

REQUIREMENTS FOR SHORT LISTING**Part I.**

Instructions: Bidders are to respond with statements on experience and skills and knowledge. Refer to clients, projects, scope of work in IT development and number of man-months of input and timeframe of projects.

| No | Requirement | Type | Response of Organisation Expressing an Interest |
|-----------|--|-------------|--|
| 1 | Experience of development of IT applications in RDBMS server environment involving system design, data base design, program coding with industry standard methodology, testing, documentation and hand-over training. | Compulsory | |
| 2 | Experience with Microsoft development environment including, <ul style="list-style-type: none"> • Microsoft SQL Server & data base design; • .NET tools for code development; • Mixed Burmese & English language screens; | Compulsory | |
| 3 | Available equipment for development in the organisation which is similar as the proposed operating environment of a server with Windows Server and with Microsoft SQL Server 2008 / 2012 as the RDBMS. | Compulsory | |

| | | | |
|----------|--|------------|--|
| 4 | System administration experience with setting different access rights for different users. Access rights relating to access to data and transaction approvals. Backup procedures. Network monitoring of performance. | Compulsory | |
| 5 | Program development employing spatial structures within MS SQL; experience with a mixed GIS and RDBMS environment and data integration. | Preferred | |
| 6 | Experience in project management of IT projects for timely delivery of software products- experience with a project requiring more than 24 man-months of IT input for systems development. | Preferred | |
| 7 | In case that at least two firms will cooperate on the project, explain how the IT development work will be assigned and managed. | | |

Part II

Attach Curriculum Vitae of Indicative IT Persons

Attach the CVs of **at least two persons** who may be proposed for the contract if the bidder is short listed. Show the relationship of each person to the organisation.

INFORMATION FOR BIDDERS

OUTLINE OF IT SERVICES REQUIRED FOR LAMP

Preamble

The role of IT in LAMP is to improve efficiency of processes and reliability of land records through the application of IT. The system should be aimed to significantly improve the existing processes and records keeping and provide new skills and confidence to SLRD managers and staff.

The application software should be well prepared and documented and be able to be configured in a future roll-out of the new application software to other districts and townships of the country. In addition, it should form a base for future IT development on the applications.

The counterparts of SLRD will be primarily the staff of the two selected townships and the staff of the training centre.

Approach

The overriding requirement is an integrated architecture and software compatibility for all land applications developed under LAMP for SLRD with all data residing in a common database. In this way skills and training courses will be maximised and future development will add-on rather than replace the existing systems. The LAMP system design is built on these principles.

There are a number of major applications which will need to be designed and developed:

SLRD Land Systems Suitable for Computerisation

| Land Systems of SLRD and those which LAMP will Re-engineer and Introduce New Technology in Pilot Sites | |
|---|---|
| i. | DCDB ¹ & mapping system; |
| ii. | Capture of past surveys and mapping; |
| iii. | Adjudication & first land registration in farmlands; |
| iv. | New survey and re-survey system; |
| v. | New deeds registration and registration of these subsequent land transactions – Land Register to be computerised; |
| vi. | Capture of past urban land registration – limited pilot; |
| vii. | LIS - pilot; |
| viii. | Land zoning in rural lands for value and capability- pilot |

Not all of these systems will be fully developed under LAMP due to time and resource limitations. Those adding most value in the short term such as (i) to (iv) will be given priority. The Land Register of item (v) would be computerised but not the full transaction processing steps. Other items (vi) to (viii) such as LIS and land valuation will be small pilot applications.

Services

A software development firm (possibly in partnership with other firms) is required by the LAMP Project to be contracted by UN Habitat to develop software for use in the LAMP Project sites. The applications will be concerned with the processing and storage of land records in textual, spatial and image forms. The design of the system and database will be

¹ DCDB is Digital Cadastral Data Base

performed by an independent consultant engaged by UN Habitat. LAMP will provide the system design, database design and testing specifications for the contractor to follow.

The systems will be made operational at two township SLRD offices and at the training centre of SLRD. LAMP will provide the SLRD with servers, networks, server software and workstations and peripherals. A detailed project work plan will be provided to the contractor.

The contractor must include skills and experience in program development suitable to the tasks. In the case of a consortium of organisations there must be only one local firm as the lead firm which will be designated as the contractor.

For software development the contractor shall use his own equipment which must be compatible to that of LAMP. The contractor will use the LAMP equipment installed at SLRD for software testing, training and to install the software for use in operations.

Sustainability.

To ensure that the system is sustainable and replicable the following strategies are to apply:

- All application software is to be owned by LAMP for the unencumbered use by SLRD and UN Habitat;
- All application software development on the RDBMS is to be performed by a single IT Contractor comprising one or more IT firm/s;
- Reports and Screens shall use a mix of Burmese and English languages, according to the design;
- Specialised IT system administration training is to be provided to local staff;
- Training of users on the applications so that IT operations knowledge is not dependent on just one person;
- All IT equipment is to be standard commonly available equipment.

The overall software development requirements are given at Attachment 2. The use of any tools which are not consistent with the requirements in this attachment must be prior approved in writing by the STA² of UN Habitat.

From time to time the team leader may issue standards to be adopted.

The system will have a module for assigning various levels of access and record creation and change authorities based on passwords for authorised officers. The system will have backup and roll-back facility.

Phases of IT Implementation

There are 4 phases;

1. Farmland Title capture & processing, and establish the RDBMS;
2. Spatial data- DCDB, surveying & mapping data in rural & urban areas;
3. Processing land registration transactions in rural & urban areas;
4. LIS and Valuation / Zoning Pilot systems.

² Senior technical Adviser.

SCOPE OF SERVICES

The purpose of the new IT systems is to streamline processes and provide better integrity of land records both spatial and textual records at project township sites.

There are 12 IT development functions split into 4 phases of work. These are shown at attachment 1.

The activity descriptions are given in the LAMP Scope Document; for example, a full automated transaction processing system will not be included in the scope of the land document registration system; the update of the land registers however, will be included as well as keeping and reporting on the types of transactions for management purposes.

The software system should be well integrated with the operations.

The SLRD project staff will include the staff of the two selected townships and the staff of the training centre.

There are 3 major types of development services: (i) programing and testing; (ii) documentation and (iii) training, which apply to the development of all systems:

Table of Scope of IT Services

| |
|--|
| <p>The scope would provide best practice in software development suitable for the situation of SLRD with a township based land service delivery operation. The scope of works includes:</p> <ul style="list-style-type: none"> • Software and database development for the major land processes and data / record sets of SLRD to operate in a network server environment and to support (see LAMP project scope document for descriptions of the functions and system): <ul style="list-style-type: none"> ▪ Survey and mapping and DCDB; ▪ First time land registration; ▪ Subsequent land transaction registration; ▪ Pilot applications. • The processes to be developed include capture of the existing land records, validation of the records and future transactions to change/add to the register and maps; • Establishing a suitable RDBMS application development environment; • A suitable GIS system will be purchased by LAMP and provide the support for the spatial data management. A long term GIS expert is to be part of the LAMP team. It is intended that the spatial data is integrated with the textual and scanned/image data in the one database for reasons of cross data linkage and integrity and resilience to withstand system hardware failures or other sources of errors. • The conditions / constraints include: <ul style="list-style-type: none"> ▪ One homogeneous system to handle both rural and urban lands in non-forest areas; ▪ Design of the data base will allow for future expansion for new applications; • Oversight and coordination with the IT related consultants; • Software bug fixes on application software which occurs during the term of this contract. • Documentation and training. |
|--|

Deliverables of the IT Services

The **deliverables** would include software, test bench and test results, documentation and training on the IT development functions in 4 phases (see the work plan for the timing of the 4 phases):

Phase 1 (Farmland Title capture & processing, RDBMS setup)

- the *software developers environment*;
- the *LAMP data base*;
- the system for *processing new titles*;
- the system for *capturing existing Farmland titles & related documents* (non-surveying);

Phase 2 (Spatial data- DCDB, surveying & mapping data in rural & urban areas)

- the creation & maintenance of the DCDB being the *spatial component of the RDBMS* and allowing access / editing of spatial data by GIS operators;
- the *data base entities / images to be retained for all land surveys and maps*;
- the system for *capturing existing land surveys and new surveys & their related documents*;
- the system for *capturing sub-division surveys and updating the DCDB*;

Phase 3 (processing land registration transactions in rural & urban areas)

- *updating the land register based on approved land transactions (rural & urban)*;
- the system for *capturing existing (past) urban titles & transaction related documents* (non-surveying);
- the reporting system for *recording the number and type of transactions submitted for registration and actually registered (rural & urban)*. Note, this is not a transaction tracking system nor is it an automated deed registration system.

Phase 4 (LIS and Valuation / Zoning Pilot systems)

- the pilot system for *capturing LIS data and processing LIS functions*;
- the pilot system for *capturing valuation / zoning data and processing farmland valuation functions*.

PLANNED INPUTS (see Attachment 1 for details)

The input plan is summarised below:

| Phase | Software Development Input (Person Months) |
|--------------|--|
| I | 8.7 |
| II | 11.5 |
| III | 7.9 |
| IV | 5.5 |
| Total | 33.6 |

Other Terms of Reference

| | |
|-----------------------|---|
| Work Plan | The Contractor is expected to develop and keep up to date a work plan. |
| M&E | <p>The M&E activity is crucial to the success of LAMP. The project aims to design and develop better processes and better application of new technology; to demonstrate the impacts in pilot townships; and, to build capacity of SLRD through the CLRDTC and the staff on the two pilot townships. The M&E system must inform the project during operations on how / where to make improvements and at the end of the project on major lessons and recommendations.</p> <p>The M&E system places emphasis on evaluation, not only monitoring. The project log frame provides a suitable project structure and evaluation and monitoring framework for the reporting.</p> |
| Steering Committee | The peak management body for LAMP is the LAMP Project Steering Committee. Consultants may be required to make special reports to the committee from time to time or to assist counterparts to make such reports. |
| Intellectual Property | Unless specified in writing, all intellectual property arising from the LAMP Project remains with UN Habitat. |
| Good Will and Trust | It is expected that all Contractor's staff shall establish and keep good relationships with counterparts and staff of the Government and also with fellow LAMP staff. As a change management project, building trust and good will is very important for developing and sustaining appropriate changes. |
| Project Partners | The main project partners to LAMP are SLRD, UN Habitat and LIFT. |
| Public Relations | LAMP has a project brochure. These and other handout materials should be used to inform the professions, government, donor community and the public on the good work of LAMP. |
| Reference Documents | <p>The key reference documents to guide Contractors on their work are:</p> <ul style="list-style-type: none"> • LAMP Project Scope Document; • Project Work Plan; • Contractor bid proposal and contract. <p>As the project proceeds decisions of the steering committee may amend the above project documents.</p> |
| Political Awareness | Contractors should not involve LAMP in any political controversies. LAMP is largely a technical project. Lessons from LAMP may however be fed into land policy discussions through various channels such as the SLRD steering committee and the national scrutinising committee. Non-citizens of Myanmar working on the LAMP Project must not get involved in any way with local politics, demonstrations or similar. |

Attachments:

- Attachment 1: IT SOFTWARE DEVELOPMENT WORKS BY OUTPUT
- Attachment 2: SOFTWARE DEVELOPMENT REQUIREMENTS

ATTACHMENT 1

IT SOFTWARE DEVELOPMENT WORK BY OUTPUT

| WBS Output | No | Function | Kind of IT Work and Indicative Inputs (person-months) | | | | | Remarks |
|------------|--------------------|---|---|-----------------------------------|----------|----------------|----------------------|---|
| | | | IT Development Phase Number | Programing and Testing with RDBMS | Document | Training (OJT) | Total Software Input | |
| 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 |
| 1 | 1 | Capture Past surveys | II | 2.0 | 0.5 | 0.5 | 3.0 | For output 1 the software development specifications will be done by the IT Consultant. Data capture and link scanned survey documents to RDBMS parcel data |
| | 2 | New survey and re-survey system; | II | 2.5 | 0.5 | 0.5 | 3.5 | Includes records of change and issuing of new parcel numbers managed by the RDBMS. |
| | 3 | DCDB Development | II | 2.5 | 0.5 | 0.5 | 3.5 | DCDB is controlled under the RDBMS but the spatial constructs are interpreted by the GIS. The RDBMS records are linked to each parcel polygon and history of changes is maintained. |
| | Sub-Total Output 1 | | II | 7.0 | 1.5 | 1.5 | 10.0 | 10.0 |
| 1 | 4 | Establish the RDBMS application development environment | I | 1.5 | 0.2 | 0.0 | 1.7 | Data Base design, system admin arrangements, design & development standards, protocols for application development to be documented by the IT Consultant. The Software Developer will write code for the programmers' environment in accordance with specifications of the IT Consultant. |
| | 5 | Data Base Development | I | 2.0 | 0.0 | 0.5 | 2.5 | Coding of the interface with the RDBMS and including security features in accordance with specification of the IT Consultant. |
| | 6 | Adjudication | I | 0.0 | 0.0 | 0.0 | 0.0 | Largely a manual process involving field work and administration thru the FAB. The forms are filed in the case file for each land parcel. |
| | 7 | First land registration in farmlands & compatible with the register for urban lands- includes capture of all past rural | I | 3.0 | 0.5 | 1.0 | 4.5 | One standard register for all types of title (urban & rural) suitable for all townships. Reporting on changes to the register. Audit records. |

| | | | | | | | | |
|--------------------|----|---|-----|------|-----|-----|------|--|
| titles | | | | | | | | |
| Sub-Total Output 1 | | | I | 6.5 | 0.7 | 1.5 | 8.7 | 8.7 |
| Total Output 1 | | | | 13.5 | 2.2 | 3.0 | 18.7 | 18.7 |
| 2 | 8 | Registration of subsequent land transactions (rural & urban) and reporting system | III | 4.5 | 0.5 | 0.5 | 5.5 | For output 2 the software specifications will be done by the IT Consultant and based on the re-engineering design of the Land Records Consultant. Common database & development as at item 5. Register updating is to be based on transactions. Statistical reporting including performance. Efficiency improvements. QA and audit trail. In this phase no automated transaction processing modules. See task 7. The one Land Register is used for all titles. |
| | 9 | Capture of past urban land registration – limited pilot; | III | 2.0 | 0.2 | 0.2 | 2.4 | |
| | 10 | Sub-division surveys | II | 1.5 | 0.0 | 0.0 | 1.5 | In addition and linked to the development work at item 2. |
| Total Output 2 | | | | 8.0 | 0.7 | 0.7 | 9.4 | 9.4 |
| 4 | 11 | LIS - pilot; | IV | 1.5 | 0.5 | 0.5 | 2.5 | RDBMS development is limited as most effort is on GIS applications. Specifications from LIS Consultant. |
| | 12 | Land zoning in rural lands for value and capability | IV | 2.0 | 0.5 | 0.5 | 3.0 | Feasibility study by expert. If agreed development will proceed starting with specifications from Valuation Consultant. Software development builds on the database developed at the first three phases and will include capturing attribute data for various polygons so that zones can be defined. RDBMS will allow effective management of attribute data. |
| Total Output 4 | | | | 3.5 | 1.0 | 1.0 | 5.5 | 5.5 |
| TOTAL Outputs | | | | 25.0 | 3.9 | 4.7 | 33.6 | 33.6 |

ATTACHMENT 2

SOFTWARE DEVELOPMENT REQUIREMENTS

Purpose

This attachment provides the broad guideline for the software development. Detailed design specifications (software functional design, database design and screen designs) will be provided to the contractor before the development of each phase is to be started.

Introduction

The **Land Records Management System (LRMS)** is an information system designed for the management of **SLRD** land records and streamlining of SLRD services. It is developed by the SLRD with technical assistance of UN Habitat to support the activities of the LAMP Project. It aims to increase efficiency, accountability and accuracy in the land related transactions involving titles, registers, and maps within the jurisdictions of the department.

The overall objective of the system is to provide a unified repository of property information that includes textual, imagery, and spatial data. The integration of information involves the investment in technology, re-engineering, and streamlining of processes. Recognizing the business needs of the government and the change management effort required in implementing the system and other considerations, LAMP has adopted a phased approach in the implementation of the LRMS. The entire solution shall be implemented in three phases plus a fourth application phase for pilot systems.

1. Data capture of titles and parcel information (textual and image scans)
2. Setup of the Digital Cadastral Data Base from cadastral maps and new surveys and update surveys
3. Deed Registration and transaction tracking system

In its first instalment, the system deals mainly with the capture of existing information from the titles, registers, and maps which will provide the base information for future requirements.

The data capture requires a streamlined process to ensure efficiency and security of documents while implementing operational standards for quality checks and encoding validation using predefined business rules. The required process flows that will ensure this operational strategy are already detailed.

System Overview

The LRMS is an application system designed for the storage and management of land related records such as titles, registers, and maps. When fully operational, the system is expected to provide the following benefits;

- Enforced data integrity and facility to identify conflicts in land information
- Efficient land transactions of core agencies
- Efficient management reporting
- Improved service delivery standards
- Allow land information exchange between branches and other ministries
- Computerized land records for backup and long term storage

The application captures all the land related datasets in the form of text, imagery, and spatial objects and stores them in a relational database system to provide unified and integrated information to the users. This information can be further analysed through preformatted reports and the use of GIS functionalities.

A unique feature of the system is the capability to provide spatial functionalities by embedding GIS controls into the application. It provides the appropriate tools to upload, compute, and validate the polygons of the parcel associated with the titles, and store the geometry of the parcel object in a spatially enabled relational database management system (RDBMS) on the same record as the textual and image information. This eliminates the errors commonly associated with a loosely linked GIS and RDBMS. It facilitates integration through the use of manually configured links. Scanned documents are stored in the same database using Microsoft SQL Filestream objects [http://technet.microsoft.com/en-us/library/bb933993\(v=sql.105\).aspx](http://technet.microsoft.com/en-us/library/bb933993(v=sql.105).aspx). This provides additional security since the documents are not stored on the file system and can only be accessed and viewed through the use of the application. Also, the mechanism of storage and retrieval is transparent to the users.

The spatial information can be viewed through the built-in map viewer. This facilitates the fast and easy retrieval of standard maps using predefined queries through a secure database connection. The system also provides spatial analysis and visualization by incorporating thematic maps, labels, and other GIS tools in the map viewer. These functionalities are accessible through a set of user friendly toolbar buttons. Printing of maps with user-defined layouts is also available in the same context menu. Maps may include related textual information from linked tables or other spatial layers stored as filestream object on the same relational database.

Design Considerations

The intention is not only to create a system design that will guide the development of the application but also to provide a working environment where the development of the system can be continuously sustained. The selection of technologies to be employed in implementing specific features and functionalities is crucial to the success of the system development and can render the maintenance and enhancements of the system sustainable.

Platform

The major issue in the sustainability efforts is the platform on which the application will run. There are several factors that should be considered when deciding what platform or technologies to use for this project. These are;

- Usability and ease of development
- Evolving technology
- Scale of the Project
- Accessibility to source codes
- Cost
- IT skill and support in the local environment
- Scalability and ease of integration with other software

Having carefully considered these factors, this design recommends developing the application using the **Microsoft .NET** technology. During the meeting in the SLRD in Nay Pyi Taw last 17 June 2014, the project counterparts stated that there is a huge local support in the use of Microsoft products and the IT community in Yangon is largely dependent on the Microsoft .NET technology. This ensures sustainability of the development as the government can easily find affordable developers who can develop and maintain the work.

While there is a cost in acquiring the necessary software to support the development and run the application, there is potential expenses to be incurred in maintaining an open source system because based on local market demand, developers specializing on open source technology may charge more and the skills necessary to sustain the maintenance and development may be more expensive in the long run. Further, certain functionalities of the system are largely dependent on the evolution of technology such as spatial tools and database storage. Microsoft technology is continually being developed and new innovations are being added into their existing products requiring minimal upgrade fees.

Most developers enjoy the Microsoft technology due to the ease of use and the .NET Framework is widely believed to be very easy to develop with. Utilizing these technologies seems to offer a range of benefits. The first and foremost is that a Windows server is much easier to configure versus a Linux server. That is not to mention the fact that configuring an open source servers such as Linux has a much steeper learning curve and takes more of an investment of time. Another benefit of utilizing Microsoft technologies is the flexibility to work with varying languages on a framework. There is also a large IT community in the internet providing technical support for free where developers share and exchange ideas.

Many developers also find some advantages of the Microsoft technology to be the quickness of being able to produce the product since there is often less obscurity and complexity when it comes to the .NET code. There is also the ability to build both Windows and Web applications, which allow for the use of multiple opportunities for builds. In addition, there is belief that utilizing the .NET allows for applications that are highly data-oriented, or applications that support huge database functions. Most of the time, Microsoft is able to identify and characterize security risks and develop a patch to these risks a lot easier than using open source technologies. They can also easily distribute the patches automatically through Windows Update.

In general, sustainability of the system can be achieved by capacitating the local stakeholder to develop the system through the use of sustainable software components including widely used programming language, easy to use GIS, and an efficient, well established database system. To ensure this objective is realized, the project shall supplement the local knowledge with appropriate training, supervision, and a sound and effective system design that is based on extensive experience on land management and administration.

Windows Forms

This first phase of the system is initially developed for townships having minimal resources and staff with basic computer skills and is therefore constrained to run on limited methods of implementation. For that reason, the system will be developed as a desktop application with provisions for using the internet as a medium of connection through the use of Windows Communication Foundation (WCF). Windows desktop applications provide easy-to-use user interface and is generally more secured and responsive than web based applications, especially when using GIS tools.

Web Services

WCF is a service layer that allows building applications that can communicate using a variety of communication mechanisms such as Named Pipes, Peer to Peer, TCP/IP etc, all available on the internet. This provides the capability for the application to connect, view, and update the database stored on a remote and centralized location using the internet as the means of communication. In practical application, the database server may be maintained in a regional or district office of SLRD and the township connects to the server using the internet. This may be a future requirement after LAMP.

GIS Runtime

The spatial functionalities of the system are provided by an ActiveX control.

Runtime licenses are used to incorporate GIS capabilities within other applications or to deploy IMS applications to web servers. Runtime licenses provide full programmatic capabilities but do not have the learning overhead included in implementing spatial functionalities. Runtime licenses do not require expert GIS skills and are "license only" products delivered as a serial number for use with product downloads.

A proven example of a low cost GIS runtime is Manifold; the runtime licenses are x64 licenses, allowing use in either 32-bit or 64-bit Windows systems. <http://www.manifold.net>

Alternately, the spatial functionalities may be provided by using an open-source control such as Sharpmap. Sharpmap controls provide a free alternative to providing spatial functionalities to the system. It is an easy-to-use mapping library for use in web and desktop applications. It provides access to many types of GIS data, enables spatial querying of that data, and renders beautiful maps. The engine is written in C# and is based on the .Net 4.0 framework. <https://sharpmap.codeplex.com>. SharpMap is released under GNU Lesser General Public License.

** The choice between Manifold, Sharpmap, or other third party GIS components will be decided in the second phase when the DCDB is designed and developed together with the GIS Contractor. It is assumed that all GIS components have similar methods of implementation and migrating from one to another simply requires reconfiguration and a straightforward conversion of some codes. This design emphasizes on the need to use GIS technology without being dependent on a particular GIS tool. The selection will be based on responsiveness, maintainability in longer term, capability, technical support, sustainability, compatibility, and cost.*

Visual Basic and C#

Visual Basic and C# are the two widely used general-purpose, high-level programming languages for the .NET framework from Microsoft. They are both object oriented, event-driven and run on the vast array and collections of reusable objects from the .Net Framework. While the choice of one language over the other is highly argumentative, this project will base the selection on the skill and expertise of the IT firm who will be contracted based on market demand. It is important to note that both languages are intrinsically similar and when compiled, produces the same Intermediate Language (IL) codes for the .NET Common Language Runtime (CLR).

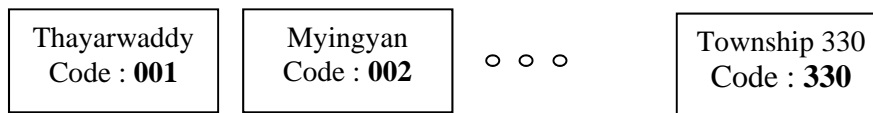
Because of its relative ease in readability, some code snippets herein provided that expound on specific algorithms are written in Visual Basic.

Architectural Strategy

The LRMS is developed for pilot townships having different situations of land types and certain peculiarities in the titling procedure. The system is developed to accommodate these various conditions into a single version and shall be installed, tested and proven in the pilot township sites. A single version makes software maintenance realistic.

Eventually, it is expected that LRMS would be rolled out by SLRD to all of their townships across the country. Provisions to combine the databases into a district, regional, or a nationwide database is included in this design. This shall be facilitated through the use of unique identifiers for land, title, and registry records. The unique identifiers shall bear the location of the township. A configuration table is designed for this purpose.

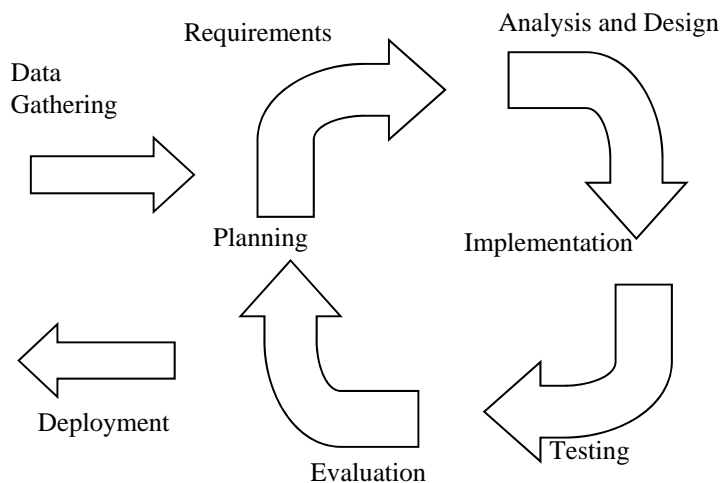
Coding for Township Sites



Software Development Methodology

The development of the system shall follow the Agile Scrum software methodology. The system follows an iterative and incremental software development process which is based on developing systems through repeated cycles (iteration) and in smaller portions at a time (incremental). This allows the team to take advantage of what is learned during the development of earlier parts or versions of the system. Learning comes from the development, testing and use of the system, where possible additional requirements may arise with a simple implementation of a subset of the software releases and iteratively enhance the evolving versions until the full system is implemented. For every iteration, the design enhancements are considered and new functional capabilities may be added.

AGILE METHOD



Modular Design

The entire design covers the system overview, database model, system architecture, coding guidelines, testing methodology, and the over-all organisation of the system.

To facilitate ease of development, distribution of tasks, and tracking of progress, the system is decomposed into software components called modules, each of which may undergo its own iterative phase of development. The following are the modules;

| Modules Phases | Data Capture | Deed Registration | Survey and Mapping | Transaction Tracking |
|--|--------------|-------------------|--------------------|----------------------|
| 1. Register and Titles | ✓ | | | |
| 2. Spatial Data and Maps | ✓ | | ✓ | |
| 3. Deed Registration and Updating of Registers | ✓ | ✓ | | ✓ |

Coding Standard

The following coding guidelines are issued as part of the sustainability standards for the development team and should be strictly followed in the refinement and further development of the system. Maintaining such standards ensure the readability, ease in debugging, and maintainability of codes.

- Follow the style of existing code. Strive to maintain consistency within the code base of an application. If further guidance is needed, look to these guidelines and the .NET framework for clarification and examples.
- Make code as simple and readable as possible. Assume that someone else will be reading your code.
- Prefer small cohesive classes and methods to large monolithic ones.
- Use a separate file for each class, struct, interface, enumeration, and delegate with the exception of those nested within another class.
- Write the comments first. When writing a new method, write the comments for each step the method will perform before coding a single statement. These comments will become the headings for each block of code that gets implemented. ***All comments must be written in English.***
- Use liberal, meaningful comments within each class, method, and block of code to document the purpose of the code.
- Mark incomplete code with // TODO: comments. When working with many classes at once, it can be very easy to lose a train of thought.
- Never hard code “magic” values into code (strings or numbers). Instead, define constants, static read-only variables, and enumerations or read the values from configuration or resource files.
- Prefer while and foreach over other available looping constructs when applicable. They are logically simpler and easier to code and debug.

- Implement a Static Class with constants, properties and public events to hold system wide variables.
- Use the StringBuilder class and it's Append(), AppendFormat(), and ToString() methods instead of the string concatenation operator (+) for much more efficient use of memory.
- Be sure Dispose() gets called on IDisposable objects that you create locally within a method. This is most commonly done in the finally clause of a try block. It's done automatically when a using statement is used.
- Never present debug information to yourself or the end user via the UI (e.g. MessageBox). Use tracing and logging facilities to output debug information.

Camel casing is the preferred manner of capitalizing identifiers. Other standard should follow Microsoft Design Guidelines. (<http://msdn.microsoft.com/en-us/library/xzf533w0%28vs.71%29.aspx>)

Test / Code / Test cycle

Test plans are to be prepared reflecting the functional requirements. The test plans are executed by end users to identify defects, bugs, and to check if the system meets the requirements as well as adhere to the agreed design. Where defects are found, remedial action shall be undertaken by modifying the design and/or re-coding. The failed test will be executed again and the cycle will be repeated until the test has passed.

The testing plan is prepared separately.

TODO: Test benches archived, for each phase

test 1. Internal, 2. User reps

Development Environment

The system is developed on the following environment:

| Component | Software | Version |
|-----------------------------------|--------------------------------|---------|
| Server Database Management System | Microsoft SQL Server 2012 | 10.5 |
| GUI Application | Microsoft Visual Basic .NET/C# | 4.0 |
| GIS Tool | GIS Component | |

The system is built on top of the .NET framework to provide a uniform structure in consuming;

- Business components/objects
- User interface
- Database management
- Intelligent Data binding
- Security module
- Localization

- Exception Handling and logging
- Customizable menu functions
- Connection pooling

A set of classes may be written as foundation classes to ensure a consistent pattern in the user interface. However, this strategy is optional. The developer may instead use intrinsic controls provided in the latest .Net framework.

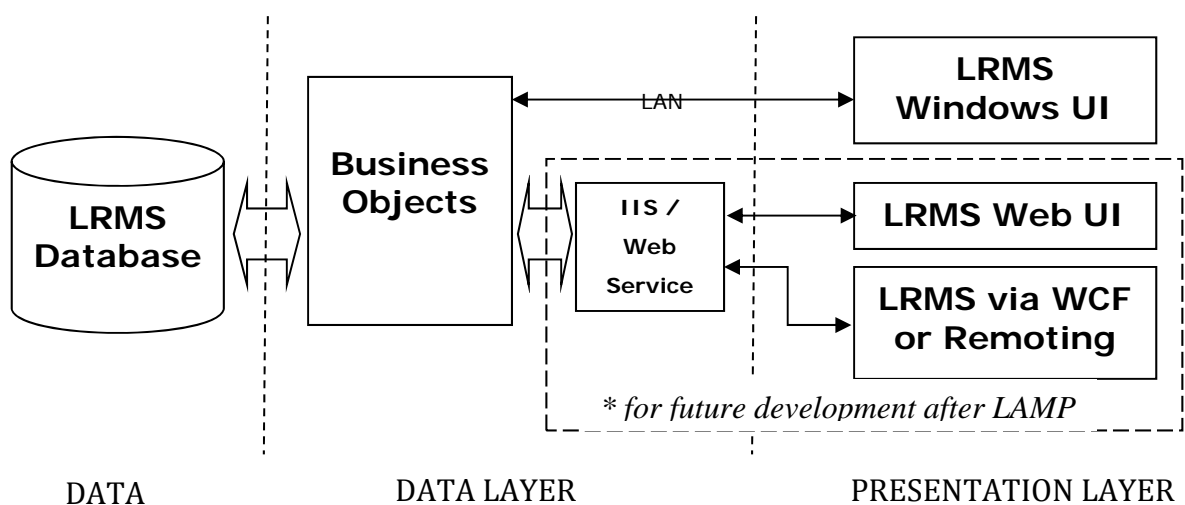
Operating Environment

The system is developed to run on ;

| Component | Software/Setting | Version |
|---------------------------|---------------------------|-------------------------------|
| Server OS | Microsoft Windows | Microsoft Windows Server 2012 |
| Workstation OS | Microsoft Windows | Windows 7, Windows 8 |
| Network Protocol | Ethernet TCP/IP, Internet | |
| Network Protocol Remote | VPN (Preferred) | <i>Future requirement</i> |
| Minimum Screen Resolution | 1280 x 768 resolution | |

System Architecture

The system is developed using a 3-tiered architecture. This represents the logical separation of the application's components into groups that represent distinct roles and functionality. Using a layered approach can improve the maintainability of the application and make it easier to scale out when necessary to improve performance.



Presentation Layer / User Interface

The presentation layer contains the components that implement and display the user interface and manages user interaction. This layer includes controls for user input and display, in addition to components that organize user interaction such as changes to data.

Data Layer / Business Object

Business Layer is concerned with the retrieval, processing, transformation, and management of application data; application of business rules and policies; and ensuring data consistency and validity. The UI components collect the required data from the user and pass it to the business layer wherein the data is used to perform the required business processes. Many business processes involve multiple steps that must be performed in the correct order, and may interact with each other through an orchestration. Business objects encapsulate the business logic and store data values and expose them through properties. They contain and manage business data used by the application and provide programmatic access to the business data and related functionality. Business entities also validate the data contained within the entity and encapsulate business logic to ensure consistency and to implement business rules and behaviour.

Data Storage Tier

The database will be stored on Microsoft SQL Server 2012. Appropriate stored procedures, indexes, constraints, and triggers are employed to enforce the validity, security, consistency, and integrity of data in the business layer.

While it is necessary that the developer creates its own class of frameworks to satisfy the requirements of this design, ready-made software packages implementing business objects and loosely coupled components may be used as long as the core concept of three-tiered architecture is followed to ensure compliance and conformity to this design. The following framework and free packages are widely available on the internet and may be used to develop the system. The choice of the package or combination of packages depends on the specific skills and preference of the IT firm and should be discussed with and approved by the project.

- *Microsoft Smart Client Software Factory*
- *Windows Communication Foundation*
- *Model-View-Presenter*
- *Model-View-Controller*