mentioned above. [1]

There is however, proprietary mobile software solutions for mobile or handheld devices developed with one or more of the platforms discussed above and made available on the market today at relatively competitive prices. These software or applications are more focused on information gathering or data logging rather than providing data processing which is mainly done outside of the field. [5]

Databases of Geological field information are now becoming constructive on the field. However, there are opposing needs that challenge the design of such databases. The challenges are; the individual need to maintain flexibility of database structure and contents to accept unexpected field situations. There is also a need to cooperate and maintain compatibility between distinct field databases to accept and accommodate their interoperability. These means the design of field information applications remains a daunting and complicated task. Another challenge is the subjective nature of observed information. Geological field exploration application must balance these two important needs by exploiting a domain ontology developed for information. The domain ontology must be the one that satisfies and enables field database flexibility and also facilitates compatibility and may comprise cartographic, geospatial, and geological objects. The ontology driven approach to Geological field applications may result in improved usability due to a user interface which is based on the geological components. [6]

3. Mobile Devices used in Mining/Geological Exploration:

Mobile *IT* in Mining and Construction goes a long way to facilitate the realization of business goals. Geologists on geological mapping exercises, for example, visit discrete sites in a field area and traditionally record observations into paper notebooks. With the adoption of computer-based field systems (e.g. *PDAs*), field data is digitally recorded in an on-going fieldwork and ideally facilitates the entire exploration work in a preparatory bid for geochemical soil sampling. Certain technologies that are significant to field systems may include Mobile Computing and computer aided drafting. [6] Latest development in communication technologies during recent years has opened a new era to Geographical Information Systems and Geoinformation technologies. The era is characterized by mobile data gathering in the field. It could even go beyond just data gathering to processing and presentation in-situ. Mobile geoinformation technologies are particularly based on mobile computers; wireless communication etc. [5] Portable devices are becoming increasingly useful in construction and the mining industries like any other business discipline. This is due to the relatively cheap and affordable small computer systems and mobile devices. The use of these devices have gone a long way to contributing to the ability and potential of geoscientist and geologist to handle and manage large and complex data such as soil geochemistry sampling data both on the field and in the office. Recently, computer aided drafting (*CAD*), database software and *GIS* (Geographic Information System) are in common use for geological data management and for that matter geologic field information are now processed as quickly and fast as possible and all is owed to Mobile Application Development.[6]

The available wireless handheld PCs or devices in use in the mineral industry run various operating system such as Palm OS, Symbian EPOC
and Microsoft Pocket PC (windows CE). These devices could wirelessly be connected to a network via satellite or the internet making use of TCP/IP (Transmission Control Protocol/Internet Protocol) protocol. [7] This research has revealed that, “among the Personal Digital Assistant available, HP iPAQ are the most popularly and widely used in mineral exploration” in Africa. An interview was conducted with four colleagues who have worked as Exploration or Senior Geologists in various mining and exploration companies in Africa, namely in Ghana, Mali, Burkina Faso, DR Congo, Guinea and Liberia. BlackBerry is trying to break into this market but is facing many challenges since the iPAQs have gained strong roots.

4. Environmental effects by Mobile Devices:

The use of mobile devices could have an adverse effect on the environment if it is not disposed off properly according to the guidelines set by the appropriate regulatory body. If these devices are not properly handled according to the health and safety regulatory measures, they could pose a great danger to people and the environment at large. Normally the IT Department puts the condemned or dilapidated devices together and either ships them to the manufacturers or has them collected by a dedicated body. [18]

5. Mobile Application Development Platform (Java ME & .NET CF):

The mobile application development platforms widely used today are Java Micro Edition and .NET Compact Framework (*CF*) programming with C#. Choosing a platform is based on factors such as flexibility, portability, stability etc of the platform and how comfortable the developer is with the platform. Applications developed with Java ME could run on any Mobile Information Development Profile (*MIDP*) supported device. [2] Mobile Applications developed using .NET CF with C# are limited to mobile devices running Windows Mobile O/S. The Pocket PC is the most successful Windows CE-powered device and this has made .NET CF supported by all versions of Pocket PC e.g. Pocket PC 200, 2002, 2003

ISBN: 978-988-17012-0-6
ISSN: 2078-0958 (Print); ISSN: 2078-0966 (Online)

etc. The Pocket PC comes pre-installed with Windows CE. [8]

6. Mobile Applications Architecture:

Various architectures were considered for the application described in this paper. The options are outlines below.

Standalone: This type of application need not connect via a network and may be suitable for an application such as a game, dictionary etc. An advantage of this is that, there is no problem of network connectivity, therefore less spending as far as airtime is concern. On the other side, data synchronization and sharing are not possible and communication with other people is impossible. [9] Architecture is not suitable for the application described in this paper.

Client/Server (fat client): This is a type of architecture in which application code and some sort of logic are installed on both the client and Server ends. Some processing takes place at the client end and periodically makes network connection to the Sever to either upload or download data. This architecture is alternatively referred to as *'smart client'*. This architecture has a direct bearing on the implementation under this document, since any MIDlet is a smart client and some processing takes place on the client side as well. Even though network connection is intermittent, carrying out processing on the field might delay work progress greatly since mineral exploration data is enormous. [9]

Client/Server (thin client): With this type of architecture, there are no codes installed on the client device at all but all the application logic and processing rest on the server end. The client device only run a general web browser e.g. MS Pocket Internet Explorer on Pocket PC. Thin client is also referred to as *'dumb client'*. [14] Until a network connection is made to the main server for data upload, data could be entered and stored in the Record Store provided by the MIDP implementation. [9]

Mobile applications developed using Java 2ME platform rely on Java Application Package Interface (*APIs*) for example to build a Graphical User Interface (*GUI*). These APIs are divided into Configurations, Profiles and Optional APIs. Each of the divisions relates to specific information about APIs and group of devices. The API responsible for GUI design or creation is the *javax.microedition.lcdu* and the one responsible for storing records into the record store is *javax.microedition.rms*. **[2][3]** The application under this implementation would make use of some of these APIs in a bid to building a User Interface (*UI*), storing data for later upload, and making file and network connections i.e. *javax.microedition.io* and *java.io*. **[2][3]**

As the adoption of wireless mobile communication and mobile devices by companies and businesses including the mining industries becomes rampant, the security risk and associated costs continue on the increase. Many employees or users, including geologists may not understand that a mobile device may contain critical information and this could lead to difficulties in controlling risk. Mobile device users may manipulate the devices anyhow, which may affect certain critical information. **[10]** Users of mobile devices need some sort of network connectivity as mentioned above to interact with other users/geologists or to upload and download data. However, there are certain restraints that could affect transmission. Signal strength can greatly affect wireless data transmission, which could consequently lead to data loss, and therefore geologic exploration data, which is wirelessly transmitted to the data centre, are always at risk of losing certain part of it.

Other constraints associated with the use of mobile devices are network issues; is the network connectivity available at all times, how long can you stay on the network? Network reliability i.e. is the network always there and ready to use, can you get connection at any time when needed? Delay variation, also known as jitter; how long does it take to get connection to the network, how long does it take to make an upload or download? Other impairments could be shadowing i.e. blocking and reflecting at large obstacles, refraction (*bending*) depending on density of the medium that would be encountered, scattering and diffraction at edges. **[10]** Satellite are now widely used for wireless WAN (*wide area network*), and this is because it could be used in even areas of low population density and that is exactly the areas where most geological exploration is carried out. Lately satellites are used to connect remote or developing areas due to their geographical location all over the world and many parts of the world also lack internet connectivity. **[11]**

Global mobile communication is the latest trend for satellites. Due to the high latency, satellites using lower orbits are needed. Mining and Mineral industries are relying on satellites for their mobile applications since exploration is mostly done in remote areas where telecommunication and internet connection are nil. Practically, the basic purpose of satellites for mobile communication is to extend the coverage area. Therefore, in the mineral exploration field where internet connectivity may not be available, satellite communication is very useful and highly recommended. **[11]**

This application was developed using Java ME. Many philosophies and designs behind the Java technology perfectly suite mobile applications. A soil geochemistry data capture application developed with Java would benefit from the following:

- The Cross platform feature of Java: Java is *WORA* (write once run anywhere), perhaps the most important in the diversified mobile device market. In a heterogeneous environment, Java is capable of developing and maintaining a single client for all devices, and this could result in a huge savings. **[1]**
- Security implementation of Java: The Java runtime has inbuilt advanced security features and it is provided through a domain-based security manager and standard security APIs. **[1]**
- The Robustness of Java: The Java byte code is verified before execution. Garbage collectors reclaim memory lacks and Sensitive applications or data on devices are not affected even if a Java application crashes, because it is contained within the Java Virtual Machine (*JVM*). **[1]**
- Java is Object-Oriented: Java is object–oriented and well- designed language with a host of library supports, existing developers and easier to test. **[12][13]**
- Java has a built-in feature to support national character set and this could be a strong hold for the implementation to become universally viable. **[12]**

In addition, Java is easy to program, and has easy networking capabilities by opening a socket on read/write

ISBN: 978-988-17012-0-6
ISSN: 2078-0958 (Print); ISSN: 2078-0966 (Online)

across a RCP/IP network. As if that is not enough, Java has the ability to create software that is relatively easy to customize for other languages and locations i.e. the support for different data formats and Unicode for different alphabets and character sets. [9]

When writing the codes for this application, assistance of two java biased text editors were employed, namely *EditPlus 3* [27] and *JCreator LE 4.0*. *EditPlus* is a 32-bit text editor. *JCreator* is a Freeware and Shareware editor, a lightweight development environment and a powerful tool for java technologies because of its reliability and efficiency compared to other IDE's. [12][14][15][16][21]

Due to the diversity of portable devices available, Developing an application could be a daunting task which involves a great deal of time investment and other indirect commitments, but when the implementation becomes richly successful, developers turn to forget about any challenges that confronted them before, and during the implementation. This application is targeting Connected, Limited Device Configuration devices (*CLDC*). This application will go through diverse stages and strategies. This application employed Java 2 Micro Edition (*J2ME*) as the developing platform to build the Graphical User Interface (*GUI*) using MIDP 2.0 implementation Screen classes. The following MIDP Screen classes were used in the application; Forms, Commands for user input, TextBox and Alert Screens. In building the device-independent User Interface, the abstraction method was adapted and therefore, all the codes written in general term allowing the MIDP implementation on the device to decide how the UI is to be rendered to the user at runtime. Let us appreciate this as an example, a code was put as follows; *pAboutAlert.addCommand(newCommand("Reject",Command.EXIT,0))*, it is up to the MIDP implementation on the device in question to determine how the "*Reject*" command is rendered to the user and this is virtually done at runtime.

In the development process, the appropriate and needed Standard Configuration and Profile APIs relevant to the implementation were first loaded. For example since the application was intended to use *Forms*, *TextBox* and *Alerts* as part of the GUI from the MIDP Screen class, it is essential that *javax.microedition.lcdui* Profile API imported and loaded before the MIDP implementation could display them on the portable device. The following core Configuration and Profile APIs were also used with respect to this application, these are java.io, java.util, java.lang, and they are the core Connected Limited Device Configuration (*CLDC)* APIs. Another is *javax.microedition.io*, a network API, which handles file connections, was adopted. The *javax.microedition.rms* API is also responsible for storing records in the Record Store. Other APIs may be *javax.microedition.midlet, javax.microedition.io.file, javax.microedition.rms.RecordStore* etc. and these are Profile APIs. All these APIs are essential to the application and were imported at the beginning of the coding. The class Geochem was then created and asked to extend MIDlet and implements CommandListener and Runnable. That is, the created class '*public Geochem'* should listen to commands placed on the GUI and being able to carry out file connection as well. A command is anything the user can invoke or GUI buttons e.g. *OK*, *EXIT* etc.

Going forward, the variables that the applications would use were declared; displayables in the *javax.microedition.lcdui* (*Alert, Form,* and *TextBox, Form)* item i.e. *ChoiceGroup* and the item that would manage the portable devices' screen i.e. *Display*. Displayable is the parent of all screen players which supports flexible interface concept; *command*. They are unicast event source that fires off events once a command is invoked. The variables declared were TextFields and String variables. Examples may include *txtSample_ID, txtGeologist, strPrimary_Grid* and *strSample_IDLookUp*. Just to mention but a few. The relevant *Forms, Alerts* and *TextBox* were created with their respective appropriate commands on them. These commands were then injected in the Command Action otherwise nothing happens when they are invoked on the screen. *CommandAction* response to user input. A Record Store method implemented by the MIDP device was called and opened, in order to get the data that have been entered for later upload stored. For the application to be able to list records entered to the user, the *listRecord* method was employed.

Nonetheless, to lookup or search for particular details in the Record Store using its unique *Sample_ID* to delete or edit, the *LookUpForSample_ID, LookUpForSample_IDToDelete and deleteSelectedRecord* methods were used. In addition, the *deleteAllRecords* method was used to get the entire data in the Record Store deleted. Finally, in a bid to implement the optional xml feature, by writing data as xml or saving data to file, a separate thread was started using the *startSaveXMLToFile* and *saveXMLToFile* to output data as xml.

Running or executing codes ensures that the application performs as expected. Java codes are first built or compiled into java bytecode. This compilation is achieved during the build process. This application resorted to NetBeans [24] and/or Sun Wireless Toolkit [25] to take the Build and Run process. These packages come with incredible emulators with GUI just like a real device and make you envisage how the application would look like as exactly when deployed onto a real device. NetBeans [24] and Sun Wireless Toolkit formerly known as J2ME Wireless Toolkit [25] were especially built to take care of Java needs.

7. Results

Just like any other new application, this prototype application was tested using generic mobile phone emulator that comes with Sun NetBeans IDE 6.7. Testing newly developed application is an empirical technical investigation conducted to ascertain the quality (*correctness, completeness, security*) of the application with respect to the intent behind the application and its operation. Errors may be caused by mistakes on the part of the developer. Application or software fault may also arise from hardware changes (platform change). [23] Generally, changes in platform to run applications may result to code alteration, but remember Java is cross platform (*WORA*), and therefore may be exception. Testing an application brings up *comparison* or *criticism* comparing the behavior and state of the application with respect to a specific standard and *Soil Geochem* was subjected to this act. Testing gives the opportunity to evaluate the application, in the context in which it was intended to operate and against the technical idea behind the coding. The application was subjected to both *static* and *dynamic* testing. Static testing took into consideration a thorough review and inspection of the codes in particular and the

ISBN: 978-988-17012-0-6
ISSN: 2078-0958 (Print); ISSN: 2078-0966 (Online)

application as a whole while dynamic testing sort to running the application with an emulator using a typical soil geochemistry data.[23]

When the application is launched, a commitment screen comes up where the user is expected to either accept to continue using the application or reject to end it there. Beyond this screen, you can enter geochemical field data, save it, delete them as and when preferred, view help pages or user guide, make http connection etc. The application was evaluated to see whether what it is doing exactly what it was intended to do whiles writing the codes or otherwise. A typical Soil Geochemical data acquired from a colleague Senior Geologist who works with Liberty Gold and Diamond Mining Inc, Liberia was used. The application was verified and validated to see if it was built rightly, using the technique of review, inspection and walkthrough. [23]

Fig. 1 Commitment screen

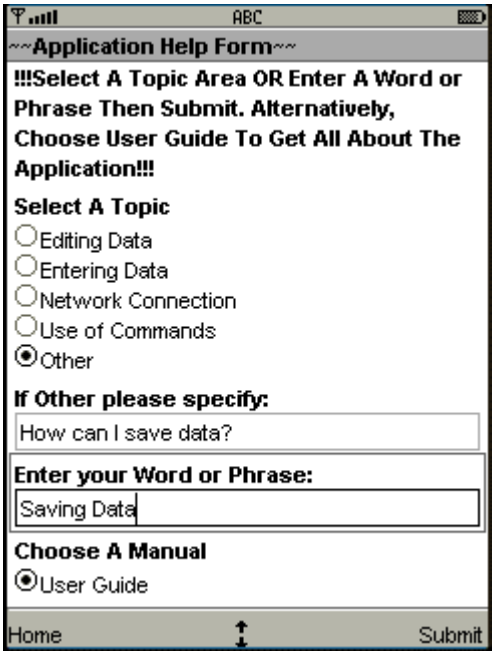


Fig. 2 Application Help Page

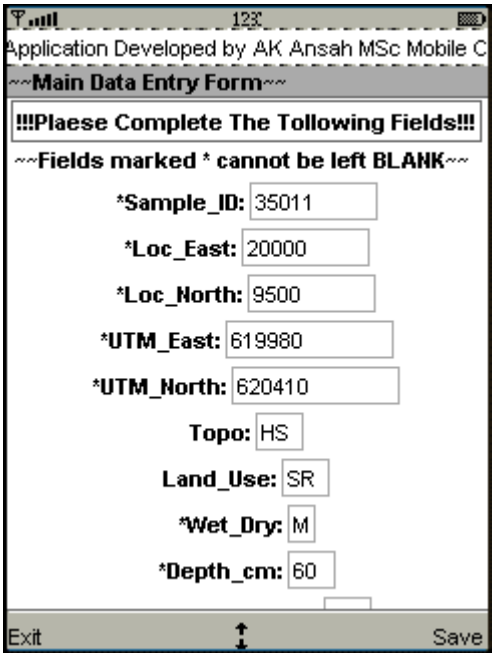


Fig. 3 Data entry Form

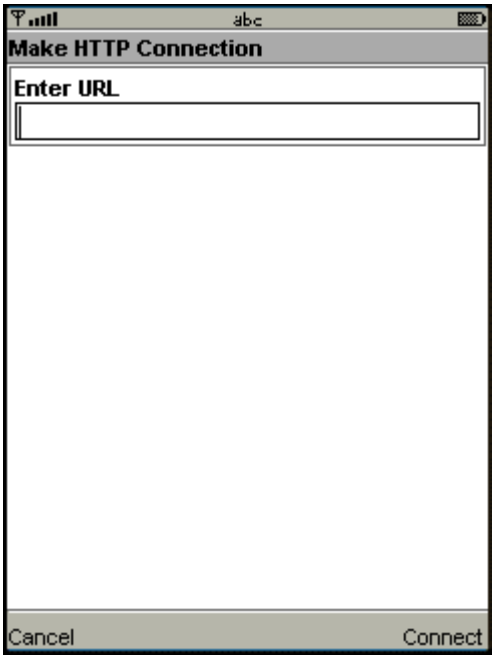


Fig. 4 Network connection screen

Fig. 1, is the commitment screen, which was explained above, a user, definitely needs to make a decision to either accept to progress with the application. To terminate the application the user must choose reject.

Fig. 2, lets the user select what to do; if a user wishes to know or learn about the application, then he selects *help* from the menu. A user may select *New* or *Edit* to start entering fresh data or changing existing data by deleting or playing around it.

Fig. 3, below is depicting the application's help page with a user querying how to save data. When that feature is fully implemented in the future, the submit button would bring out all the necessary steps to saving data successfully.

Fig. 4, is also showing a typical geochemical data being entered onto the data entry form. These are actually a representation of how the implementation would look like on a real device.

ISBN: 978-988-17012-0-6
ISSN: 2078-0958 (Print); ISSN: 2078-0966 (Online)

8. Discussions

Designing and implementing an application to meet its technical challenges could be a daunting task which is why programmers often than not place more emphasis on the structure and flow of the application. This area of the document depicts the entire implementation, digesting into the design, which spells out the structure and the flow of the implementation, by virtue of how the codes have been arranged to bring out the main idea behind the application. A user interface was created with the appropriate codes to display the features of the application including the commands on the screen.

Use case models and sequence diagrams are being used to provide model interaction between a user and the application and the flow of logic of the implementation. This enables the developer to document and validate the logic. [19][20] Furthermore, this section also talks about the strength of the application that is being developed after its analysis and testing with a typical geochemical data. The data to be entered on the application has been categorized into two; namely, *Common data* and *Main data* and therefore two separate forms have been created for them. The division was carefully done to avoid monotonous in the data that would be entered and to save some time as well. For example, *data_set* or *batch_no*, which may affect every sample collected, need not be entered every time for each sample and therefore all of its likes have been categorized as common data, which is entered once at the beginning of the fieldwork. All the other constantly changing data e.g. *sample_id*, *soil texture*, *topology* (*topo*) etc are categorized as Main data.

One great and unique thing about the application is that, it has two main branches. The *Edit* and the *New* tabs on the *Choice Alert* subclass of the screen class is used to welcome users of the application and both could lead the user to the Main Data Entry Form via *Sample_IDLookUp* TextBox and Common Data Entry Form respectively to begin fresh entries of data. Again, both the *Sample_IDLookUp* TextBox and the Common Data Entry Form are capable of performing *delete*, *lookup* (*search*), *http Connection* and *list* functions. Effectively, the flexibility and the friendliness provided by the design make it possible to perform either task without restarting or re-launching the application. For instance, if a user intends to make changes to an existing data and mistakenly chooses *'NEW'* on the *Choice Alert* welcome screen or even during the course of data entry process, he could still achieve his intension by choosing *'LookUp Using Sample_ID'* or *'Delete Selected Data'* tab on the Common Data Entry Form menu to achieve his wish. Nonetheless, if a user ends up selecting *'EDIT'* instead of *'NEW'*, his day is not finished, though. He could just hit the cancel tab on the *Sample_ID LookUp* TextBox and gain access to the Main Data Entry Form without restarting or re-launching the application. Remember time and tide wait for no man and in mineral exploration; time is more than a valuable entity.

In addition, the *'Next'* tab on the menu of the *Sample_ID LookUp* TextBox to redirect a user to the Main Data Entry Form to begin fresh entries without re-launching the application. Time is pretty precious and costly in the area of geology and mining field exploration. In other words, the application has been developed to meet even the least exploration trainee who might have got little or no ideas as to the use of handheld devices for mineral exploration. One screen could deal with almost all the field exploration tasks i.e. *capture, edit, list* and *delete* field data without too much navigation on the application, by selecting the appropriate tabs on the screen menu. There is also an inclusion of a *Help/Assistance* form or a user guide, which could assist users to familiarize themselves with the application by selecting a topic area or entering appropriate word or phrase. This is not fully implemented in this application, though. The *http connection* is also considered to some extent. The application also incorporates an optional feature, which is an exclusive discretion of the users to make use of it. This feature is the application's ability to produce or write the data in *XML* format, which is been handled by a separate thread to spare the GUI from blockage and freezing.

During the design of the application, use was made of a device-independent user interface. The abstraction method supported by MIDP implementation was employed to create the graphical user interface (*GUI*). MIDP makes it possible to combine both abstraction and discovery approaches to develop one application, but this application was developed solely with the MIDP abstraction approach only. The reason is that, the abstraction approach is easier to port to many devices and is more efficient. The main aim of the application is to be able to accept geochemistry data and stores it in the record store, to list data entered when needed, to edit or make changes to existing data. It is also to explore an existing data using the unique *sample_ID*, show users data they enter before saving, to delete any entry they wish and to make network connection.

The application was structured to encompass the following:

- A welcome Alert, which admonishes the user to make a commitment before proceeding to running the application called ***About Alert.***
- A second Alert which proceeds the ***About Alert*** after accepting the commitment which gives a user the opportunity to read help pages, *edit* an existing data or begin a fresh data entries known as ***Choice Alert.***
- Forms to accept personal data and common data and main data called ***Common Data Entry*** and ***Main Data Entry*** Forms respectively.
- A ***TextBox*** for *Sample_ID* lookup to make changes or to delete data
- An ***Exit Alert*** that lets the user confirms his intension to exit. It caters for inadvertent exiting as well.
- A ***LookUp Alert*** that brings up the lookup result
- ***Data Delete Alert***, which apparently asks the user to confirm his intention to delete any data.
- A data delete confirmed screen showing a confirmation of the deleted data. Data Delete Done Alert identifies the screen**.**
- ****Connect Form*** allows connection to be established between the device and the Local Area Network/Wide Area Network via http connection.

*****An ***Application Help Form*** responsible to giving a user an opportunity to get help as to how the application can be used effectively including the use of commands.

*****Not fully implemented

ISBN: 978-988-17012-0-6
ISSN: 2078-0958 (Print); ISSN: 2078-0966 (Online)

As part of object oriented application design, Use Case Diagrams have been used to illustrate user-application interaction. Use Case Diagrams are behavioral diagrams, which are defined by a Unified Modeling Language (*UML)*. It is a visual or graphical representation of the overview or a distinct business of the functionality that is being provided by a System regarding its primary elements (*actors*), their goals or process (*use cases*) and any dependencies between those use cases. Uses Cases and Actors are represented by *oval* and *stick figures* respectively. The elements/components of the system are the *use cases* and the *actors*. [27]

In figure 5 below, the actors are shown on the left and right side as the brown stick figure shape, the use cases are the light-blue oval shape. The connecting lines represent the associations between the actor and the use cases. The developer creates the use case model. Sequence diagram was used to describe the dynamic behaviour of the application i.e. sequence of actions that takes place in the application. Two elements are identified with the sequence diagram: *Object*, which is an instance of a class and *Message*, an interaction between different objects. Sequence diagrams have diverse advantages. [19][20][22] Fig. 6 below also shows typical Sequence Diagram illustrating different processes in horizontal line arrows and the messages exchanged between these processes or objects according to the order they occur. The arrows indicate messages sent from one object to another while the broken lines represent timeline stretching from top to bottom of the application. The solid horizontal arrows as seen on the system represent synchronous calls. It can be concluded that:

- The mobile application (prototype) for geochemical soil sampling was successfully developed.
- The MIDlet codes that were created and the Graphical User Interface (GUI) produced could be built using NetBeans IDE and run on an emulator.
- The dynamic testing of the application with the emulator using a true geochemical data proved that the application can accept data and store it in the Record Store

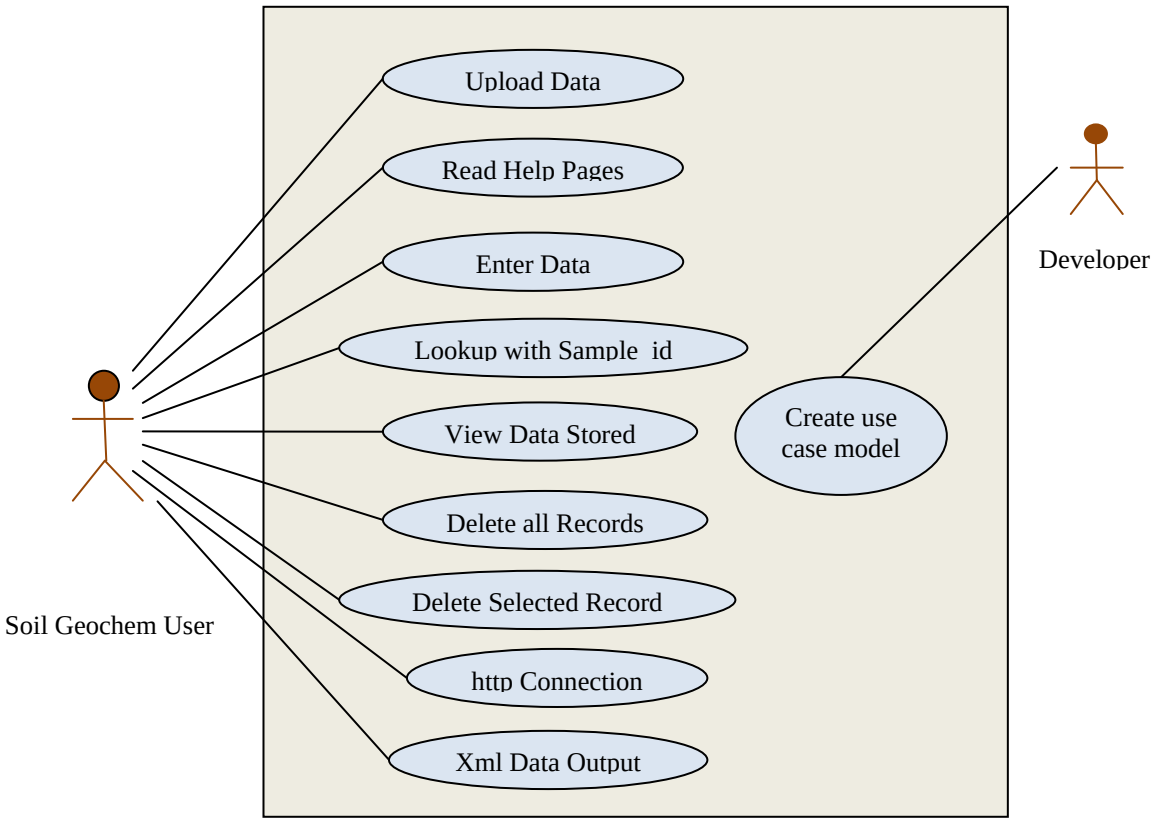
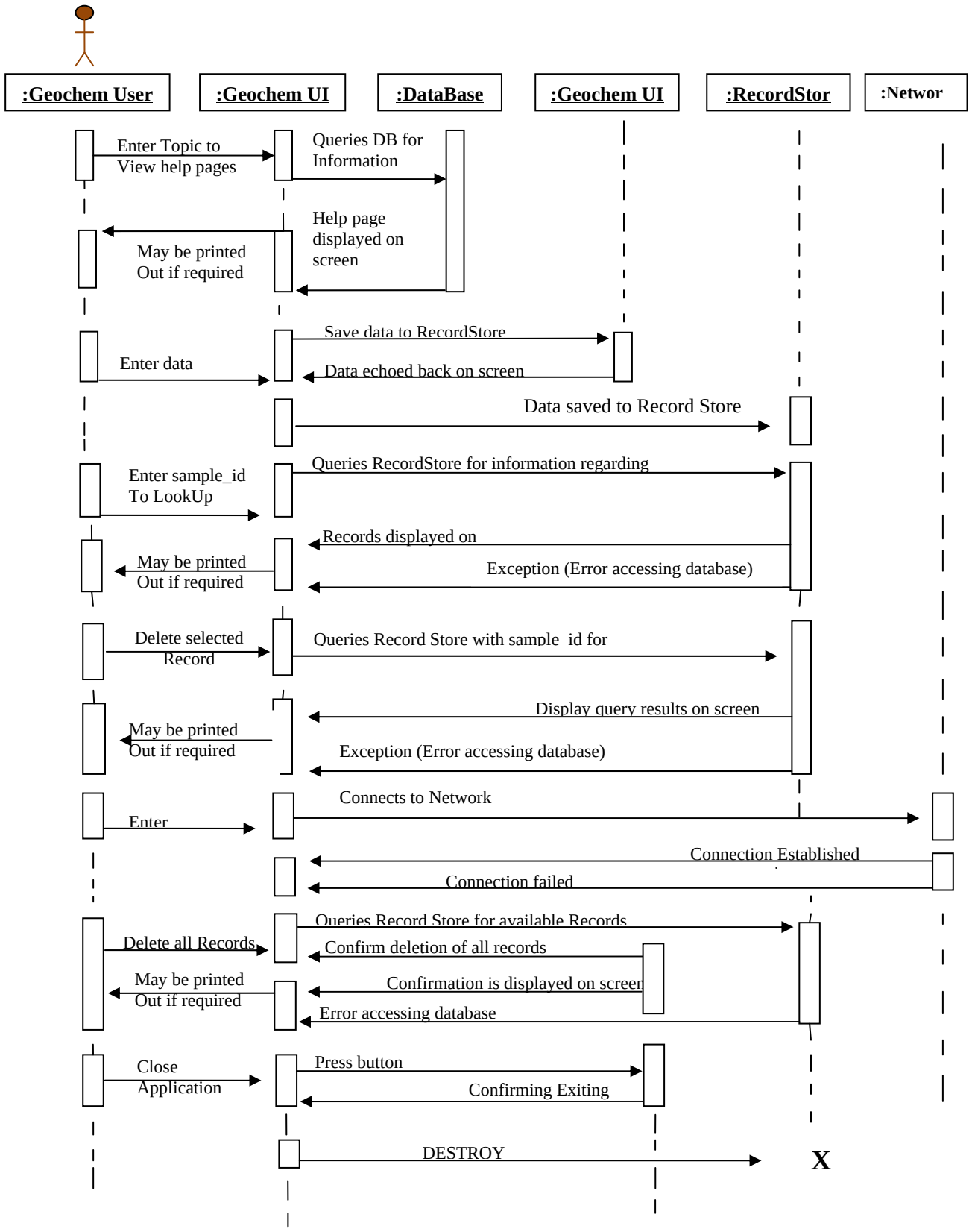


Fig. 5 Typical Use Case Diagram illustrating a user-application interaction

ISBN: 978-988-17012-0-6
ISSN: 2078-0958 (Print); ISSN: 2078-0966 (Online)



Fig, 6 A typical Sequence Diagram with objects and messages

ISBN: 978-988-17012-0-6
ISSN: 2078-0958 (Print); ISSN: 2078-0966 (Online)

ACKNOWLEDGMENT

A.K. Ansah thanks Prof. Adetunde, I. A, Dean of Faculty of Engineering, University of Mines and Technology, Tarawa, Ghana, for his advice, valuable comments and suggestions.

REFERENCES

[1] Yuan MJ. Enterprise J2ME, Developing Mobile Java Application. Prentice Hall PTR, New Jersey, 2006:9-13,16-20.

[2] Sing L, Knudsen J. Beginning J2ME from Novice to Professional. IIIrd ed. Apress, USA, 2005:1,5,17.

[3] Knudsen J. Wireless Java, Developing with J2ME. IInd ed. Apress, USA, 2003:1-2.

[4] Marjoaribanks R. Geological Methods in Mineral Exploration and Mining. Chapman and Hall, London, 1997:3

[5] Sides EJ. Mobile Device Software or Application. [Online] Available http://www.blackwellsynergy.com 2007:(accessed 17 December 2007 1719hrs).

[6] Brodaric B. The Design of GSC Fieldlog, Ontology-Based Software for Computer Aided Geological Field Mapping. [Online] Available http://www.sciencedirect.com/science 2003:(accessed 17 December 2007 1747hrs).

[7] Chen G, Kotz D. [Online] Available http://wiki.daimi.au.dk/pca/files/survey.pdf 2007: (accessed 09 September 2007 1820hrs).

[8] Yao P, Durant D. .NET Compact Framework Programming with C#. Addison-Wesley, USA, 2004:6-7,28.

[9] Windall G. Mobile Application Development Lecture Handout, Introduction to Developing Software for Mobile Devices. CMS Department University of Greenwich, Unpublished, 2006:3,11-14.

[10] McKenzie S. Mobile Technologies. Lecture Handout, Wireless Communication, CMS Dept. University of Greenwich, Unpublished, 2007:3,11,20.

[11] Schiller J. Mobile Communication. IInd ed. Addison-Wesley, USA, 2003:165-167

[12] Hortons I. Beginning Java2SD. 1.4 ed. Wiley, Canada, 2003:8.

[13] Hortons I. Beginning Java2SD. 1.4 ed. Wiley, Canada, 2003:19-20.

[14] Xinox. JCreator. [Online] Available http://www.jcreator.com 2008:(accessed 25 February 2008 1654hrs).

[15] Xinox. JCreator. [Online] Available http://www.download.com/jcreator-LE/3000-2 2124- 10044893.HTML 2008:(accessed 25 February 2008 1654hrs).

[16] Wikipedia. JCreator. [Online] Available http://en.wikipedia.org/wiki/JCreator 2008:(accessed 25 February 2008 1656).

[17] Java Community Process. Evolution of Java Technology, The Java Community Process. [Online] Available http://jcp.org/en/procedures/jcp2#1 2007:(accessed 22 September 2007 1456hrs).

[18] Forge S. Mobile Device Management [Online] Available http://inderscience.metapress.com/app/home 2007:(accessed 06 July 2007 1554hrs).

[19] Wikipedia. Sequence Diagram. [Online] Available http://en.wikipedia.org/wiki/Sequencediagram 2008:(accessed 25 February 2008 1607hrs).

[20] Chitnis M, Tiwari P, Ananthamurthy L. Sequence in UML. [Online] Available http://www.developer.com/design/article.php 2008: (accessed 25 February 2008 1611hrs).

[21] CNET. JCreator. [Online] Available http://www.download.com/jcreator-LE/3000-2 2124-10044893.HTML 2008:(accessed 25 February 2008 1700hrs).

[22] Effexis Software. Create Professional Sequence Diagram in less time, Sequence Diagram Editor. [Online] Available http://www.seuencediagrameditor.com/ 2008: (accessed 25 February 2008 1618).

[23] Wikipedia. Software Testing. [Online] Available http://en.wikipedia.org/wiki/softwaretesting 2008:(accessed 28 February 2008 1513hrs)

[24] NetBeans. NetBeans IDE 6.0.1 Download [Online] Available http://download.netbeans.org/netbeans/6.0/final 2008:(03 January 2008 0927hrs).

[25] Sun Developer Network. Sun Java Wireless Toolkit for CLDC. Available http://java.sun.com/downloads 2008:(accessed 03 Jan 2009 1002hrs).

[26] Wikipedia. PDAs. [Online] Available http://en.wikipedia/wiki/personaldigitalassist 2008: (accessed 25 February 2008 1704hrs)

[27] EditPlus. [Online] Available http://www.editplus.com 2008:(accessed 25 February 2008 1658hrs)