

A Generic and Reliable Land Acquisition Protocol Implementation for Sub-Saharan Africa Countries

Bayomock Linwa André Claude, Traore Hamidou

Department of Computer Science, International University of Grand-Bassam, Grand-Bassam, Ivory Coast

Email: bayomock.a@iugb.edu.ci, traoreh2@iugb.edu.ci

How to cite this paper: Claude, B.L.A. and Hamidou, T. (2024) A Generic and Reliable Land Acquisition Protocol Implementation for Sub-Saharan Africa Countries. *Journal of Computer and Communications*, 12, 208-245.

<https://doi.org/10.4236/jcc.2024.121015>

Received: December 16, 2023

Accepted: January 28, 2024

Published: January 31, 2024

Copyright © 2024 by author(s) and Scientific Research Publishing Inc. This work is licensed under the Creative Commons Attribution International License (CC BY 4.0).

<http://creativecommons.org/licenses/by/4.0/>



Open Access

Abstract

Land acquisition is subject of many frauds worldwide. In Africa countries, land acquisition frauds are a real curse. Main reasons of the high land acquisition frauds in Africa are a lack of a reliable land acquisition protocol implementation, no centralized information system that records land transactions and integrates all actors, lack of education about the acquisition protocol and lack of applied sanctions in case of frauds. As of sanctions, the issue is bound to the corruption plague in Africa. Authorities that are supposed to rule the laws are not well paid and are part of the corruption scheme. Most frequent frauds are multiple sales of the same land, falsified land title, false owner identification. In Africa, a land belongs generally to the first residents of the area. To claim ownership of a land, an owner should be recognized by the rural or city authorities. To achieve this recognition, he must obtain from those authorities an attribution letter. The author proposed a complete and reliable land acquisition protocol named ACLAP (Africa Countries Land Acquisition Protocol). The purpose of this paper is to explain the main components of the protocol implementation and show its efficiency when confronting fraud threats. In this paper, a Generic and Reliable Land Acquisition Protocol Implementation in Sub-Saharan Africa countries context is proposed. The platform keeps relevant land transaction information for administrators, managers, buyers, sellers, or for people consultation purpose. The application may be used in village, city, department, region, or country for land management decision making.

Keywords

Land, Acquisition, Protocol, Reliable, Africa

1. Introduction

Land is the subject of the interest in the land acquisition. What is a land? In literature many land definitions are published. But in the context of this paper, a land is an earth area that has well known geographical boundaries. A land has adjacent lands and may have vegetation, rivers, forest, animals, and other things included in the identified area. Acquiring a land is a process that aims at the end to assign a land ownership to a buyer, a physical or moral person.

Meanwhile questions remained as to whether the seller is really the owner. Does he have the right to sell? When managing villages, cities, departments, regions or countries, the administrators' main challenge is to first know or establish the geographical boundaries of the area under their management. Main actors (seller, buyer, and/or domain administrators) of a land acquisition endeavor are confronted with the same problem. In addition to areas limits, answers must be provided to other questions as the land ownership determination problem? Does the land belong to the seller? Does the seller have the required owner representative rights as per the law?

In sub-Saharan Africa countries, the land first owners are the inhabitants, *i.e.*, the families that first established themselves in the concerned areas, Mwangi Gideon W. (2011). Their land area limits are generally recognized using landscape signs as lakes, trees, or roads. Most of the time, the physical recognition of those signs is not recorded on papers or electronically, making it very difficult for those that have been instructed about those limit signs to be fair, concise or precise in the recognition of the area limits. With years and the effects of nature, used signs may disappear, and may create dispute scenarios.

In the case of a land mutation (sell or rent), the land owner is sometimes questionable. The land owner anterior history may require laws to be said and applied. These aggrieving issues make the land acquisition process in Sub-Saharan Africa countries a real bottleneck and something to be very cautious before entering into.

The authors [1] proposed a robust and reliable land acquisition protocol (ACLAP) that takes in account those challenges and minimizes the frauds using a strong control of data integrity.

The acquisition process is a set of activities/services that are well-orchestrated to meet the final goal of assigning the land to the right person. The services orchestration is included in an architecture that takes into account the requirements that the proposed implementation must solve. These services are discovered, and structured and their related data model is elaborated. The minimal infrastructure required to deploy the land acquisition is designed considering the security constraints aspects.

Solutions proposed relied in land management or administration without strong considerations to solve land acquisitions frauds. None of the proposed solution is built through a protocol that takes into account all actors of the land acquisition process.

The organization of this paper is as follows: after the introduction, the motivations that sustain this work are explained, and then a summary of related work is provided, followed by the requirements land acquisition. Afterwards, the proposed architecture is presented including the data model, the services and their interactions; then the implementation guides, technologies and codes are explained. The obtained results are published and analyzed; finally the conclusion is given.

2. Motivations

We have three main motivations for this work and these are: economic, social and reduction of potential conflicts.

Land is an important resource for a nation. Generally defined as a part of earth not covered by water, the land is the first and most important resource of a country. It is used for habitation, for agriculture, for mines extraction, for petrol extraction, to bury human bodies, to build or support infrastructures, and more. Some of its values are bound to human works and other values are nature gifts. Land is thus very important for international and national perspectives. A country must know its area boundaries and the boundaries with its neighbor countries. Country boundaries are generally recognized by United Nations Organization. Internally, a country has administrative subdivisions. Those administrative structures put in place for governance purpose are divided in regions, departments, cities, or villages. Each of those divisions must be clearly identified geographically and its boundaries must be known. Land revenue generated by a country can then be drilled down to the granularity of a region, department, city or village to show the contribution of each subdivision to the economic wealth of a country. Therefore, the land and its economic aspects (boundaries, geographical situation, pictures, detected natural resources, and evaluation) must be well identified and recorded in a land information system.

In some communities of Sub-Saharan country (e.g. Ivory Coast), a land belongs to a cultural/traditional group. These cultural groups have implemented a system to manage social problems and these organizations oversee the development of activities in their respective region. The leaders of those communities play an important role in applying the policy put in place. In this regard, their decisions are recognized by the government. On the other hand, in the cities, the government has administrative personnel that is in charge of the land acquisition process. A land information system must identify all authorities that are involved in the land acquisition approval and integrity, determine the land owner and the land buyer, register the land acquisition transaction and the land boundaries. The land information system must record also the historical information to maintain the traceability of owners, authorities or others actors over time.

Even when the rules and land acquisition processes are well implemented, there are some gray parts in the process that may generate conflictual situations.

Non honest individuals may sell the same land to many potential acquirers before the transaction is actually recorded in the system. Causes of this issue may be due in fact that a dishonest owner has some complicity at certain decision level or the buyer may not be aware of the whole land acquisition process. Other scenarios may be a fraudster that sells a land that he does not own to one or many clients. Thus, a land information system can be critical in helping avoid some of these frauds. In the meantime, the use of the land information system must be emphasized and people involved in a land purchasing must be educated on the official land acquisition process to reduce frauds percentage. And potential acquirers may thereby consult available land information to know more about the most recent data about the coveted land.

Many countries, mainly those that are under-developed, are not using reliable data as inputs in land costs. This means that the error margin and incertitude of the provided costs are generally unknown and the stated cost may be way higher than the normal one. With a reliable land information system, a higher quality of recorded data can be used as inputs in the land functions dealing with costs and can also be instrumental in readjusting the land governance policies. In fact, Land governance is a set of policies that go across different administration units to optimize the land cost, usability, tenure and/or development of land management [2].

In Sub-Saharan Africa countries, almost each person knows at least one relative (friend, family member) that has been abused in a land acquisition transaction. Sometimes, even tactical and temporary solution, shared by experienced people may not work. Considering these facts, software research may contribute in reducing the abuse risks through a Generic and Reliable Land Acquisition Protocol Implementation for the Sub-Saharan Africa countries context. A Generic and Reliable Land Acquisition Protocol (ACLAP) proposed by [1] will be used to support the implementation.

3. Related Work

Related publications may be classified in three categories: land governance, land acquisition, and land information system frameworks.

Land governance is a set of business functions or objectives, resource supports, constraints, data, and organization of administrative units, business processes and their information flows, delivery of required documents with the goal to optimize the land administration costs. The implementation of a governance vision requires the usage of a structured framework that drives its deployment through the different administrative units. The author in [3] proposes a framework GLAP (Global Land Administration Perspective) that has four business functions: land tenure, land use, land value and land development. The *land tenure* englobes activities as the processes and institutions related to securing access to land and inventing commodities in land, and their allocation, recording and security; cadastral mapping and legal surveys to determine parcel boun-

daries; creating new properties or altering existing properties; the transfer of property or use from one party to another through sale, lease or credit security; and the management and adjudication of doubts and disputes regarding land rights and parcel boundaries. The *land value* contains the processes and organizations in charge of the land value assessment and properties; the tax revenue calculation and revenue gathering; and the management and adjudication of land valuation and taxation disputes. The *land use* controls the processes and institutions of land use through adoption of planning policies and land use regulations at national, regional and local levels; the enforcement of land use regulations; and the management and adjudication of land use conflicts [4]. The *land development* is related to the processes and institutions in charge of building new physical infrastructure and utilities; the implementation of construction planning; public acquisition of land [5]; expropriation; change of land use through granting of planning permissions, building and land use permits; and finally the distribution of development costs [6]. Authors in [7] propose a framework named FELA (Framework for Effective Land Administration). FELA provides a framework that drives the implementation of country sustainable development goals. FELA also guides for the establishment of responsible land governance structures, effective and efficient land institutions, and leadership in the land sector, with clearly defined propositions, ideally centered upon transparency, accountability, inclusiveness, accessibility, participation, gender and youth sensitivity, consideration of the vulnerable and marginalized people. FELA has six business objectives: land use, land development, land tenure, land value, land policies and integrated geospatial information. Land Policies objective is an additional business objective compared to GLAP, and it is related to laws and policies surrounding the establishment, renewal and strengthening of land administration and management systems; it is to be driven by responses to social, economic and environmental needs. In general, having holistic land policies based around sustainability will result in a more efficient legislative design and administration [8] [9]. The integrated geospatial information provides a basis, a reference and a mechanism to establish or improve national geospatial information management arrangements. It coordinates alignment activities between existing national capacities, and infrastructures, including in the land sector.

Both GLAP and FELA are high quality frameworks used as standards in the development of land administration management. However, those frameworks are guidelines to optimize land governance and do not define how information could be extracted from processes to build a business intelligence system that sustain decision making and reduce land conflicts.

Another research axis is the “land management applications”. In [10], a land information system for villages and towns is proposed. It handles land usability, farm land protection, and land registration with spatial geo-information, laws enforcement and maintenance services. In [3], the author proposes the land information systems for Denmark. The solution is based on GLAP framework and

large module interactions without a clear identification of their interfaces with the business intelligence system.

One important land research axis is the land planning, use and management. In [11] [12], the authors propose an approach to measure land use and management through audit functions. The approach requires collecting evaluation measures of land potentials, as well as measures relative to how the land has been used so far, then the environment impact measures of the land usability. The authors in [13] [14] use the same evaluation measures to take land planning decisions relative to the land management orientations. Both papers are very high quality but they do not elaborate on the land management infrastructure support: the land information system.

Other aspect is the implementation GAP of the LIS (Land Information System) in developed countries versus developing countries. Unlike in developed countries where LIS are well implemented and largely used, implementing a LIS in developing countries is challenging [15] [16]. In developing countries where such information systems are in place, they remain latent; hence, essential decisions are made based on limited information [17]. For example, after efficiently implementing LIS in Rwanda, the bureaucratic nature of the registration system discouraged landowners from registering their properties, thereby making their LIS non-reliable for decision making [18]. Similarly, Kenya's LIS is not maintained and their data records are not up to date due to informal registration that is not documented [19] [20]. Other challenges may result from inadequate land data, poor data quality, and the government's unwillingness to invest in LIS [21] [22]. In the scope of innovation in information technology such as LIS, the authors in [23] argued that the system's adoption and usage are crucial factors in achieving the overall success of implementation. LIS must be used to achieve the expected outcome; however, this is often not the case in many developing countries. Liberia, for instance, has failed to integrate their LIS into their land administration function [24].

Our proposed is based on a simple, reliable land acquisition that include the main actors of the process. But its success is bound to the education of the users that must know about its existence and broking the resistance of administrative that are comfortable with the lack of strong rules of current land acquisition process in Sub-Saharan countries.

4. Used Land Acquisition Process and Protocol Description

4.1. Land Acquisition Process

The land acquisition process is varying from one country to another. In [1], the land acquisition processes of three selected countries are described: Côte d'Ivoire, Senegal and Cameroun.

1) Côte d'Ivoire

The land acquisition involves five main actors: the seller, the buyer, the approval committee that confirms the transaction, the urban administrator repre-

sentative, the domain administrator representative [25]. The buyer is the land acquirer. He collects from the seller, the existing land documents that he wants to buy. Then he checks with an urban administrator representative (MCLAU representative) about the current owner and if the land is not a private domain of the nation. If there is no problem with the checks, he may continue the acquisition process or stops. In case he chooses to continue the process, he will then proceed to the payment part by first negotiating the payment date with the approval committee. Then an invitation is sent to the approval committee team members about the land acquisition payment. At the payment date, the buyer pays the land at the negotiated price along with the related administration fees. Afterwards, a bill receipt is delivered immediately to him. And one to two weeks later, an attribution letter signed by the highest member of the approval committee team is delivered to the buyer. The buyer introduces an ACD (Attestation of definitive concession) request to an urban administrator with all requested documents (technical drawing and location, request fees, tax of land title). The urban administrator checks the genuineness of the provided documents and transfers the file to the domain administrator for the land title establishment. After verifying the successful completion of land title, an urban administrator delivered the ACD to the buyer.

2) Cameroun

In Cameroon, the land acquisition is done in five main phases and involves six main actors: the buyer, the seller, the geometer, the urban agent/administrator, the notary, and the domain agent/administrator [26] [27]. The first phase is the land split and mutation. In this phase the seller wants to sell part or total area of his land. The seller is the initiator of this act. He fills the request of land split or mutation in favor of the buyer. The land split or mutation form signed by all parties is then joined to the technical documents produced by a geometer after the drawing of various graphs regarding the topography of the land, the geographical area limits of the land, the position of the land with respect to other well registered and adjacent lands and their submission to an urban administrator. In the second phase, the request form is processed by a cadastral administrative unit and a validation form is given attesting the conformity of produced documents as well as the urban map road plan in the concerned area. In the third phase, when the urban administrative unit receives a positive feedback from the cadastral unit, he then issues an urban certificate. Using the urban certificate, the seller and buyer meet with a notary to conclude the land purchase. The payment of the land at the agreement price is done in front of the notary at the fourth phase. At the fifth phase, the land certificate is established by the notary and a copy is given to the buyer.

3) Senegal

In Senegal, the land acquisition is done in four main phases and involves six main actors [28]. The first phase is a request introduced by the seller to the cadastral administrative unit. The request goal is to obtain the cadastral drawings

that determine the geographical area limits of the land, the position of the land with respect to other well registered and adjacent lands, the land classification (public, individual private, national private domain, or national domain). In case the land belongs to an individual, the purchase must be completed in the presence of a notary and he/she must produce an official purchase document that is joined to the land title request. Then, the seller in conjunction with the buyer make a land title request to APIX (a central organization that is handling all land titles' processing) organization. The file received by APIX must contain the following documents: the purchase form signed by a notary, the cadastral land drawings, the building house project architecture drawings and engineering, project cost estimates, the number of jobs created. After a successful land title request processing, the APIX delivers a land title to the buyer.

4.2. Proposed and Used Land Acquisition Protocol ACLAP

1) Land Acquisition Protocol Work Flow

The land acquisition processes in Cameroun, Senegal and Côte d'Ivoire have similar type of actors [1]. But those actors are sometimes named differently. For example, in Cameroun, the urban agent/administrator actor is equivalent to MCLAU agent in Côte d'Ivoire and an APIX agent in Senegal. In the generic work flow of land title acquisition process (Figure 1), we are using the urban agent word to identify the concerned agent. For integrity, transparency and earlier fraud detections, a generic work flow of land title acquisition is proposed. This work flow has the advantage that it can be applied in many African countries as they have similar land acquisition processes. The integrity and conformity of documents are part of the work flow orientations and decisions.

2) ACLAP Protocol Description

a) The buyer checks or lookups for land on sale in a particular area. He/She may use different channels (web sites, well known sellers, recommended sellers, etc.). The buyer must provide his metrics in terms of price, location, square area, etc...

b) The seller agent sends back available lands that match the given criteria along with the associated ownership documents (land title, land location, land drawings, etc.).

c) The buyer picks the land that he likes and sends the corresponding land ownership documents to an urban agent for an integrity verification.

d) The urban agent checks the documents conformity with the domain administrator and sends back his response about the conformity of the documents. He also includes in the response whether the land's ownership has not changed in the past or whether it is in the process of changing ownership.

e) In case the land ownership is good, the seller requests to the urban agent the design of the land drawings that match the targeted land.

f) The urban agent forwards the request of the land drawing task to the cadastral services, and then, the request is assigned to a geometer.

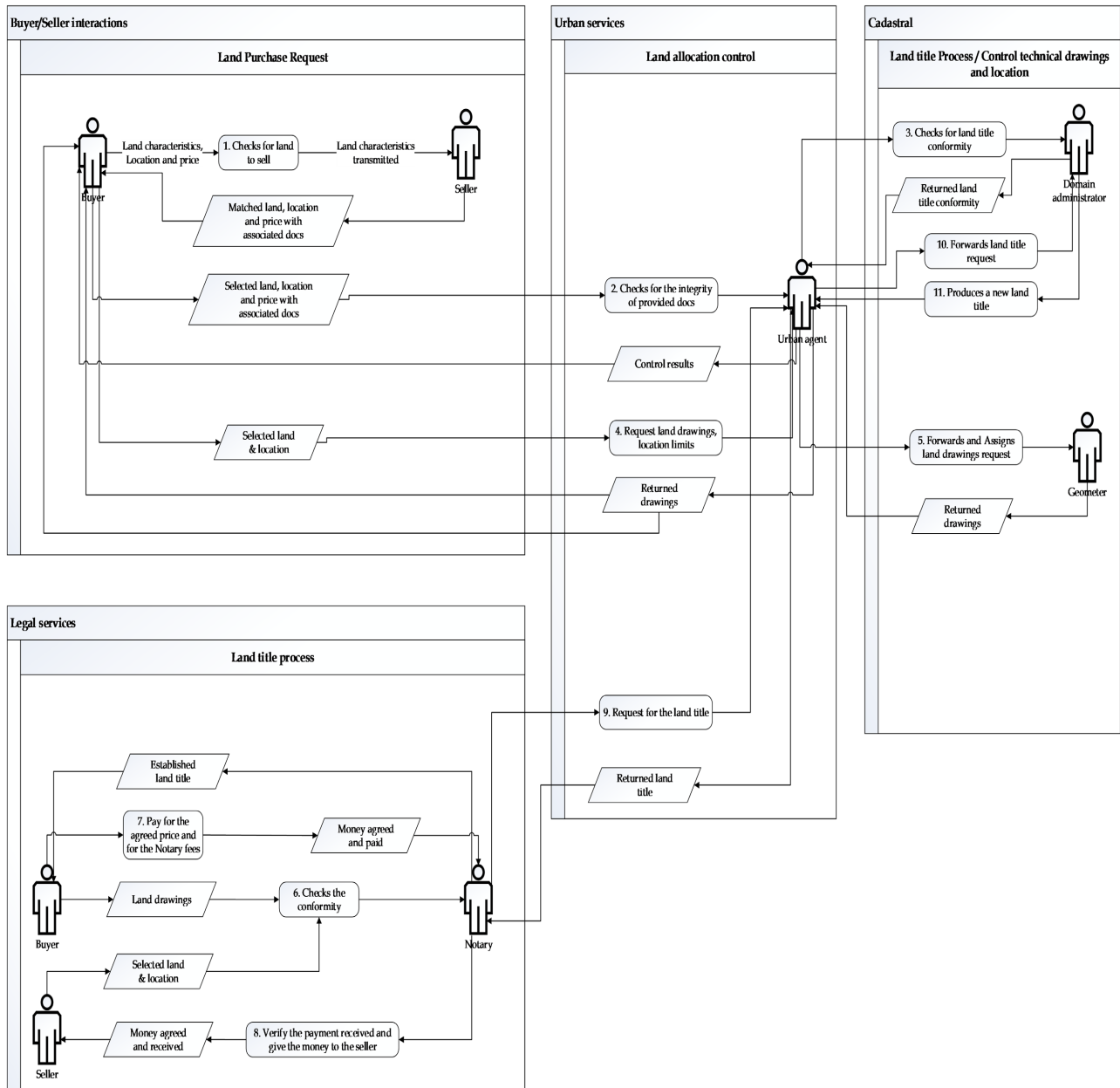


Figure 1. Generic land acquisition work flow of ACLAP.

- g) The geometer makes the land drawings and the cadastral services return them to the urban agent.
- h) The urban agent then gives a copy of the land drawings to the buyer.
- i) The buyer and the seller meet with the notary for the payment.
- j) The buyer and the seller provide to the notary the original land documents as well as the approved documents and land drawings obtained from the urban agent.
- k) The notary checks the conformity of the documents.
- l) After a successful land conformity check, the buyer pays for the agreed price and fees, and gives the money to the notary.
- m) The notary verifies that the amount matches the agreed price plus his own

fees.

n) The notary then gives to the buyer his due and produces two receipts one for the seller and one for the buyer.

o) The notary requests to an urban agent a land title and ownership change on behalf of the buyer and the seller.

p) The urban agent forwards to the cadastral services a land title request establishment.

q) The domain administrator then checks for the land title and establishes a land title and returns it to the urban agent.

r) The urban agent notifies and sends the land title to the notary.

s) The notary notifies the buyer and delivers to him/her a land title.

5. Protocol Requirements and Solution Approaches

The requirements that are in scope of the implemented protocol requirements are: integrity, confidentiality and reliability.

5.1. Integrity

The importance of integrity in the land acquisition is very high. We define as integrity, the protection of the important data collected and saved in the information system of the land acquisition. These collected data that must be protected are related to the identity of the application actors (buyer, seller, owner, notary, land title administrator, urban agent, and urban administrator) as well as land document authenticity (urban land certification, attribution land letter, or land title).

An attacker of the system may be external or internal to the involved organizations that have access rights on the information system resources. The proposed solution takes in account the different cases (**Table 1**).

5.2. Confidentiality

Sensitive data that are intercepted or read from the database won't be exploitable by the attacker as they will be encrypted by a RSA or 3DES Key (**Table 2**).

5.3. Reliability

Users are expected to use the land information system anytime. Degradation of the quality of service should not stop the user accessibility to the information system. To achieve this requirement (**Table 3**), servers must be redundant to allow the service relay in case the primary server instance failed to deliver services, the secondary server instance will be used. Strong security policy must also be put in place to protect the software and other system resource from attacks as the DoS (Denial of Services).

6. Proposed Architecture

6.1. Infrastructure Architecture

The infrastructure architecture is composed of client and server subsystems

Table 1. Integrity solution approach.

Threat	Description	Solution	Technologies
Theft of the identity	Theft of a user identity consists to take advantage of the weak access rights constraints to the information system or the lack of users' IT knowledge and then obtain their credentials to the information system.	In case the attacker got credentials of an authorize user as the username, it is imperative to make it hard to him to get access to the sensitive data of the information system. The proposed solution is to use something that belongs to the user meaning the <i>biometric technology</i> .	The iris biometric is the recommended one. It is known as the best biometric in term of accuracy (the image match incertitude is less than 1%).
Alteration of an authentic document	The attacker may use a replay technique by altering an original and authentic document received from the information system.	The solution consists to stamp a shared document with a date and a unique code. The code is uniquely recorded in the system and it is signed with a private key. Any shared document has a validity of 24 h maximum.	The size of the code is an important factor in the solution. A 512 bits is challenging enough when it is coupled with a RSA encryption algorithm.

Table 2. Confidentiality solution approach.

Threat	Description	Solution	Technologies
Plain text document captured during data transportation	Non encrypted data sent between two end machines that are communication, may be captured by an attacker.	The solution resides to use a strong encryption algorithm (3DES or RSA).	Use RSA private and public keys. A key size of 2048 bits is an excellent choice.
Non encrypted and Saved sensitive data	An attacker that works for an organization that produces and has access to sensitive data may use this advantage to attack the information system.	The solution resides in encrypting sensitive data first, then save them in the storage system. The use of a strong encryption algorithm (3DES or RSA) is recommended.	Use RSA private and public keys. A key size of 2048 bits is an excellent choice.

Table 3. Reliability solution approach.

Threat	Description	Solution	Technologies
DoS attack	Prevent the information system to use its resource, pushing to a big degradation of the quality of services.	Put in place a network and application firewall as well a good anti-virus tool to preserve server machines.	Use RSA private and public keys. A key size of 2048 bits is an excellent choice.
Degradation of services offered	The information system is not able to deliver an acceptable quality of services.	A redundancy plan is recommended.	Secondary server should have similar characteristics as the principal server.

(Figure 2).

- The client subsystem is the set of machines used by clients (buyer, seller, urban agent/administrator, domain administrator, geometer, and notary). Those machines are either mobile phones, tabs, laptops, or computers. They are connected to the proposed solution through Web browser and Internet Protocol.
- The server subsystem is the set of Web servers (main server and backup

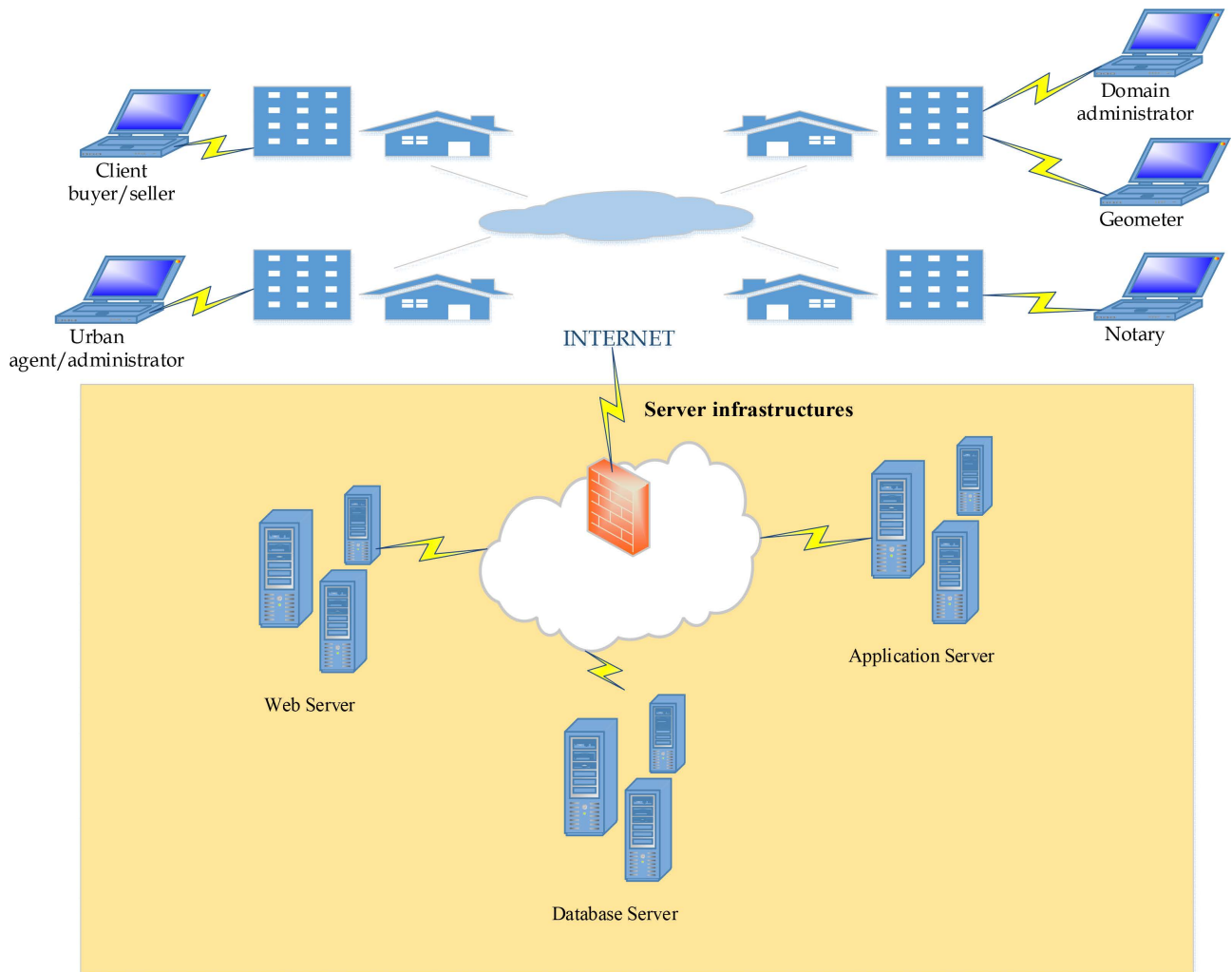


Figure 2. Infrastructure architecture.

servers), application server and database servers (urban and domains servers). The server subsystem contains the application software. The database subsystem manages the land data and the land transaction information. The database subsystem contains the set of metadata and data required and that supports the land application.

6.2. Software Architecture

The software architecture (**Figure 3**) is made of 6 layers: web pages' orchestration and navigation, web listeners' services, web views, entity management, data entities and data access management.

- The web pages' orchestration and navigation layer contains web pages and the navigation flow between web pages. A web page is designed using the MVC pattern. Each web page is bound to a view object and may incorporate listeners to capture the user events.
- The web listeners' service layer is a set of classes that handle user events on the web pages. The web page user events are bound to the listeners. The listener

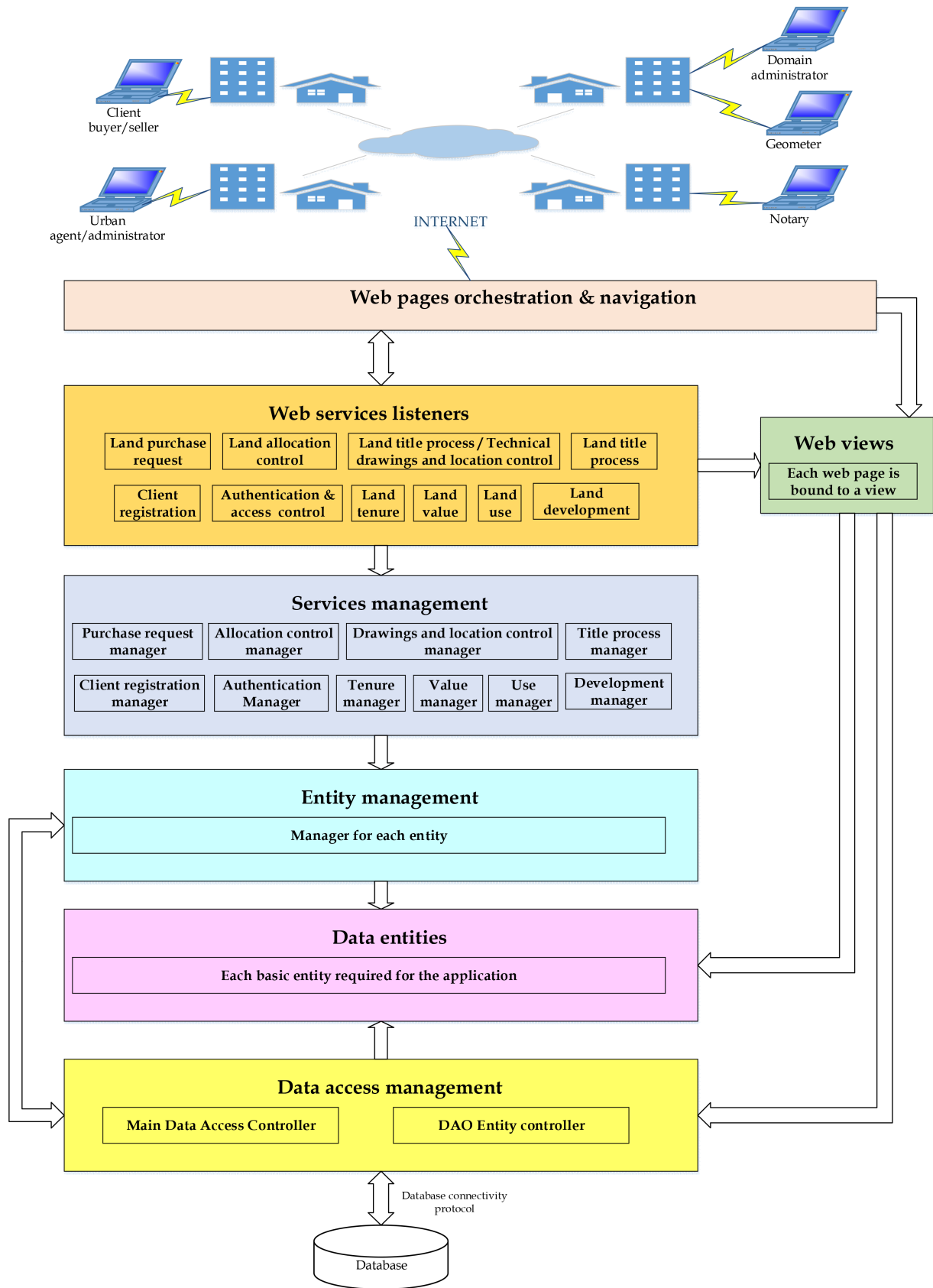


Figure 3. Software architecture.

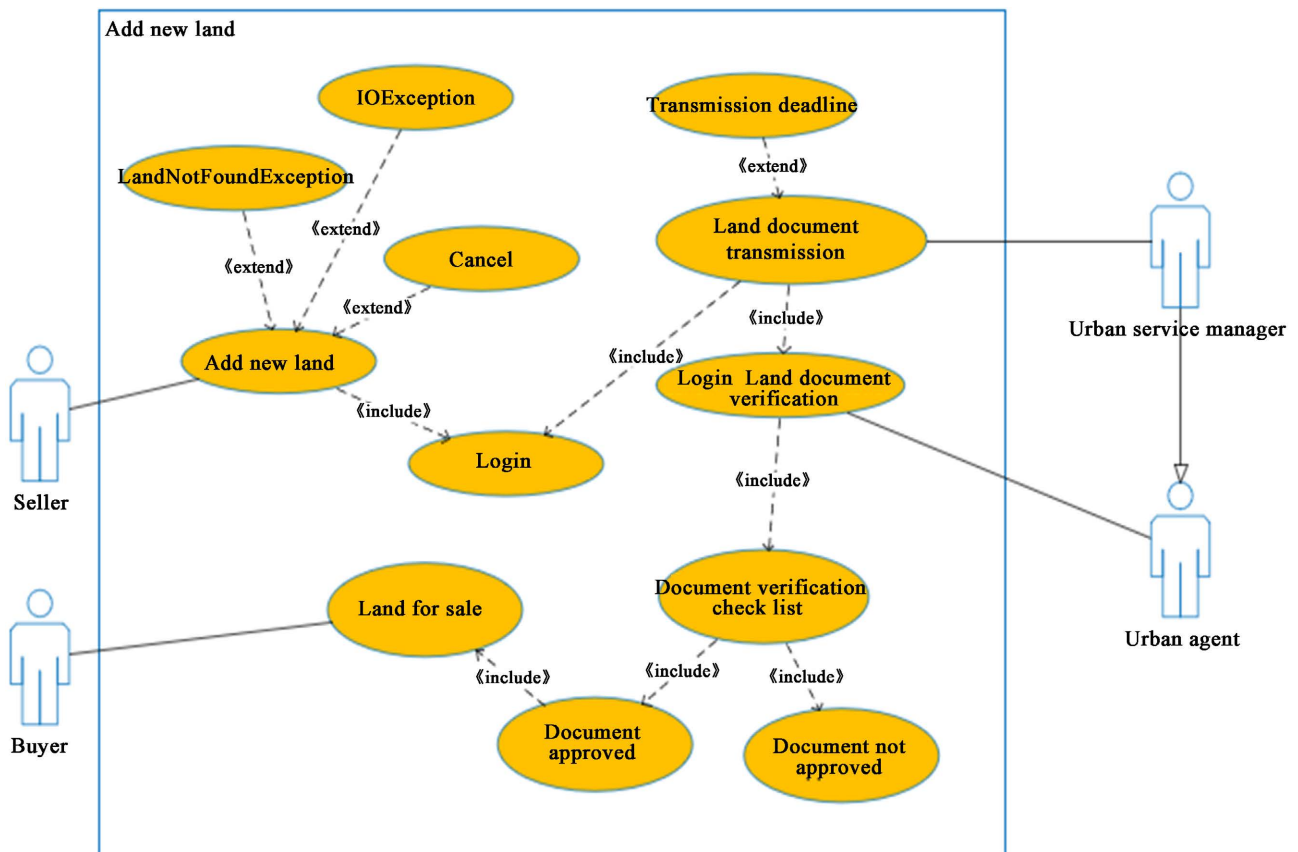


Figure 4. New land registration' use case diagram of ACLAP 1st implementation from [9].

performs the requested action and returns the results. The results are rendered by calling a returned page or by displaying a dialog message.

- The web view layer is a set of classes that encapsulates the application data displayed on a web pages. Each class acting a view, has a scope (application, session, view or request). The view in an application scope is acting as a global or shareable view and its object content is applied to all application users. In case of the session scope, the view is active for a user full session. In case of the view scope, the view is active as long as the view is still used in the web page navigation. In the request scope, the view is active only for the current request.
- The entity management layer contains the set of classes that manages (creates, finds, updates and/or deletes) the basic or plain data entity.
- The data entities layer contains the set of plain classes required to build the application. In general, each data entity is mapped to a database table.
- The data access layer contains the class or classes that interact with the database tables. The interaction with the database is achieved via a database connectivity protocol.
- The database subsystem contains the set of metadata and data required and that supports the land application.

6.3. Use Cases

The implementation of the solution is an ongoing process. A use case is a service offered to the users to interact with the server machines. Due to the implementation complexity and limited funds for the project, for the first implementation version, the following use cases have been taken in account (**Figure 4**). In **Tables 4-15**, the main use cases of the application are described.

Table 4. Add urban/cadastral/notary manager account.

Use case Name	Add Urban/Cadastral Manager Account
Primary actor	System administrator
Secondary actor	Application system
Brief description	This use case allows the system administrator to add urban/cadastral manager login account to the application system.
Pre-condition	<ol style="list-style-type: none"> 1. The administrator is logged into the application system. 2. The application system displays a welcome page with all services.
Normal flows	<ol style="list-style-type: none"> 1. The domain administrator clicks on “Account Services” button 2. The system returns the “Account Services” that contains the following items: <ol style="list-style-type: none"> a. Urban Services b. Cadastral Services c. Notary Services 3. In case, the domain administrator selects “Urban Services”, then the system returns a table containing users’ information plus the (create, refresh, search) buttons at the top left corner of the table. 4. The domain administrator selects “Create”, then the system returns the following information: <ul style="list-style-type: none"> ✚ Domain administrator ID ✚ Manager code ✚ Service name ✚ first name ✚ last name ✚ Nationality ✚ Nickname ✚ City ✚ Address 1 & 2 ✚ email ✚ user name ✚ password ✚ Role ID ✚ Postal code ✚ Photo 5. The administrator fills the information, then clicks on “Save” 6. The system validates the information and returns a successful account creation or a failure message. 7. In case of successful creation, either the urban manager, the cadastral manager or the notary receives an automate email for successful account creation that is strongly requesting him to update his account password between 24 hours. 8. The system extracts the data entered by the domain administrator and add new record to the account data table.

Continued

Alternate flows	<ol style="list-style-type: none"> 1. The domain administrator enters username that already exists. 2. The system displays an error message “Username already created, please try another one”. 3. The domain administrator can either try again or abort the operation by clicking on “Cancel” button 4. If the administrator clicks on “Cancel” button, the system returns back the previous page.
Post-condition	<ul style="list-style-type: none"> • The urban manager account is created • The cadastral manager account is created
Exceptions	<ul style="list-style-type: none"> • Username already created • I/O Exception • MessagingException

Table 5. Find urban/cadastral/notary manager account.

Use case Name	Find Urban/Cadastral Manager Account
Primary actor	System administrator
Secondary actor	Application System
Brief description	This use case allows the administrator to search Account record in the application system.
Pre-condition	<ol style="list-style-type: none"> 1. The user is logged into the application system. 2. The application system displays a welcome page with all services.
Normal flows	<ol style="list-style-type: none"> 1. The user clicks on “Account Services” button 2. The system returns the “Account Services” sub-menu which contains the following items: <ol style="list-style-type: none"> a. Urban Services b. Cadastral Services c. Notary Services 3. The domain administrator selects the service he/she wants to visit 4. In case, the administrator selects “Urban Services”, then the system returns a table containing users’ information plus the (create, refresh, search) buttons at the top left corner of the table. 5. The domain administrator selects the “Search” button, then the system returns an input window. 6. The domain administrator enters a keyword and clicks on “Search” button to display the record associated to the keyword entered. 7. In case, the administrator selects “Cadastral Services”, then the system returns the similar information as at step 4 8. The system displays the corresponding record that matches the keyword entered by the domain administrator.
Alternate flows	<ol style="list-style-type: none"> 1. The administrator enters a keyword that does not match any record. 2. The system displays an error message “Record not found”. 3. The administrator can either try again or cancel the operation by clicking on “Erase” button. 4. If the administrator clicks on “Erase” button, the system returns back to the table containing the users’ information page.
Post-condition	If the use case was successful, the system displays the record that match the entered keyword. Otherwise, the system displays no record.

Table 6. Delete/deactivate urban/cadastral manager account.

Use case Name	Delete/Deactivate Urban/Cadastral Manager account
Primary actor	System administrator
Secondary actor	Application System
Brief description	This use case allows the system administrator to delete/Deactivate Account record from the application system.
Pre-condition	<ol style="list-style-type: none"> 1. The user is logged into the application system. 2. The application system displays a welcome page with all services
Normal flows	<ol style="list-style-type: none"> 1. The administrator clicks on “Account Services” button 2. The system returns the “Account Services” sub-menu which contains the following items: <ol style="list-style-type: none"> a. Urban Services b. Cadastral Services c. Notary Services 3. In case, the user selects “Urban Services”, then the system returns a table containing users’ information plus the (create, refresh, search) buttons at the top left corner of the table and three buttons (Edit, Delete, PDF) on the right end of the table. 4. The administrator clicks on “Delete” button 5. The system returns a window with “Yes” and “No” to confirm the deletion. 6. In case the administrator selects “Yes”, the system extracts the user data with the id and delete the record associated from the database. 7. The operation is canceled if the administrator selects “No” 8. The system returns to the previous state.
Alternate flows	<ol style="list-style-type: none"> 1. The administrator can either try again or cancel the operation by clicking on “No” button. 2. If the administrator clicks on “No” button, the system returns back to the previous state.
Post-condition	If the use case was successful, the selected record is deleted from the system. Otherwise, the state of the system is unchanged by the use case.
Exceptions	The administrator aborts the delete operation by clicking on “No” button at step 7 of the Normal Flow

Table 7. Update urban/cadastral/notary manager account.

Use case Name	Update Urban/Cadastral/Notary Manager Account
Primary actor	System administrator, Urban/Cadastral Manager
Secondary actor	Application System
Brief description	This use case allows the user to update an account record in the application system.
Pre-condition	<ol style="list-style-type: none"> 1. The user is logged into the application system. 2. The application system displays a welcome page with all services.
Normal flows	<ol style="list-style-type: none"> 1. The user clicks on “Account Services” button 2. The system returns the “Account Services” sub-menu which contains the following items: <ol style="list-style-type: none"> a. Urban Services b. Cadastral Services c. Notary Services

Continued

3. In case, the user selects “Urban Services”, then the system returns a table containing users’ information plus the (create, refresh, search) buttons at the top left corner of the table and three buttons (Edit, Delete, PDF) on the right end of the table.
4. The domain administrator clicks on “Edit”, the system returns the following information:
 - ✚ Domain administrator ID
 - ✚ Manager code
 - ✚ Service name
 - ✚ first name
 - ✚ last name
 - ✚ Nationality
 - ✚ Nickname
 - ✚ City
 - ✚ Address 1 & 2
 - ✚ email
 - ✚ user name
 - ✚ password
 - ✚ Role ID
 - ✚ Postal code
 - ✚ Photo
5. In case the user selects “Cadastral Service” or “Notary Services”, then the system returns the similar information as at step 3
6. The user fills the information, then clicks on “Save”
7. The system validates the information and returns a successful account update.
8. In case of successful update, either the urban manager, the cadastral manager or notary receives an automate email for successful account update.
9. The system extracts the data entered by the user and update the record to the account data table.

- Alternate flows
1. The system displays an error message “Record not found”.
 2. The administrator can abort the operation by clicking on “Account Services” button.
 3. If the administrator clicks on “Account Services” button, the system sends back the previous page.

Post-condition If the use case was successful, the record is updated in the system. Otherwise, the state of the system is unchanged by the use case.

Exceptions The user aborts the update operation by clicking on “Account Services” button.

Table 8. Add buyer/seller account.

Use case Name	Add Buyer/Seller Account
Primary actor	Buyer, Seller
Secondary actor	Application system, Urban Service Manager
Brief description	This use case allows the urban service manager to add a buyer or a seller account to the application system.
Pre-condition	The user does not have an account created yet.

Continued

Normal flows	<ol style="list-style-type: none">1. The buyer/seller clicks on “Sign Up” then2. The system displays the following:<ol style="list-style-type: none">a. As Buyerb. As Seller3. In the case the user clicks on “As Buyer”.4. The system displays the “Add New Buyer” page.5. The user enters the following information:<ul style="list-style-type: none">✚ First name✚ last name✚ email✚ user name✚ nickname✚ password✚ address✚ phone number✚ parent (father and mother name)✚ Citizenship✚ P.O. Box✚ Father’s full name✚ Mother’s full name✚ picture (current photo)✚ ID Card picture (current photo)✚ Document picture (user should write his name, date, time, signature on a paper, then taped his id card on that paper and holds the document near his head to take a new picture then upload it in the system)✚ Account Type6. The user then clicks on “Save” button.7. The system extracts the data entered by the user and add new record to the “buyer” account data table.8. The Urban Manager checks manually the conformity of the user account creation with an urban agent.9. In case, the user account data is valid, the urban service manager sends a successful account creation email to the user, otherwise, if the user account data is not valid, then the urban service manager delete the user account data and sends a failure account creation email to the user.10. In the case the user clicks on “As Seller”, the system returns similar scenarios to those of “As Buyer” account creation.
Alternate flows	<ol style="list-style-type: none">1. The system displays an error message “Username already created, please try another one” or “Unknow response from server”.2. The user can either try again or cancel the operation by clicking on “Cancel”3. If the user clicks on “Cancel”, the system returns back the home page.
Post-condition	If the use case was successful, the urban service manager sends an email “Congratulation your account has been successfully created” to the user.
Exceptions	<ul style="list-style-type: none">• Username already created warning message• The user aborts the operation by clicking on “Cancel” at step 3 of the Alternate Flow.

Table 9. Find Buyer/Seller account solves: FR13, FR27.

Use case Name	Find Buyer/Seller Account
Primary actor	Urban Service Manager
Secondary actor	Application System
Brief description	This use case allows the urban service manager to search for buyer, seller account record in the application system.
Pre-condition	<ol style="list-style-type: none"> 1. The user is logged into the application system. 2. The application system displays the home page with all services.
Normal flows	<ol style="list-style-type: none"> 1. The urban service manager clicks on “Manage Buyers Information” button 2. The system returns the “List of buyers” page. With a search bar at the top of the list. 3. The urban service manager enters a keyword (code, first name, last name, e-mail, phone number) and clicks on “Search” button 4. In the case, the urban service manager selects “Manage Sellers Information”, then the system returns the similar information as at step 3. 5. The system extracts the keyword and displays the corresponding record that matches the keyword entered by the urban service manager.
Alternate flows	<ol style="list-style-type: none"> 1. The urban service manager enters a keyword (code, first name, last name, e-mail, phone number) that does not match any record. 2. The system displays an error message “No record found”. 3. The urban service manager can either try again or cancel the operation by clicking on “Go Back” button. 4. If the urban service manager clicks on “Erase” icon, the system displays back the “List of users” page.
Post-condition	If the use case was successful, the system displays the record that match the entered keyword. Otherwise, the system displays no record.

Table 10. Delete/deactivate buyer/seller account management.

Use case Name	Delete/Deactivate Buyer/Seller Account
Primary actor	Urban Service Manager
Secondary actor	Application system, Buyer, Seller
Brief description	This use case allows the Urban Service Manager to delete/Deactivate a buyer, seller login account to the application system.
Pre-condition	<ol style="list-style-type: none"> 1. The user is logged into the application system. 2. The application system displays the home page with all services.
Normal flows	<ol style="list-style-type: none"> 1. The urban service manager clicks on “Manage Buyers Information” button 2. The system returns the “List of buyers” page. 3. The urban service manager clicks on the “Enabled” column button of the record he/she wants to disable. 4. The system returns a conformation message “The buyer has been disabled” 5. The system extracts the corresponding record and delete the associated record in the account data table. 6. In the case, the urban service manager selects “Manage Sellers Information”, then the system returns the similar information as at step 2.
Alternate flows	<ol style="list-style-type: none"> 1. The system displays an error message “Record not found” 2. If the urban service manager clicks on “Enabled” button, the system displays a message “The buyer has been enabled”.
Post-condition	If the use case was successful, the urban service manager sends an email warning the user that his/her account has been deleted/deactivated by providing the reason.

Table 11. Add geometer.

Use case Name	Add Geometer Account
Primary actor	Cadastral Service Manager
Secondary actor	Application system
Brief description	This use case allows the cadastral service manager to add a Geometer login account to the application system.
Pre-condition	<ol style="list-style-type: none"> 1. The Cadastral Service Manager is logged into the application system. 2. The application system displays the home page with all services.
Normal flows	<ol style="list-style-type: none"> 1. The Cadastral Service Manager select “Geometer Information Management”. 2. The system returns the “Geometer Management Services” 3. The Cadastral Service Manager selects “Add Geometer Account”, then the system returns the following information: <ul style="list-style-type: none"> ✚ Cadastral service manager ✚ Manager code ✚ Service name ✚ first name ✚ last name ✚ Nationality ✚ Nickname ✚ Address 1 & 2 ✚ email ✚ user name ✚ password ✚ Role ID ✚ Postal code ✚ Photo 4. The Cadastral Service Manager fills the information, then clicks on “Save” 5. The system validates the information and returns a successful account creation or a failure message. 6. In case of successful creation, the Geometer receives an automate email for successful account creation that is strongly requesting him to update his account password between 24 hours. 7. The system extracts the data entered by the Cadastral Service Manager and add new record to the account data table.
Alternate flows	<ol style="list-style-type: none"> 1. The Cadastral Service Manager enters username that already exists. 2. The system displays an error message “Username already created, please try another one” or “Unknown response from server”. 3. The Cadastral Service Manager can either try again or cancel the operation by clicking on “Cancel” button 4. If the Cadastral Service Manager clicks on “Cancel” button, the system returns back the previous page.
Post-condition	The Geometer account is created
Exceptions	Username already created

Table 12. User login.

Use case Name	User login
Primary actor	System administrator, Urban/Cadastral managers, buyers, sellers, Notary, Geometer, Cadastral/Urban agent
Secondary actor	Application system
Brief description	This use case allows user to log onto the application system and then performs the desired operation.
Pre-condition	The user has a valid account (username and password) and would like to log into the system.
Normal flows	<ol style="list-style-type: none"> 1. The user launches the application system. 2. The system returns back the login page. 3. The user enters username and password then click on “Login” button that sends the credential to the application system. 4. The application system extracts the account type, username and password, then check for validity. 5. The system displays the home menu page with all services, access to services is based on the user level of authority.
Alternate flows	<ol style="list-style-type: none"> 1. The user enters wrong username and/or password. 2. The user account is disabled 3. The system displays an error message “Bad credentials” or “user account disabled” 4. The user can try again, or abort the operation by clicking on the exit button at which point the application system is closed.
Post-condition	If the use case was successful, the user is logged into the system. Otherwise, the user does not get access to the application system’s functionalities.
Exceptions	<ul style="list-style-type: none"> • Bad credentials • User account disabled

Table 13. Add land.

Use case Name	Add Land
Primary actor	Seller
Secondary actor	Application system, urban service manager, Urban Agent, domain administrator
Brief description	This use case allows a land seller to add new land information to the application system for selling purpose.
Pre-condition	<ol style="list-style-type: none"> 1. The seller is logged into the application system. 2. The application system displays the home page with all services.
Normal flows	<ol style="list-style-type: none"> 1. The Seller selects “Documents” in the tools. 2. The system returns the “Documents” sub-menu which contains the following items: <ol style="list-style-type: none"> a. Land drawing b. My lands approved c. My lands registered d. Register my land

Continued

3. The Seller selects “Register my land”, then the system returns the “registration application form” containing the following information:

- ✚ Date of birth
- ✚ Birth-place
- ✚ Common -law partner
- ✚ Occupation
- ✚ ID card no
- ✚ Land title no
- ✚ Lot no
- ✚ ILot no
- ✚ Parcel no
- ✚ Section
- ✚ Price
- ✚ Customary ownership certificate
- ✚ Country
- ✚ City
- ✚ Longitude
- ✚ Latitude
- ✚ Property type
- ✚ Category
- ✚ Square area
- ✚ Location description
- ✚ Land images (Main image, extra image 1, 2, 3, 4, 5, 6, 7, 8, 9)
- ✚ Seller ID

4. The seller then clicks on “Save” button.

5. The system extracts the data entered by the seller and add new record to the land data table. The land information is displayed in a data table that has a column named “approval”. The column value maybe (**pending...**, **approved**, **Not approved**)

6. The Urban Manager checks manually the conformity of the land documents with the domain administrator.

7. In case, the documents provided by the seller are valid, the urban service manager sends a successful land registration email to the seller and changes the approval column value from “**pending**” to “**approved**” otherwise, the urban service manager sends a land registration failure email to the seller, requesting him to update land ownership information (proof, documents, etc...) in between a couple of days otherwise his land registration information will be deleted from the system.

Alternate flows

8. The seller enters information about a land that is already registered.

9. The system displays an error message “Land Record already exists, please try another one”.

10. The seller can either try again or cancel the operation by clicking on “Go Back” button, in case, the seller clicks on “Go Back” button, the system returns back the previous.

Post-condition The land registration is approved

Exceptions Land already registered

Table 14. Search land.

Use case Name	Search land
Primary actor	Buyer, people visiting the platform
Secondary actor	The application system
Brief description	This use case allows a land buyer, or people visiting the platform to search for a land on sale in a particular area.
Pre-condition	<ol style="list-style-type: none"> 1. The user launches the application system 2. The application system displays the home page with a search form and other services.
Normal flows	<ol style="list-style-type: none"> 1. The user enters the land characteristics he/she is looking for. Characteristic such as: <ul style="list-style-type: none"> ✚ Land Type (Purchase, Rent) ✚ Country ✚ City ✚ Min & Max price ✚ Min & Max surface 2. The user then clicks on “Confirm” button. 3. The application system displays a list of lands that match the given characteristics. 4. The user selects the land that he/she likes by clicking on “see details” to have more details of the selected land. 5. The application system displays all the details of the selected land.
Alternate flow	<ol style="list-style-type: none"> 1. The buyer enters land characteristics that do not exist in the application system. 2. The application displays “No match found...please try again”
Post-condition	If the use case was successful, the application system displays a list of lands that matched the research characteristics, otherwise the application system displays a warning message.
Exception	Land not found (“No match found...please try again”)




Table 15. Land document verification.

Use case Name	Land document verification
Primary actor	Buyer
Secondary actor	The application system, the seller, the urban service manager, the urban agent.
Brief description	This use case allows to submit land document verification to the urban service manager.
Pre-condition	<ol style="list-style-type: none"> 1. The buyer is logged into the application system. 2. The buyer has selected the land he/she likes. 3. The application system displays the land information as well as the owner information.
Normal flows	<ol style="list-style-type: none"> 1. The buyer makes a phone call to the land seller to get an “agreement code” 2. The seller selects “Inbox”, then the system returns the following: <ul style="list-style-type: none"> ✚ Send agreement code ✚ Urban service response 3. The seller clicks on “Send agreement code” 4. The system displays a form requesting the following inputs:

Continued

- Agreement code
 - Select land document ID
 - From
 - To
 - Object
 - Message
5. The seller clicks on “Generate code” button.
 6. The system generates a unique code, the seller pastes that code in the “agreement code” input and complete other information, then click on “Send” button.
 7. The system extracts the information, and records it in the “seller agreement message” data table and send it the buyer.
 8. The buyer selects “Land Info”, the system returns the following:
 - ✚ Documents conformity
 - ✚ Verification Agreement
 9. The buyer clicks on “Verification Agreement”
 10. The system displays a list of all agreement messages, each agreement message containing:
 - Reference
 - Sending date
 - Expiration date (agreement code expiration date)
 - Object
 - Message (if any)
 - Agreement code
 11. The buyer copies the agreement code, then click on the code.
 12. The system displays the corresponding land details.
 13. The buyer clicks on “Document verification” button.
 14. The system displays the “Land document integrity verification” form.
 15. The buyer pastes the “agreement code” to the “Verification agreement code” input, the buyer may provide additional information in the message input area, then clicks on “Send” button
 16. The system extracts the user inputs along with the land document information as well as the seller information and add a new record in the application system.
 17. The urban service manager clicks on “Land for sale” button
 18. The system returns the “Land documents management” page
 19. The urban service manager clicks on “Transmission to agents” buttons.
 20. The system returns the “Land documents transmission” page.
 21. The urban service manager enters the following document transmission details:
 - ✚ Document ID
 - ✚ Urban agent
 - ✚ Deadline
 22. The urban service manager clicks on “Assign” button.
 23. The system extracts the entered details and add the new transmission detail record to the document transmission data table, and displays the details in a table.
 24. The urban agent clicks on “case processing”
 25. The system displays the following:
-

Continued

-  Document List
-  Verification Office
-  Profile Control

26. The urban agent selects “Document List” to displays the list of land documents verification assigned to him/her.
27. The system displays a table of documents.
28. The urban agent copies the “transmission code”, then clicks on “Verification Office” the system has displayed at step 26.
29. The system displays the following:
30. New land conformity check
 - a. Edit land conformity check
 - b. Delete land conformity check
 - c. New profile control
 - d. Edit profile control
 - e. Delete profile control
31. The urban agent clicks on “New land conformity check”
32. The system displays the check list that allows to the urban agent to verify land documents and submit it to the urban service manager”.
33. The urban service manager receives the check list from the urban agent”, then validate the conformity of the land documents.
34. The system extracts all lands with documents that are conformed then makes them available for buying.

Alternate flow The system displays an error message “The document doesn’t exist or has been verified”.

Post-condition The land documents are conformed.

6.4. Database Encryption/Decryption

The land information system has many sensitive data. Almost all tables have data that are classified. There are two options to assure the protection of those data: secure the data at the attribute level or secure the whole tablespace as all tables are impacted. Our recommendation and the implemented solution is the encryption the database tablespace.

The database tablespace is a file that contains all tables of the MySQL database instance. Encrypting the tablespace file with the AES algorithm, allows the database access rights to be achieved only through the developed Land application using its stored private key. Attackers that may want to skip the strong checking access through the application level using username/password and a biometric data (in this case the facial recognition), and interact directly with the database instance address, will be challenged to provide the shared secret (meaning the private key). The attackers will then fail to access the plain data in the database. To achieve the Database encryption/decryption the “alliance key manager” was used and it is based on KMIP 1.1 standards. The master key generates with the

certificate authority key and the primary key (that acts as the private key), is used to encrypt the database. The database decryption is realized using the primary key.

7. Build and Development

7.1. Application Design

7.1.1. Web Security Configuration

The code below makes sure any request to the application is authenticated. Its purpose is to protect against common attacks.

```
@SuppressWarnings("deprecation")
@Configuration
@EnableWebSecurity
public class WebSecurityConfig extends WebSecurityConfigurerAdapter {

    @Override
    protected void configure(HttpSecurity http) throws Exception {
        http.csrf().csrfTokenRepository(CookieCsrfTokenRepository.withHttpOnlyFalse());
        http.authorizeRequests()
            .antMatchers("/verifyUrbanManager", "/verifyCadastralManager", "/verifyNotary",
                "/verifyGeometer", "/verifyUrbanAgent", "/verifyCadastralAgent").permitAll()
            .antMatchers("/land_for_sales/land_title_production/management/title/**").hasAnyAuthority("Domain Administrator",
                "Urban Manager", "Notary")
            .antMatchers("/land_for_sales/land_ownership_documents_verification/**", "/land_mutation/details/**",
                "/land_for_sales/land_title_production/management/**").hasAnyAuthority("Domain Administrator", "Urban Manager")
            .antMatchers("/domainAdmin/**", "/urban/**", "/cadastral/**", "/notary/**", "/registered_lands/**",
                "/land_revokation_message/**", "/land_for_sales/seller_agreement/**").hasAuthority("Domain Administrator")
            .antMatchers("/urban_manager/**", "/buyer/**", "/seller/**", "/urban_manager/account/**",
                "/urban_agent/**", "/land_for_sales/**", "/land_for_sales/transmit_documents_to_agent/**", "/user_profile_control/**").hasAuthority("Urban Manager")
            .antMatchers("/cadastral_manager/**", "/cadastral_manager/account/**", "/property_category/**",
                "/geometer/**", "/land_drawing/**").hasAuthority("Cadastral Manager")
            .antMatchers("/document_list/**", "/verification_office/**", "/profile_control/**").hasAuthority("Urban Agent")
            .antMatchers("/cadastral_document_list/**", "/cadastral_verification_office/**",
                "/cadastral_profile_control/**").hasAuthority("Cadastral Agent")
            .antMatchers("/drawing/**").hasAuthority("Geometer")
            .antMatchers("/land_conformity_check/**", "/payment/**", "/authorize_payment/**", "/payment_error/**",
                "/cancel/**", "/success/**", "/receipt/**").hasAuthority("Notary")

        .anyRequest().authenticated()
        .and()
        .formLogin()
            .loginPage("/login")
            .usernameParameter("userName")
            .permitAll()
            .and().logout().permitAll()
            .and()
            .rememberMe()
                .key("AbcDefghIjKlmnOpqrs1234567890")
                .tokenValiditySeconds(7 * 24 * 60 * 60);
        http.headers().frameOptions().sameOrigin();
    }
}
```

7.1.2. Load User Details

```
@Override
public UserDetails loadUserByUsername(String username) throws UsernameNotFoundException {
    DomainAdministrator admin = adminRepo.getAdminByUsername(username);
    UrbanServiceManager urban = urbanServiceManagerRepo.getUrbanManagerByUsername(username);
    CadastralServiceManager cadastral = cadastralServiceManagerRepo.getCadastralManagerByUsername(username);
    Notary notary = notaryRepo.getNotaryByUsername(username);
    Geometer geometer = geometerRepo.getGeometerByUsername(username);
    UrbanAgent urbanAgent = urbanRepo.getUrbanAgentByUsername(username);
    CadastralAgent cadastralAgent = cadastralRepo.getCadastralAgentByUsername(username);
    if(admin != null) {
        return new LandUserDetails<DomainAdministrator>(admin);
    } else if(urban != null) {
        return new LandUserDetails<UrbanServiceManager>(urban);
    } else if(cadastral != null) {
        return new LandUserDetails<CadastralServiceManager>(cadastral);
    } else if(notary != null) {
        return new LandUserDetails<Notary>(notary);
    } else if(geometer != null) {
        return new LandUserDetails<Geometer>(geometer);
    } else if(urbanAgent != null) {
        return new LandUserDetails<UrbanAgent>(urbanAgent);
    } else if(cadastralAgent != null) {
        return new LandUserDetails<CadastralAgent>(cadastralAgent);
    }
    throw new UsernameNotFoundException("Could not find user with username: " + username);
}
}
```

```

@Override
public UserDetails loadUserByUsername(String username) throws UsernameNotFoundException {
    Buyer buyer = bRepo.getBuyerByUsername(username);
    Seller seller = sRepo.getSellerByUsername(username);
    if(buyer != null) {
        return new AuthenticatedUserDetails<Buyer>(buyer);
    } else if(seller != null) {
        return new AuthenticatedUserDetails<Seller>(seller);
    }
    throw new UsernameNotFoundException("No user found with the user name " + username);
}

```

The code above allows extracting the username and password entered by the user and verify the user credentials. In case, the account is valid the user get access to the system only if the user is not disabled by the manager. Otherwise the system displays an error message.

7.1.3. Common Queries

```

public abstract class Queries {

    public static final String getAdminUsermane = "SELECT admin FROM DomainAdministrator admin WHERE admin.userName = :user_name";
    public static final String getAdminUpdatedUnabledStatus = "UPDATE DomainAdministrator admin SET admin.enabled = ?2 WHERE admin.id = ?1";
    public static final String adminFindAll = "SELECT admin FROM DomainAdministrator admin WHERE CONCAT(admin.id, ' ', admin.email, ' ', admin.firstName, ' ', admin.lastName) LIKE ?1";

    public static final String getUrbanServiceManagerUsermane = "SELECT urbanServiceManager FROM UrbanServiceManager urbanServiceManager WHERE urbanServiceManager.userName = :user_name";
    public static final String getUrbanServiceManagerUpdatedEnabledStatus = "UPDATE UrbanServiceManager urbanServiceManager SET urbanServiceManager.enabled = ?2 WHERE urbanServiceManager.id = ?1";
    public static final String urbanServiceManagerFindAll = "SELECT urbanServiceManager FROM UrbanServiceManager urbanServiceManager WHERE CONCAT(urbanServiceManager.id, ' ', urbanServiceManager.email, ' ', urbanServiceManager.firstName, ' ', urbanServiceManager.lastName) LIKE ?1";
    public static final String getUrbanServiceManagerEnabledStatusWithVerificationCode = "UPDATE UrbanServiceManager urbanServiceManager SET urbanServiceManager.enabled = true WHERE urbanServiceManager.verificationCode = ?1";
    public static final String getUrbanServiceManagerByVerificationCode = "SELECT urbanServiceManager FROM UrbanServiceManager urbanServiceManager WHERE urbanServiceManager.verificationCode = ?1";
    public static final String getUrbanServiceManagerByDomainAdministrator = "SELECT urbanServiceManager FROM UrbanServiceManager urbanServiceManager WHERE urbanServiceManager.domainAdministratorId = ?1";

    public static final String getCadastralServiceManagerUsermane = "SELECT cadastralServiceManager FROM CadastralServiceManager cadastralServiceManager WHERE cadastralServiceManager.userName = :user_name";
    public static final String getCadastralServiceManagerUpdatedEnabledStatus = "UPDATE CadastralServiceManager cadastralServiceManager SET cadastralServiceManager.enabled = ?2 WHERE cadastralServiceManager.id = ?1";
    public static final String cadastralServiceManagerFindAll = "SELECT cadastralServiceManager FROM CadastralServiceManager cadastralServiceManager WHERE CONCAT(cadastralServiceManager.id, ' ', cadastralServiceManager.email, ' ', cadastralServiceManager.firstName, ' ', cadastralServiceManager.lastName) LIKE ?1";
    public static final String getCadastralServiceManagerEnabledStatusWithVerificationCode = "UPDATE CadastralServiceManager cadastralServiceManager SET cadastralServiceManager.enabled = true WHERE cadastralServiceManager.verificationCode = ?1";
    public static final String getCadastralServiceManagerByVerificationCode = "SELECT cadastralServiceManager FROM CadastralServiceManager cadastralServiceManager WHERE cadastralServiceManager.verificationCode = ?1";

    public static final String getNotaryUsermane = "SELECT notary FROM Notary notary WHERE notary.userName = :user_name";
    public static final String getNotaryUpdatedEnabledStatus = "UPDATE Notary notary SET notary.enabled = ?2 WHERE notary.id = ?1";
    public static final String getNotaryEnabledStatusWithVerificationCode = "UPDATE Notary notary SET notary.enabled = true, notary.verificationCode = null WHERE notary.id = ?1";
    public static final String notaryFindAll = "SELECT notary FROM Notary notary WHERE CONCAT(notary.id, ' ', notary.email, ' ', notary.firstName, ' ', notary.lastName) LIKE ?1";
    public static final String getNotaryByVerificationCode = "SELECT notary FROM Notary notary WHERE notary.verificationCode = ?1";

    public static final String getBuyerUsermane = "SELECT buyer FROM Buyer buyer WHERE buyer.userName = :user_name";
    public static final String getBuyerUpdatedEnabledStatus = "UPDATE Buyer buyer SET buyer.enabled = ?2 WHERE buyer.id = ?1";
    public static final String buyerFindAll = "SELECT buyer FROM Buyer buyer WHERE CONCAT(buyer.email, ' ', buyer.firstName, ' ', buyer.lastName, ' ', buyer.nickname, ' ', buyer.userName) LIKE ?1";
    public static final String getBuyerByFirstNameAndLastName = "SELECT b FROM Buyer b WHERE b.firstName = ?1 AND b.lastName = ?2";
    public static final String findApprovedBuyers = "SELECT buyer FROM Buyer buyer WHERE buyer.enabled = true";

    public static final String getSellerUsermane = "SELECT seller FROM Seller seller WHERE seller.userName = :user_name";
    public static final String getSellerUpdatedEnabledStatus = "UPDATE Seller seller SET seller.enabled = ?2 WHERE seller.id = ?1";
    public static final String sellerFindAll = "SELECT seller FROM Seller seller WHERE CONCAT(seller.email, ' ', seller.firstName, ' ', seller.lastName, ' ', seller.nickname, ' ', seller.userName) LIKE ?1";
    public static final String getSellerByFirstNameAndLastName = "SELECT s FROM Seller s WHERE s.firstName = ?1 AND s.lastName = ?2";
}

```

The code above allows sending querying the database system. It contains reused query when needed.

7.1.4. Database Connectivity

```

spring.datasource.url=jdbc:mysql://localhost:3306/landacquisitiondb
spring.datasource.username=root
spring.datasource.password=traore

spring.jpa.hibernate.ddl-auto=update
spring.jpa.show-sql=true
spring.jpa.properties.hibernate.format_sql=true
spring.jpa.properties.hibernate.dialect-org.hibernate.dialect.MySQL8Dialect
spring.output.ansi.enabled=always

```

The code above allows connection to the database system that allow performing the CRUD operations.

7.1.5. Add New Land Registration

```
@PostMapping("register_land/save")
public String createNewLand(Land land, RedirectAttributes redirectAttributes,
    @RequestParam(value = "image_0", required = true) MultipartFile ownershipCertificate, @RequestParam(value = "image_1", required = true) MultipartFile mainImage,
    @RequestParam(value = "image_2", required = true) MultipartFile extraImage1, @RequestParam(value = "image_3", required = true) MultipartFile extraImage2,
    @RequestParam(value = "image_4", required = true) MultipartFile extraImage3, @RequestParam(value = "image_5", required = true) MultipartFile extraImage4,
    @RequestParam(value = "image_6", required = true) MultipartFile extraImage5, @RequestParam(value = "image_7", required = true) MultipartFile extraImage6,
    @RequestParam(value = "image_8", required = true) MultipartFile extraImage7, @RequestParam(value = "image_9", required = true) MultipartFile extraImage8,
    @RequestParam(value = "image_10", required = true) MultipartFile extraImage9, @Param("longitude") String longitude, @Param("latitude") String latitude,
    @Param("section") String section, @Param("price") String price, @Param("city") String city, @Param("squareArea") String squareArea, HttpServletRequest request) {
    try {
        Object principal = request.getUserPrincipal();
        if(ownershipCertificate.isEmpty() && !mainImage.isEmpty()) {
            land.setLand_metrics(new LandMetrics(new Point(Double.valueOf(longitude), Double.valueOf(latitude)), price, city, section, squareArea));
            land.setCustomerOwnershipCertificate(Base64.getEncoder().encodeToString(ownershipCertificate.getBytes()));
            land.setMainImage(Base64.getEncoder().encodeToString(mainImage.getBytes()));
        }

        if(principal instanceof UsernamePasswordAuthenticationToken || principal instanceof RememberMeAuthenticationToken) {
            AuthenticatedUserDetails<Seller> userDetails = getCustomerDetailsObject(principal);
            Seller seller = (Seller) userDetails.getUser();
            land.setSeller(seller);
        }

        if(extraImage1.isEmpty())
            land.setImageExtra1(null);
        else
            land.setImageExtra1(Base64.getEncoder().encodeToString(extraImage1.getBytes()));

        if(extraImage2.isEmpty())
            land.setImageExtra2(null);
        else
            land.setImageExtra2(Base64.getEncoder().encodeToString(extraImage2.getBytes()));

        if(extraImage3.isEmpty())
            land.setImageExtra3(null);
        else
            land.setImageExtra3(Base64.getEncoder().encodeToString(extraImage3.getBytes()));
```

```
        if(extraImage4.isEmpty())
            land.setImageExtra4(null);
        else
            land.setImageExtra4(Base64.getEncoder().encodeToString(extraImage4.getBytes()));

        if(extraImage5.isEmpty())
            land.setImageExtra5(null);
        else
            land.setImageExtra5(Base64.getEncoder().encodeToString(extraImage5.getBytes()));

        if(extraImage6.isEmpty())
            land.setImageExtra6(null);
        else
            land.setImageExtra6(Base64.getEncoder().encodeToString(extraImage6.getBytes()));

        if(extraImage7.isEmpty())
            land.setImageExtra7(null);
        else
            land.setImageExtra7(Base64.getEncoder().encodeToString(extraImage7.getBytes()));

        if(extraImage8.isEmpty())
            land.setImageExtra8(null);
        else
            land.setImageExtra8(Base64.getEncoder().encodeToString(extraImage8.getBytes()));
        if(extraImage9.isEmpty())
            land.setImageExtra9(null);
        else
            land.setImageExtra9(Base64.getEncoder().encodeToString(extraImage9.getBytes()));

        service.register(land);
    }
    redirectAttributes.addFlashAttribute("message", "We've successfully received your land registration request!");
    return "redirect:/";
} catch (FileNotFoundException | IOException e) {
    redirectAttributes.addFlashAttribute("message", e.getMessage());
    return "redirect:/register_land";
}
```

The *getCustomerDetailsObjects* is the method that is going to collect the currently logged in user details. The object parameter contains the actually user details

```
@SuppressWarnings("unchecked")
private AuthenticatedUserDetails<Seller> getCustomerDetailsObject(Object principal) {
    AuthenticatedUserDetails<Seller> userDetails = null;
    if(principal instanceof UsernamePasswordAuthenticationToken) {
        UsernamePasswordAuthenticationToken token = (UsernamePasswordAuthenticationToken) principal;
        userDetails = (AuthenticatedUserDetails<Seller>) token.getPrincipal();
    } else if (principal instanceof RememberMeAuthenticationToken) {
        RememberMeAuthenticationToken token = (RememberMeAuthenticationToken) principal;
        userDetails = (AuthenticatedUserDetails<Seller>) token.getPrincipal();
    }
    return userDetails;
}
```

7.1.6. Search land

```
@GetMapping("/search/page/{id}/details")
public String viewLandDetails(@PathVariable("id") Integer id, Model model) {
    try {
        Land land = service.get(id);
        model.addAttribute("land", land);
        model.addAttribute("pageTitle", land.getDocumentId());
        return "land_details";
    } catch (LandNotFoundException e) {
        return "error/404";
    }
}
```

This code above allows users to search for a particular land detail.

7.1.7. Land Document Integrity Verification Submission

```
@PostMapping("/documents_integrity_verification/save")
public String saveLandDocumentsIntegrityVerification(@Param("agreementCode") String agreementCode, @Param("landId") Integer landId, @Param("message") String message,
    RedirectAttributes redirectAttributes, HttpServletRequest request) {
    try {
        Land land = service.get(landId);
        Buyer buyer = getAuthenticatedBuyer(request);
        SellerAgreementMessage sellerAgreementCode = aService.findAgreementByCode(agreementCode);
        UrbanLandVerification verificationRequest = new UrbanLandVerification();
        service.saveLandDocumentsIntegrityVerification(land, verificationRequest, sellerAgreementCode, buyer, message);
    } catch (LandNotFoundException e) {
        redirectAttributes.addFlashAttribute("message", e.getMessage());
    } catch (SellerAgreementMessageNotFoundException e) {
        redirectAttributes.addFlashAttribute("message", e.getMessage());
    }
    return "redirect:/" + landId + "/details";
}
```

This code above allows a buyer to submit a land document integrity verification request the urban service manager.

7.1.8. Land Drawing Request Submission

```
@GetMapping("/land_for_sales/issue_land_drawing/request/{id}")
public String viewRequestDetails(@PathVariable("id") Integer id, RedirectAttributes redirectAttributes, Model model) {
    try {
        SellerLandDrawingRequest sRequest = service.getRequest(id);
        model.addAttribute("sRequest", sRequest);
        return "urban/land/request_details";
    } catch (RequestNotFoundException e) {
        redirectAttributes.addFlashAttribute("message", e.getMessage());
        return defaultRedirectRequestListURL;
    }
}
```

This code above allows the urban service manager to submit a land drawing request to the cadastral service manager.

8. Tests and Results

8.1. Integrity, Confidentiality and Reliability

For tests purpose (**Table 16**), 10 client machines (to simulate buyer, seller, domain administrator, urban agent/administrator, notary, and attacker) were used. One server machine that acted as web, application and database server was used. ZeroSSL was used to get free SSL certificates for the client. All machines are on Windows 10 operating system.

The confidentiality tests were done at the application level all data was signed by a TSL certificate (Standard TSL 1.3) and transportation was secure.

Table 16. Integrity test cases.

Test case	Number of tests	Frequency	Success	Solution efficiency (%)
Attacker login to Web application using username/password and forging facial image through web cam	25	30 minutes	No	100%
Attacker login directly to Database application using database username/password and a primary key for decryption	25	30 minutes	No	100%

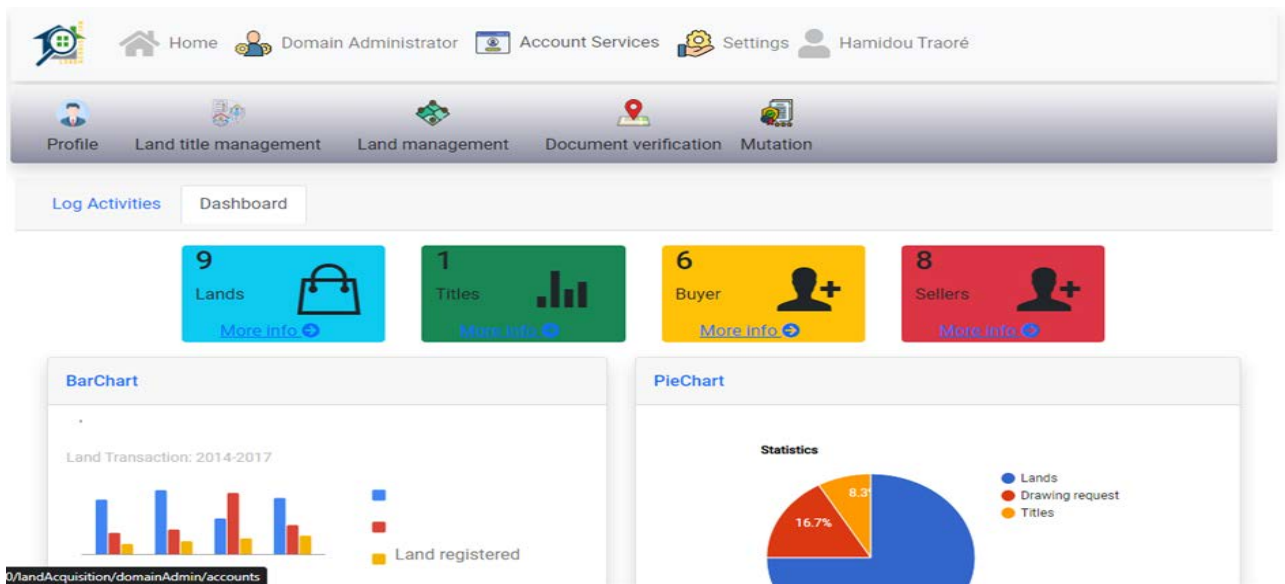


Figure 5. Domain administrator home page.

Robustness testing have not been conducted as only one server was used for test purpose. However in the production environment, we recommend a cloud deployment approach as the resources are available for a target budget. Secondary server will then be connected to the information system and takes the relay in case of any Denial of Service attack.

8.2. Application User Interface Design

For tests purpose, 10 client machines (to simulate buyer, seller, domain administrator, urban agent/administrator,

8.2.1. Domain Administrator Home Page

This page (**Figure 5**) is the domain administrator home page after a successful login.

8.2.2. Urban Service Manager Home Page

This page (**Figure 6**) is the service manager home page after a successful login.

8.2.3. Cadastral Service Manager Home Page

After the user successfully logged into the system, the system displays the home page, **Figure 7**. The user can perform the desired operation based on his/her level

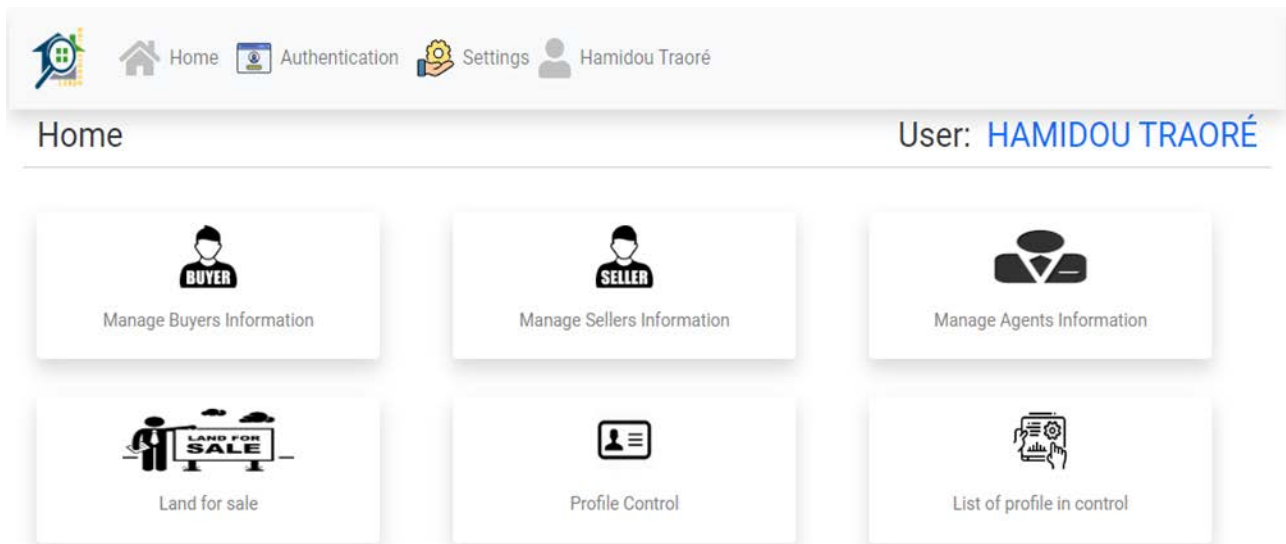


Figure 6. Urban service manager home page.

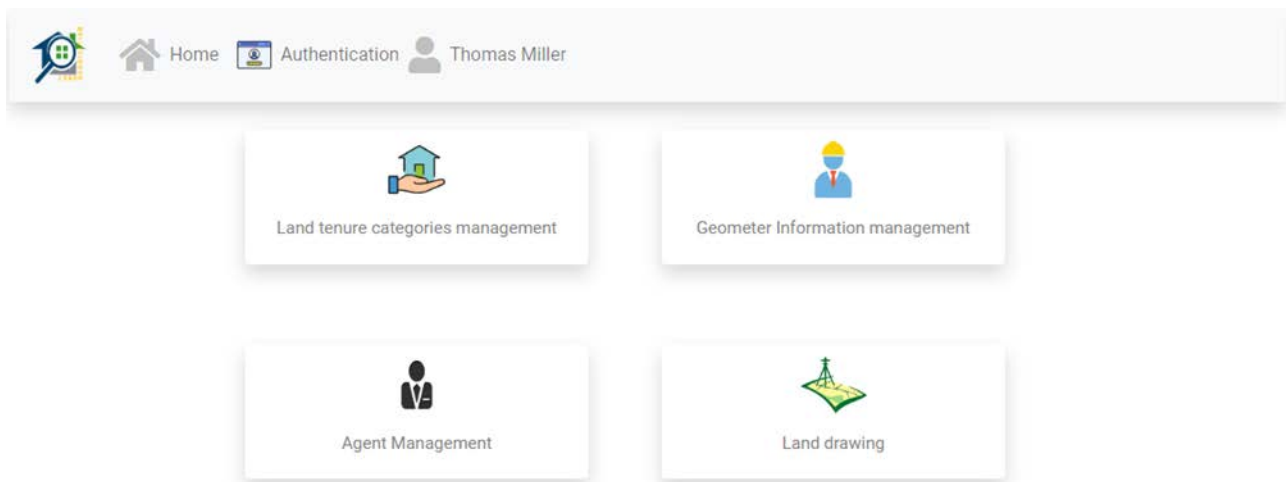


Figure 7. Cadastral service manager home page.

of authorization.

8.2.4. Add New Domain Administrator Page

Figure 8 is related to the creation of domain administrator interface.

8.2.5. Urban/Cadastral/Notary Service Management Page

This page (**Figure 9**) is strictly reserved to the domain administrator to be able to manage (add, update, deactivate) the urban service manager, the cadastral service manager and the notaries information.

8.2.6. Domain Administrator Setting Page

This page (**Figure 10**) is strictly reserved to the domain administrator to be able to manage roles, general setting, countries, districts, regions, departments, mail server, mail templates of the application system.

Manage Domain Administrator | [Create New Administrator](#)

Code	E-mail
<input type="text"/>	<input type="text"/>
First name	Last name
<input type="text"/>	<input type="text"/>
Nickname	User name
<input type="text"/>	<input type="text"/>
Password	Phone number
<input type="text"/>	<input type="text"/>
Nationality	Roles <input type="checkbox"/> Domain Administrator - Manage everything
<input type="text"/>	
Enabled <input checked="" type="checkbox"/>	Photos
	<input type="button" value="Choisir un fichier"/> Aucun fichier choisi
<input type="button" value="Save"/>	<input type="button" value="Cancel"/>

Figure 8. Add new domain administrator home page.

Welcome *Hamidou Traoré*

Urban Services

Cadastral Services

Notary Services

Figure 9. Urban/Cadastral/Notary service management home page.

8.2.7. Clients Home page

The page (Figure 11) is the home page for clients (buyers/seller/people visiting the platform). It allows clients to search for a land in a particular area.

8.2.8. Land Research Result

The page (Figure 12) is displayed after the user has entered land characteristics and clicks on “Confirm” button of the search form.

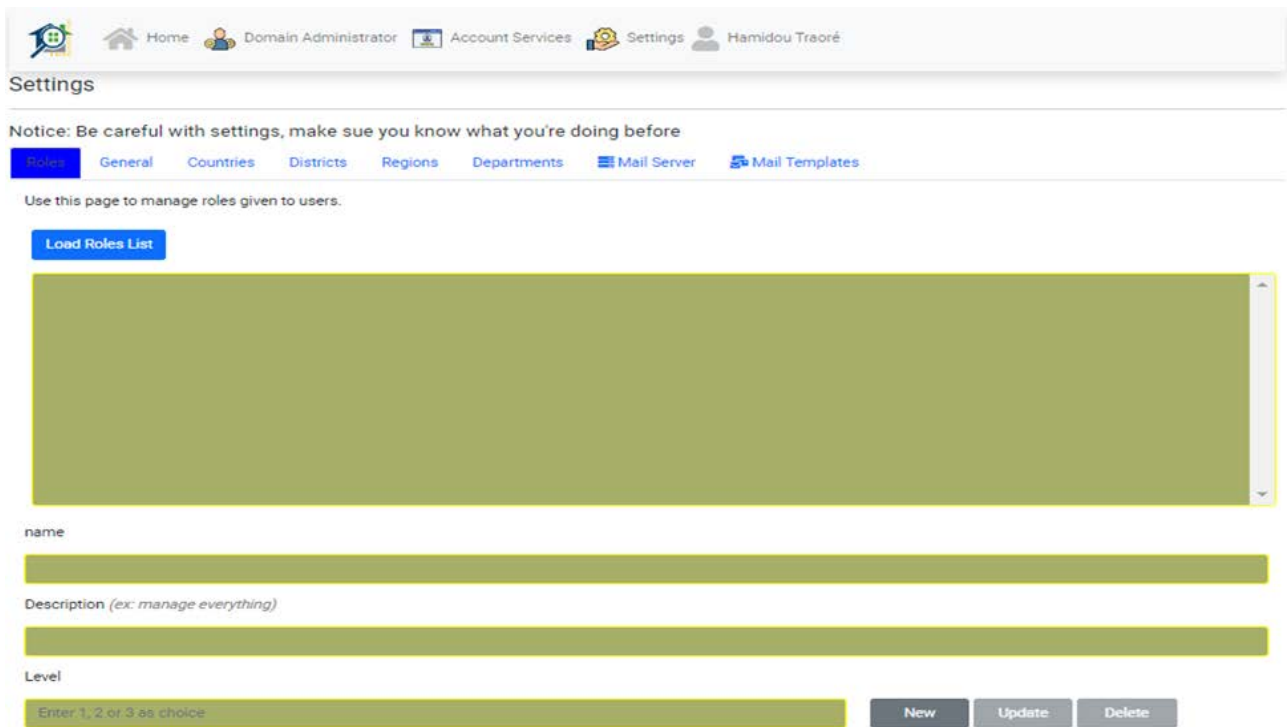


Figure 10. Domain administrator setting page.

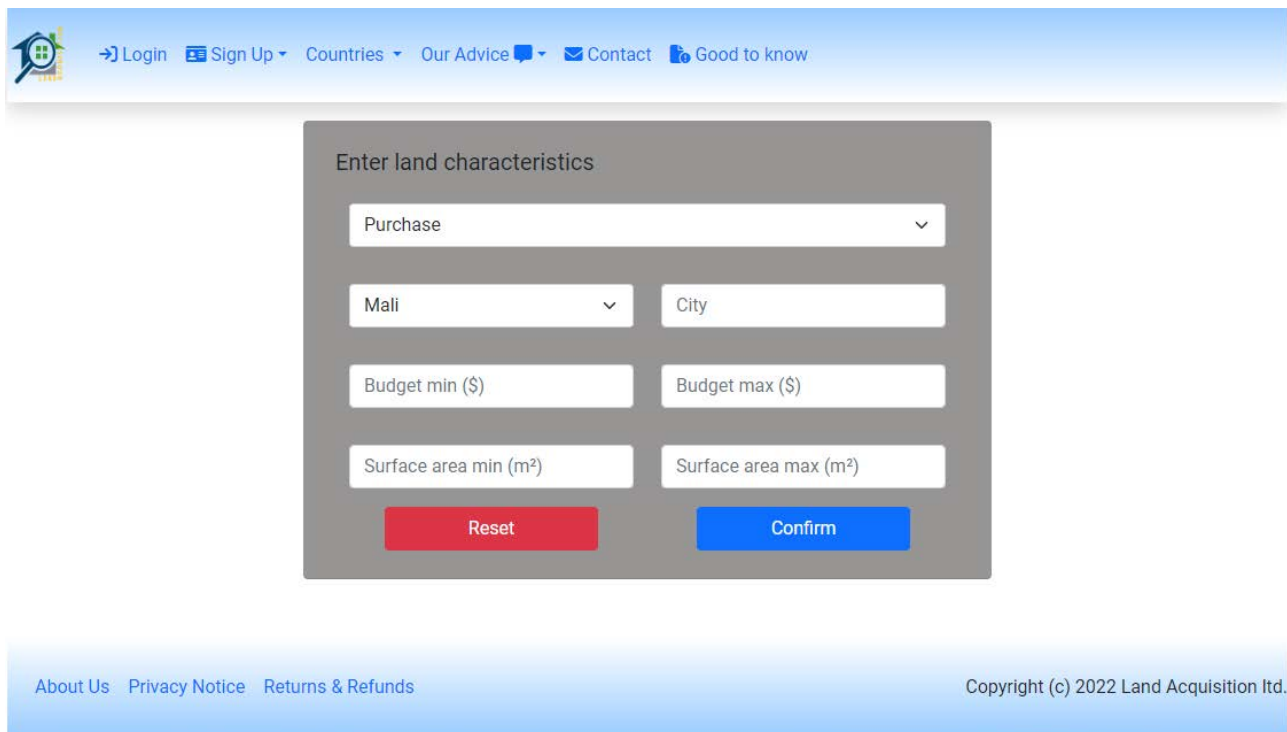


Figure 11. Client home page.

8.2.9. Sign-Up Page (Buyer/Seller)

The page (Figure 13) allows a new seller to sign up. The sign-up pages of buyer are similar to those of the seller.

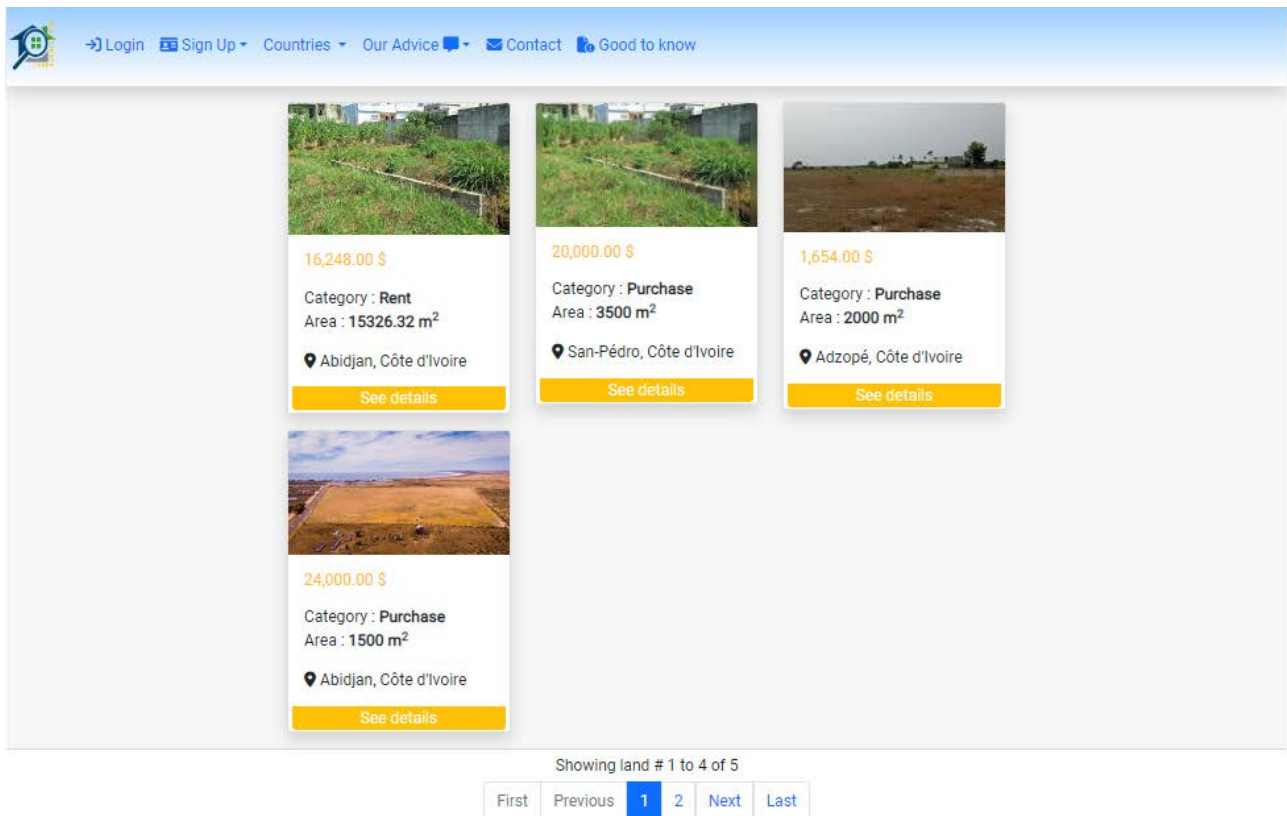


Figure 12. Land research result page.

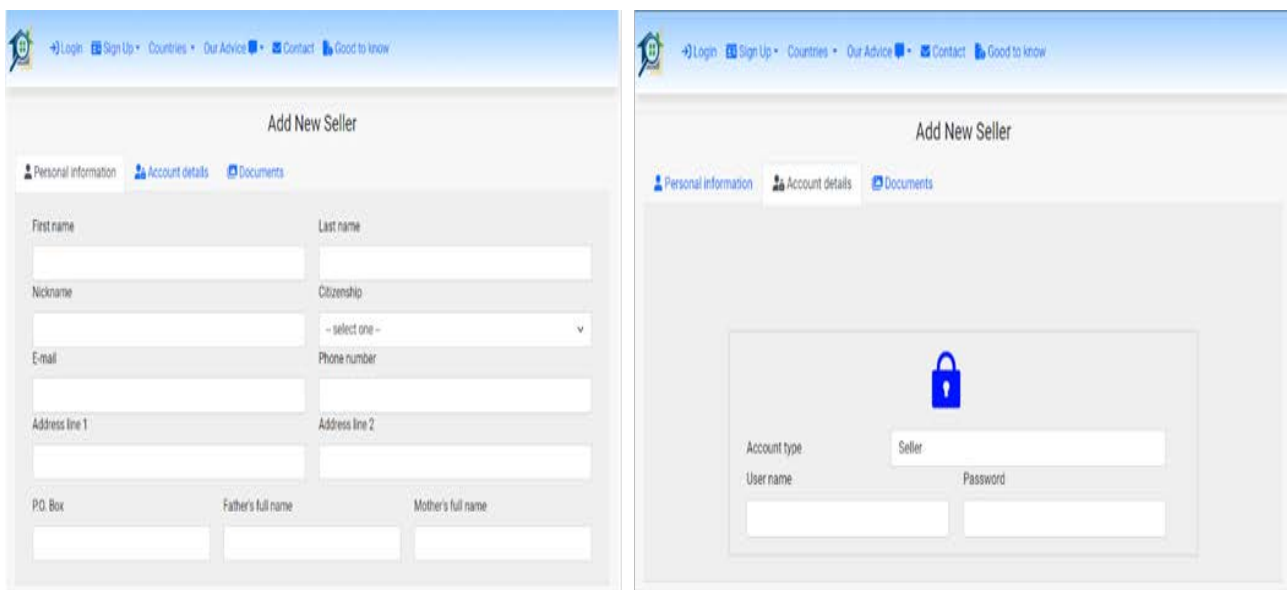


Figure 13. Sign-up page.

9. Conclusions

In this paper, a generic and reliable land acquisition protocol has been implemented. This first implementation version uses the defined land acquisition protocol ACLAP as the sustainable orchestration logic. The infrastructure architec-

ture has 2 main domains: client and server. The client domain is the set of machine belonging to the user of the information system as the buyer, seller, domain administrator, urban agent, urban administrator, geometer, and notary. To ensure the integrity of data, client get access rights to the information system after a successful checking of the username/password and a biometric validation. The server domain is composed of web server, application and database server. The web server contains web pages, resources files, CSS files, war deployed files, and library files. The application contains database access classes, main classes, entity management classes, and the MC classes as views and listeners. The database keeps all required data of the application. The transport of data between machine client and server is realized using the TLS 1.3 protocol encapsulated in the https protocol.

The results show that the first version of the ACLAP implementation is efficient in term of integrity and confidentiality. The reliability was not tested but the proposed solution is robust as we recommend to use a cloud environment where server instances can be made redundant.

The next paper on this topic will show the relational data model and the business intelligence model that can be used to exploit the data analytics.

Conflicts of Interest

The authors declare no conflicts of interest regarding the publication of this paper.

References

- [1] Linwa, A.C.B. and Lebi, J.-M.D. (2022) A Generic and Reliable Land Acquisition Protocol on Software Architecture for Sub-Saharan Africa Countries. *International Technologist*, **18**, 1-13.
- [2] Palmer, D., Fricska, S. and Wehrmann, B. (2009) Pour une Meilleure Gouvernance des Terres. FAO, 74.
- [3] Enemark, S. (2009) Land Administration Systems-Managing Rights, Restrictions and Responsibilities in Land. Map World Forum, Hyderabad, 10-13 February 2009, 9.
- [4] PNUD (2012) Guide pratique pour la prévention et la gestion des conflits liés à la terre et aux ressources naturelles: Terre et Conflits. 96.
- [5] Williamson, I.P. and Wallace, J. (2007) New Roles of Land Administration. *Proceedings, International Workshop on Good Land Administration—Its Role in Economic Development*, Ulaanbaatar, Mongolia, <http://hdl.handle.net/11343/34936>
- [6] WOCAT (2011) La pratique de la gestion durable des terres. TerrAfrica, Pages 243, La pratique de la gestion durable des terres (fao.org).
- [7] De Zeeuw, K., Benn, T., Unger, E.-M., Bennett, R. and Teo, C.H. (2020) United Nations-Framework for Effective Land Administration (FELA). *2020 Annual World Bank Conference on Land and Poverty*, Washington DC, 16-20 March 2020, 30.
- [8] Hurni, H. (1997) Concepts of sustainable land management. *ITC Journal*, **34**, 210-215.
- [9] Traoré, H. (2023) A Generic and Reliable Land Acquisition Application for Sub-Saharan Countries. Capstone Thesis, International University of Grand-Bassam,

Grand-Bassam.

- [10] Liang, Y.S., Sun, W.B., Diao, H.T. and Li, Y. (2010) The Design and Development of the Land Management System in Dingzhuang Town Based on Spatial Data. *4th Conference on Computer and Computing Technologies in Agriculture (CCTA)*, Nanchang, October 2010, 57-65. https://doi.org/10.1007/978-3-642-18354-6_9
- [11] De Maria, M. (2019) Understanding Land in the Context of Large-Scale Land Acquisitions: A Brief History of Land in Economics. *Land*, **8**, 15. <https://doi.org/10.3390/land8010015>
- [12] Intosai, W. (2012) Land Use and Land Management Practices in Environmental. INTOSAI Working Group on Environmental Auditing (WGEA), 81.
- [13] Bürgi, M., Hersperger, A.M., Hall, M., Southgate, E.W.B. and Schneeberger, N. (2007) Using the Past to Understand the Present Land Use and Land Cover. In: Kienast, F., Wildi, O. and Ghosh, S., Eds., *A Changing World*, Springer, Dordrecht, 133-144. https://doi.org/10.1007/978-1-4020-4436-6_9
- [14] Ziadat, F. (2017) Land Resource Planning for Sustainable Land Management. FAO: Land and Water Division Working Papers, 70.
- [15] Berisso, Z.A. and de Vries, W.T. (2010) Exploring Characteristics of GIS Adoption Decisions and Type of Induced Changes in Developing Countries: The Case of Ethiopia. *The Electronic Journal of Information Systems in Developing Countries*, **40**, 1-16. <https://doi.org/10.1002/j.1681-4835.2010.tb00284.x>
- [16] Correia, Z.X., Moore, A.B. and Goodwin, D.P. (2021) Accessibility of Land Data and Information Integration in Recently-Independent Countries: Timor-Leste Case Study. *Papers in Regional Science*, **100**, 203-225. <https://doi.org/10.1111/pirs.12560>
- [17] Karikari, I., Stillwell, J. and Carver, S. (2003) Land Administration and GIS: The Case of Ghana. *Progress in Development Studies*, **3**, 223-242. <https://doi.org/10.1191/1464993403ps050ra>
- [18] Biraro, M. (2014) Land Information Updating: Assessment and Options for Rwanda. University of Twente, Enschede.
- [19] Mwangi Gideon, W. (2011) Development of Integrated Land Information Management System in Kenya. Master Sciences Thesis, University of Nairobi, Nairobi.
- [20] Siriba, D.N. and Dalyot, S. (2017) Adoption of Volunteered Geographic Information into the Formal Land Administration System in Kenya. *Land Use Policy*, **63**, 279-287. <https://doi.org/10.1016/j.landusepol.2017.01.036>
- [21] Gluszak, M. (2008) Land Acquisition in Development Projects: Investment Value and Risks. SRE-Discussion Papers 2008/06, Vienna University of Economics and Business, Vienna.
- [22] Yaw Adiaba, S. (2014) A Framework for Land Information Management in Ghana. PhD Thesis, University of Wolverhampton, Wolverhampton. <https://wlv.openrepository.com/bitstream/handle/2436/332138/THESIS0920118doc.pdf?sequence=1&isAllowed=y>
- [23] Davis, F.D. (1989) Perceived Usefulness, Perceived Ease of Use, and User Acceptance of Information Technology. *MIS Quarterly*, **13**, 319-340. <https://doi.org/10.2307/249008>
- [24] Zeng, Z. and Cleon, C.B. (2018) Factors Affecting the Adoption of a Land Information System: An Empirical Analysis in Liberia. *Land Use Policy*, **73**, 353-362. <https://doi.org/10.1016/j.landusepol.2017.12.034>
- [25] Ouattara, A. and Gouvernement de Côte d'Ivoire (2013) Les Règles d'Acquisition des Terrains Urbains.

- [26] Nguiffo, S. and Seigneret, A. (2021) La réforme foncière au Cameroun: une vision cohérente proposée par la société civile. IIED, London.
<https://pubs.iied.org/20126IIED>
- [27] Yang, P. (1er Ministre) (2014) Texte régulation la gestion de la terre.
<https://extwprlegs1.fao.org/docs/pdf/Cmr176509.pdf>
- [28] Direction des Collectivités Locales (2012) Manuel des procédures Foncières des Communautés Rurales. Ministère de l'Aménagement du Territoire et des Collectivités Locales du Sénégal, 81.