



THE UNIVERSITY OF NAIROBI

SCHOOL OF COMPUTING AND INFORMATICS

AGENT BASED INTEGRATION MODEL: A CASE STUDY OF
KRA AND ITS MEDICAL SERVICE PROVIDERS

BY

RAPHAEL MWANIKI NJERU

PROJECT REPORT SUBMITTED IN PARTIAL FULLFILMENT OF THE
DEGREE OF MASTER OF SCIENCE IN COMPUTER SCIENCE.

OCTOBER 2013

DECLARATION

I Mr. Raphael Mwaniki Njeru do hereby declare that this project report is my original work and to the best of my knowledge, it has not been presented to any other institution or examination body.

Signed:.....

Date:.....

Registration Number: P58/70540/2008

This project report has been presented for examination with my approval as the university supervisor.

Signed:.....

Date:.....

Name: DR. ELISHA T. OPIYO OMULO

Lecturer, School of Computing and Informatics

ABSTRACT

Kenya Revenue Authority provides medical care services to all its members of staff and their dependants through accredited medical service providers. These medical service providers use their own patient management systems to manage patient records. A member of KRA staff or beneficiary can visit any of these medical facilities and get treatment, after which a hardcopy invoice is sent to the authority for settlement. Real time invoices are not available and the physical ones could even take a month to get delivered to the Authority. On the other hand, the service providers do not have a proper mechanism of validating the staff or beneficiary before treatment can commence. A prototype has been developed which validates staff members and beneficiary before the start of the treatment commences. This validation is done by the facility admission office using a web interface which the facility users must first login into, after which validation can be done against the staff ID or beneficiary's medical card no. The patient invoices from the health centres are consolidated and relayed to KRA medical system via Multi Agents System approach, which work in the background to inform the central synchronization server of any patient treatment information taking place. Approaches that have been used previously in systems integration include building new Systems, Web services integration using XML, common data storage mechanisms using data ware houses, data replication and by the use of integration by middleware like SQL middleware. The integration approach used in this model revolves around abstraction at the data level with the aid of JADE frame work. The system agents work cooperatively with other agents in their social network that comprises heterogeneous systems comprising of syntactically and semantically differing data sources. The system delivers output by consulting and retrieving patient treatment records at a facility and where necessary updating the historical data warehouse while maintaining the ACID properties.

ACKNOWLEDEMENT

The process of researching and compiling this report entailed consultation with professionals in the wide ICT domain specifically in databases, network infrastructure and data security. On engaging selected domain experts at KRA, I managed to gather vast technological contribution in various capacities and hereby do wish to thank Mr. Kirwa Kemei, Mr. David Muniu and Madam Salome Kanyugo. I also wish to pass my gratitude to my lecturer, DR. Elisha Opiyo for supervising my work and providing the necessary guidance , Mr.Renken Muinde, Postal Corporation of Kenya for his long term advice on FOSS Tools (Free Open Source Software). Finally, I sincerely wish to thank my dear wife Beatrice, son Kristian and my parents Prof & Mrs E.H.N.Njeru for their endless encouragement and moral support.

TABLE OF CONTENTS

DECLARATION	ii
ABSTRACT.....	iii
ACKNOWLEDEMENT	iv
TABLE OF CONTENTS.....	v
DEFINITION OF TERMS	ix
CHAPTER ONE: INTRODUCTION	1
1.0 Background to the problem.....	1
1.1 Problem statement.....	2
1.2 Goal of the Study	2
1.3 Objectives of the study.....	2
1.4 Significance of the study.....	2
1.5 Business Analysis	3
1.6 Assumptions and scope of the research	3
CHAPTER TWO: LITERATURE REVIEW	5
Introduction.....	5
2.1 Related work on Data integration	5
2.11 The use of mobile agents in an itinerary	5
2.12 Ontology in integrating heterogeneous RDBMs.....	6
2.13 Common Data Storage integration.....	6
2.14 Additional layer integration	7
2.15 Service-Oriented Architecture	7
2.16 Build new System to meet requirements.....	8
2.17 The use of commercial tools in data replication	8
2.2 The Agent Architecture.....	9
2.21 JADE Framework (with MAS support).....	9
2.22 Agent Communication	10
2.23 Communication Modes.....	10
2.24 Communication Languages.....	10
2.3 Methodologies for MAS Engineering.....	11
2.31 SADAAM Methodology.....	11
2.32 Gaia Methodology	12
2.33 MaSE Methodology	12
2.34 MESSAGE Methodology	13
2.35 MESSAGE Concepts	13
2.36 Types of Activities.....	14
2.4 ANALYSIS MODEL VIEWS.....	15
2.5 The Analysis Process	17

2.51 Refinement Approach:	17
2.52 Analysis Refinement strategies	18
CHAPTER THREE: METHODOLOGY	19
3.0 Introduction.....	19
3.1 Requirements gathering	19
3.2 Sources of data and its relevancy.....	19
3.3 MESSAGE methodology in System development	19
3.4 System implementation, testing and Validation.....	20
3.5 System Documentation	20
The documents produced at every stage of system development have been bundled into technical and user based documentation together generating this report write up.....	20
CHAPTER FOUR: ANALYSIS, DESIGN AND IMPLEMENTATION	21
4.1 Introduction.....	21
4.11 System Analysis requirements	21
4.12 Coordination model design	21
4.13 Agent Data Communication Design	22
4.14 Activity and Interaction Diagrams	23
4.15 Staff/Patient accessing medical services.....	23
4.16 Treat Patient (Staff/Dependant)	23
4.17 Invoice payment to Facility/Medical Service providers	24
4.2 Analysis using the MESSAGE concept.....	24
4.3 Agent Task and Responsibility Analysis	25
4.4 General class, Interaction and Sequence Diagrams	27
4.41 Global Interaction diagram between Facility Agents and SYNCA	28
4.42 Interaction diagram between KMSA and SYNCA	28
4.43 Sequence diagram between KMSA and SYNCA	29
4.44 Interaction diagram between NBH100A and SYNCA	29
4.45 Sequence diagram between NBH100A and SYNCA	30
4.46 Interaction diagram between MH100A and SYNCA	30
4.47 Sequence diagram between MHH100A and SYNCA	31
4.48 Interaction diagram between AA, SYNCA, GH104A, MH100A and NBH100A.....	31
4.49 Sequence diagram between AA, SYNCA, NBH100A and MHH100A	32
4.5 Prototype implementation, testing and evaluation.....	32
4.51 Virtualization and Web services	32
4.52 Virtual environment design.....	33
4.53 Testing and Evaluation	34
4.6 System Resources Required.....	37
4.61 Hardware Requirement:	37
4.62 Software Requirement:	37
4.63 Agents implementation	37

4.64 Database implementation.....	37
4.65 Virtual LAN implementation.....	37
4.66 Summary of Prototype Implementation.....	37
CHAPTER FIVE: PROTOTYPE TESTING AND VALIDATION	39
5.1 Introduction.....	39
5.2 Results Analysis (Input, Processing and Output).....	39
5.21 How System Objectives were achieved.....	39
5.22 Introducing a Staff/dependant into the scheme.....	40
5.23 Validating or Authenticating KRA Staff	43
5.24 Validating or Authenticating Dependand	45
5.25 Sending of invoices from Facility/Hospital to KRA.....	45
5.26 System Agents captured using the Sniffer Agent.	47
5.27 Shutting down agent platform and agent deregistration	48
5.3 System access and Usage.....	48
5.31 The login interface	49
5.32 CPM (Central Parameter Management) interface.....	49
5.33 User Management interface.	49
5.4 Sample output screen shots.....	49
5.5 KRAMAS System Monitoring.....	50
5.6 User friendly Crystal Report Outputs	51
CHAPTER SIX: CONCLUSION	53
6.1 Results and output.....	53
6.2 Research Objectives Achievement.....	53
6.3 Relationship to previous work	54
6.4 Agent based integration value to an organization.....	54
6.5 Major findings & Research Contribution.....	54
6.6 Recommendation to future researchers.....	54
REFERENCES	56
APPENDICIES	58
APPENDIX 1: KRAMAS UAT QUESTIONNAIRE	58
APPEDIX 2 : SAMPLE CODE (Synchronization Agent- SYNCA).....	59

LIST OF FIGURES & TABLES

Figure 1 : Overall Architecture	3
Figure 2 : MESSAGE Concepts diagram	15
Figure 3 : MESSAGE Analysis Activity diagram	17
Figure 4 : Coordination Model.....	22
Figure 5 : Online Validation Architecture	33
Figure 6 : Infrastructure Diagram	34

Table 1: Mater Hospital data source	35
Table 2: Nairobi Hospital data source.....	35
Table 3: Getrude Hospital data source.....	35
Table 4: Agent Responsibilities.....	36

DEFINITION OF TERMS

ACL – Agent Communication Language

ADP – Agent Development Process

AgDL – Agent Definition Language.

AgML – Agent Modelling Language

AMS – Agent Management System.

CLI – Command Line Interface.

DF – Directory Facilitator

DML – Data Manipulation Language

EMR– Electronic Medical Records

FIPA – Foundation for Intelligent Physical Agents

FSM – Finite State Machine

GUI – Graphical User Interface

IDE – Integrated Development Environment

JADE – Java Agent Development Environment

JDK/JRE –Java Development Kit/Java Runtime Environment

JVM – Java Virtual Machine

MESSAGE – Methodology for Engineering Systems of Software AGENTS

MAS – Multi Agent System

MaSE – Multi-Agent Systems Engineering

R/W – Read / Write Access

RMI – Remote Method Invocation

RDBMS – Relational Database Management System

SOA – Service-Oriented Architecture

TDD –Test-Driven Development

UDF – User Defined Functions

UML – Unified Modelling Language

VM – Virtual Machine

CHAPTER ONE: INTRODUCTION

1.0 Background to the problem

Kenya Revenue Authority has a workforce of about 4,500 members of staff who are distributed all over the country. The Authority has a medical scheme for all its members of staff and their dependants, who access the benefit through an approved network of fifty two (52) medical service providers. The Authority has a medical management system that manages staff and dependant records with such features as identification, entitlement management, staff entry and exit and management of treatment history from all the approved or accredited medical centres or facilities. This medical system is a standalone system and therefore does not transmit updated staff or dependant information to the medical centres or facilities. It does not also receive real time patient treatment data from the medical facilities once the patient has been attended to, but rather a hard copy of the invoice is delivered to the Authority. The medical card must be produced by the dependant or the staff ID before medical attention can commence after which the physical invoice raised must be signed by the patient before leaving the medical facility. The listed medical facilities or service providers are frequently furnished with an updated list of staff and beneficiary data, which they use to update their systems to minimise risks in the validation process.

The Authority has been facing various challenges in the past which include authenticating the invoice details. On the other hand, the medical service providers do not have a reliable means of identifying or authenticating KRA staff members and their beneficiary. When a service provider's invoice has been paid or settled by KRA, the service provider is not notified in real time of this update or if the invoice has been rejected. A printed list of the paid invoices is grouped in a batch alongside the payment information and sent to the respective service provider to settle the services that had been rendered. KRA and service providers have their own customized and heterogeneous patient and medical management systems, that run diverse RDBMs like Oracle, MS SQL, Informix, Mysql, Postgres and Sybase which are hosted on operating systems like Windows, Unix, Open source variants. These systems utilize different data types, syntactical and semantically varying data representations to refer to the information they hold or store. This poses a big challenge when trying to integrate or harmonise the same. All these diverse technologies need to be interlinked together so as to share data in a transparent way.

1.1 Problem statement

KRA manages its own stand alone medical system. The service providers on the other hand use customised proprietary systems to manage their patient records. There is no real time data relayed from the service providers to the KRA medical system and vice versa. The mechanisms used at the facilities or the medical centres for authenticating or validating KRA scheme members are not effective. Real time statistics regarding invoices from the service providers is not available.

1.2 Goal of the Study

To build and evaluate an integrated system that will enable inter linking between KRA medical system and the medical service provider systems to facilitate transparent information sharing for better decision making and compliance monitoring to reduce business risks.

1.3 Objectives of the study

Project objective:

To design, implement and evaluate an integrated data model between KRA medical system and the approved medical service providers.

System objectives:

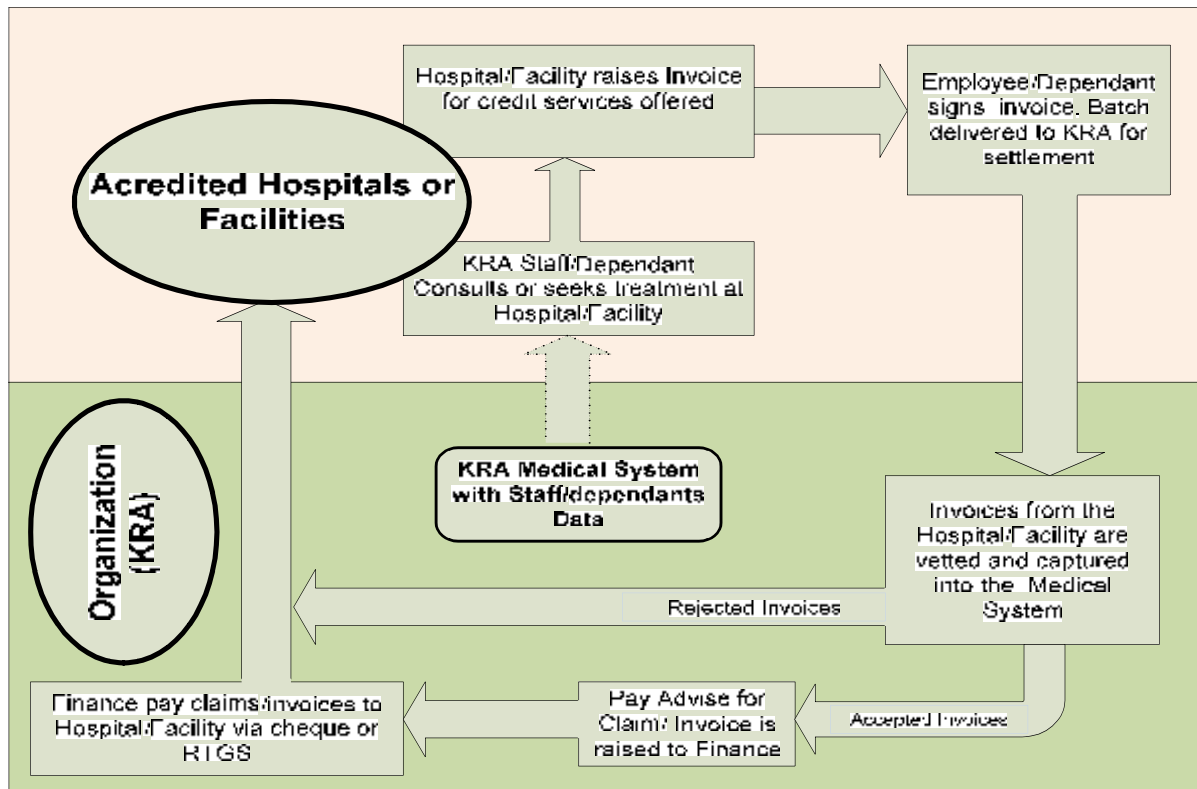
- a) To enable the approved medical service providers authenticate or identify all KRA members of staff or beneficiary by querying an online service.
- b) Have the KRA medical system notified by service provider systems about the invoices as they are generated i.e. the treatment information and the incurred cost. It should be possible to know the number of patients that have been treated per day and per facility.
- c) To ensure that the KRA medical system inform the service providers of the honoured/paid or the rejected invoices.
- d) To develop, evaluate and test the proposed model.

1.4 Significance of the study

- By integrating the heterogeneous systems, there will be timely decision making and planning for KRA. By having the real time invoices, the Authority will mitigate the risks of paying fraudulent invoices or double payment and also act as a reference for validating the physical invoices brought in for settlement from the various medical facilities.
- Medical service providers will be able to authenticate patients accurately since the web service will display patient photograph in addition to staff or dependant ID. The project will also pave way for the inclusion of a biometric identification system at the point of

treatment. The technology may be extended with ease to manage access control and staff movement within the KRA establishments.

Figure 1 : Overall Architecture



1.5 Business Analysis

The model will be designed to achieve the following business aspect:

- Enable the Authority to have access to the treatment invoices or bills in real time mode from the facility or the medical service providers. This will be beneficial since daily reports will be available for patients who accessed the scheme's medical benefits in whichever medical facility country wide, therefore leading to better decision making.

1.6 Assumptions and scope of the research

- a) The medical service providers or facilities have access to stable internet services and have a computerised patient management system supported by RDBMs in the back end.
- b) The integration and information exchange between the participating systems is done at the data level and not at the application level.
- c) The medical service providers are to provide access to treatment records, only for patients associated with the KRA scheme i.e. through a replication site or a secluded database on their end, where the local Agents will harvest data from.

- d) Service providers will use the web interface to authenticate patients seeking treatment or consultation by login in using the provided credentials. Access to this web interface will be limited only to authorised service providers and will be controlled by implementing an IP Access list on the firewall and implementing an IP traffic filter at the Apache web server level.

CHAPTER TWO: LITERATURE REVIEW

Introduction

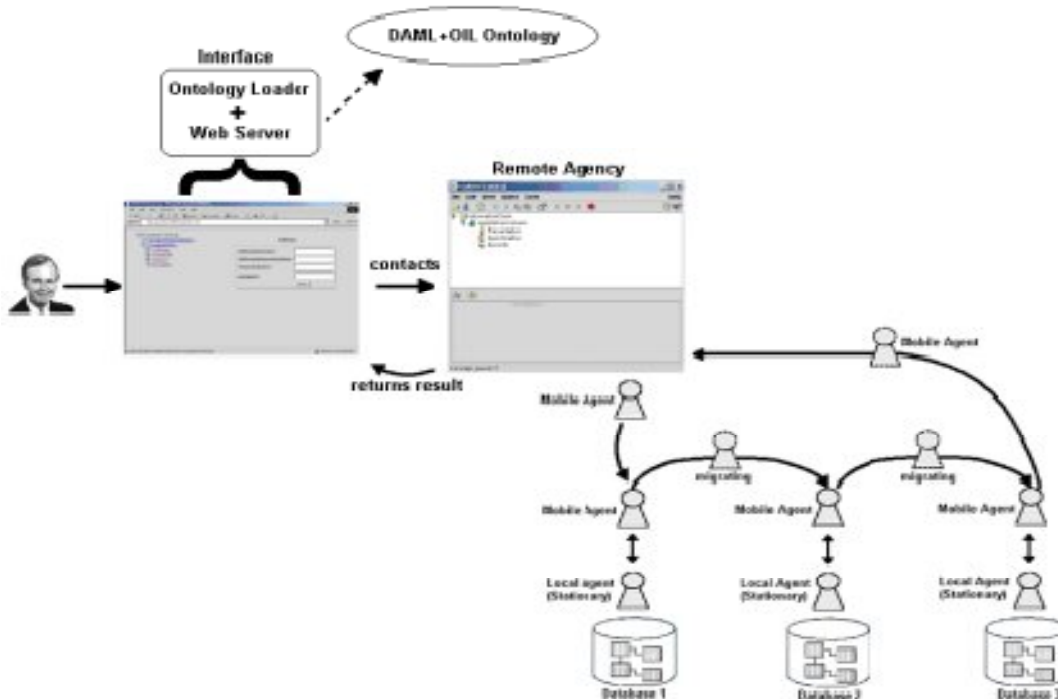
The deviation from the native and silo type or centralised systems has emerged out of the need to integrate disparate data systems as they stand. Systems developed to meet the needs of medical service providers have emerged and are usually designed for specific needs. Service provider mergers and purchases have also contributed to disparate systems within the medical domain (McDonald et al. 2000). The need to integrate these disparate systems has therefore resulted to heterogeneous distributed systems, which need to be harmonised so as to facilitate data sharing.

2.1 Related work on Data integration

There has been a wide range of approaches and methodologies proposed for the integrated access of information from autonomous systems and databases, stored in different software and hardware platforms, including federated database systems, multi-database systems, heterogeneous database systems, and mediator systems. The common integration approaches that have been attempted have been classified according to the level of abstraction at the point where the integration is being performed as defined below. (Ziegler et al. 2004).

2.1.1 The use of mobile agents in an itinerary

(Philip Et al,2002) have used mobile agents to solve the data integration models of Federated databases, which enforces the virtual approach; data warehousing, which is based on the materialized approach; and mediators, which uses the virtual approach. Garcia-Molina (Garcia-Molina et al, 2002) discusses some of the important issues regarding data integration that there are different data types in the different sources, data sources may use different names or format to refer the same object, the terms may have different semantics in the way they are stored in the data source and there are data that are not present in all the data sources. In this regard data Integration using agents (DIA) has been used by in this context for semantic integration of a federated database using mobile agents and ontologies where an itinerary is created, where the mobile agents must follow visiting each node along the path/and carrying with it the required data from that node. In this capacity the availability of an explicit ontology for the data sources was at the core of the integration process, as it represented the global schema of the federated database. The following diagram illustrates the integration approach taken by (Philip Et al,2002).



The payload is in the form of XML which increments in size as the mobile agents move from node to the other within the route. This kind of approach cannot be used in our case because of the number of nodes within the route/itinerary that will need to be visited.

2.12 Ontology in integrating heterogeneous RDBMs

The notion of Ontology provides definitions for the vocabulary used to represent knowledge in a certain domain like medical. Therefore to share and interchange information among different database systems would involve the presence of a common vocabulary, because syntactic and semantic conflicts would result from lack of standardization (consistency) in the meaning of concepts, terms and structures found in these data sources. Ontology does the standardization since it requires a precise semantic representation. There are two ways that have been used to resolve semantic conflict using ontology, first is to build the ontology for each system, then directly create the mapping between these ontology-based systems, or to create common ontology to specify the vocabularies and terms to describe and interpret shared information among its users, and build the mapping between the systems through the common ontology (Huang, 2002).

2.13 Common Data Storage integration

In this approach, the physical data integration is performed by migrating the existing data to a new environment and the local data sources can either be retired or remain operational. Physical data integration provides fast data access, the only problem is that discontinuing or retiring the local

data source will render the applications that access these datasets unusable until they are made to point or reference the new environment (McDonald,2000). If the local data sources or legacy systems must remain operational then periodic updating or refreshing of the common data storage elements must be factored in.

2.14 Additional layer integration

This is the inclusion of an additional layer of software for logical data integration and is based on the concept of integrated global conceptual schema which is achieved after converting the local schemas into the global model. This integration process requires expensive and time consuming alterations and modifications that need to be done on the local applications and existing conceptual model. According to Barret (1996), other approaches involve the use of frameworks that integrate specific data sources and use a particular query language that users need to learn in order to use the system. According to Andrea (2001), the creation of the global conceptual schema is supported by local conceptual model comparison, equivalence identification and conflict resolution. This is achieved by translating the local representation of data into global standards, the local and related applications are modified as well as the local data structure. Solutions based on mediators provide data access through a specific software layer between users applications and data sources. This middle-layer provides data access and data integration to users, but as pointed out by Cohen (2000), the main problem with this solution is that it requires additional processing and the inclusion of rules for each participant database and hence not suitable for our case.

2.15 Service-Oriented Architecture integration

SOA generally provides a way for consumers of services, such as web-based applications, to be aware of available SOA-based services. XML is often used for interfacing with SOA services. SOA defines how to integrate widely disparate applications for a Web-based environment and uses multiple implementation platforms. According to Wikipedia (2013), rather than defining an API, SOA defines the interface in terms of protocols and functionality. Service-orientation requires loose coupling of services with operating systems and other technologies that underlie applications. SOA separates functions into distinct units, or services, which developers make accessible over a network in order to allow users to combine and reuse them in the production of applications. These services and their corresponding consumers communicate with each other by passing data in a well-defined, shared format, or by coordinating an activity between two or more services triggered at the application level.

SOA is not an option to be considered in our integration problem because the systems requiring harmonization are not web based and also the fact that our scope revolves around the data and not application level integration.

2.16 Build new System to meet requirements

This includes *Building a new System* where a new system is developed to cater for the needs of the stake holders by soliciting user requirements as the initial step. *Web services integration* is also widely adopted where the data exchange occurs over the internet and is commonly achieved via XML based (Extensible Mark up language). In *Common Data Storage & Warehousing*, the warehouse system extracts, transforms, and loads data from heterogeneous sources into a single common schema hence data becomes compatible with each other. However, this approach offers a tightly coupled architecture because the data is already physically reconciled in a single repository at query-time, so it usually takes little time to resolve queries. But problems lie in the data freshness, that is, information in warehouse is not always up-to-date. Therefore, when an original data source gets updated, the warehouse still retains outdated data and the ETL process needs re-execution for synchronization. Change tracking (to monitor what may have changed) can be done using various methods such as enabling CT in MS-SQL server or by using triggers and UDFs. *Integration By Middleware* is a mediation kind of interface e.g. SQL middleware or JADE framework where the Agent platform can be distributed over the LAN or WAN and the Computers need not share the same OS or similar Data sources and it is the approach we have adopted in our case. Also there exists *Data Replication* which entails copying and maintaining database objects in multiple databases that constitute a distributed database system.

2.17 The use of commercial tools in data replication

This includes tools such as *DBSync* which can only do updates by synchronising the entire database but would take hours in case of a large data source. *Daffodil Replicator (E)* is another tool based on Java and ideal for data synchronization, data integration, data migration, and data backup between database servers. Daffodil Replicator works over standard JDBC driver and supports replication across heterogeneous databases. At present, it supports following databases: Microsoft SQL Server, Oracle, Daffodil DB, DB2, Derby, MySql, PostgreSQL and Firebird. Another proprietary tool is the *Microsoft BizTalk Server* which enables companies to integrate and manage automated business processes by exchanging business documents such as purchase orders and invoices between disparate applications, within or across organizational boundaries. Another

method that has been used is *two-way sync* with triggers which is tricky because the triggers could fire each other and cause strange locking errors will crop up. Linked servers can also be used in an environment where MSDTC (Distributed Transaction Coordinator) is already set up. If the schemas match, then replication should be ideal for both one-way and two-way synchronization.

2.2 The Agent Architecture

According to Wooldridge (2002), the structure of an agent generally consists of three parts namely: *input*, *process* and the *output*. The input consists of a sensor which captures the changes and stimuli of the environment in which the agent lives. The process part, also called kernel, processes the input data, performs the reasoning, and makes decisions based on knowledge and rules. The output part takes the appropriate actions to fulfill the agent's goals which may be to change the environment. A multi-agent system is a system that consists of a number of alive agents working together in an environment to accomplish designed goals in order to solve a common problem. By the above definition therefore, an agency system has four key elements: alive agents, cooperation, environment, and goals.

Alive agents: Refers to active entities such as persons, robots, intelligent objects, software agents that have resources, possess knowledge and skills, and can autonomously perform actions.

Cooperation: The agents are required to socialise in their natural environment or social networking using their defined rules and communication languages.

Environment: This represents the settings in which the agents live and act, which acts as a source of motivation and stimuli.

Goals: A MAS must fulfil the objectives to achieve its goals for which it was designed for.

2.21 JADE Framework (with MAS support)

JADE is a software framework fully implemented in Java language. It simplifies the implementation of multi-agent systems through a middle-ware that complies with the FIPA specifications. The agent platform can be distributed across machines over the LAN or WAN and which do not need to share the same kind of OS and the configuration can be done through the remote GUI. An application based on JADE is made of software agents each one having a unique name and these agents execute tasks and interact by exchanging messages. Agents live on top of a *Platform* that is composed of one or more *Containers*. Containers can be executed on different hosts thus achieving a distributed platform and each container can contain zero to n agents. There is a special container called the *Main Container* that exists in the platform and can contain agents but it is unique from any other container that may be started because;

- a) It must be the first container to start in the platform; all other containers will register to it at boot.
- b) It includes two special agents: the *AMS* that represents the authority in the platform and is the only agent able to perform platform management actions such as starting and killing agents or shutting down the whole platform. The *DF* that provides the Yellow Pages service where agents can publish the services they provide and find other agents providing the services that they need.

We have chosen JADE because of its open source nature, simplicity and is also compliant with the FIPA specifications. JADE conceptualises an agent as an independent and autonomous process that has an identity, possibly persistent, and that requires communication (e.g. collaboration or competition) with other agents in order to fulfil its tasks. This communication is implemented through asynchronous message passing.

2.22 Agent Communication

The agents must communicate between them to cooperate through, querying and identifying other agents, expressing desires and intentions to another agent, transmitting data to agents, task coordination with other agents. Each communication is characterized by a communication mode, a communication protocol, a communication language, and by using ontology's that represent common knowledge of a domain.

2.23 Communication Modes

There exist three communication modes:

One-to-One. It is also called point-to-point. This communication mode allows an agent to send a message to another agent. For instance a facility agent communicates with the synchronization agent to relay the treatment medical record. *One-to-Many*. It is also called multicast. This communication mode allows an agent to send a message to a group of agents.

One-to-All. It is also called broadcast. This communication mode allows an agent to send a message to all agents in the system, such as a notification of an important event, an alert. For instance when a member of staff ceases to be a member of the scheme then all facility agents need to be notified of the new developments.

2.24 Communication Languages

Since the communication between agents is based on the exchange of messages, each message must be encoded in a language that the agents understand. The communication language is divided into two parts; the languages used to define the structure of a message and those to describe its

content or the body. The structure of a message contains the sender's address, the receiver's address, the subject of the message, the information content to be sent, the language which defines the syntax and the encoding scheme of the information content and the name of ontology if one has been defined. The languages used to define the structure of a message include KQML and FIPA ACL while those that describe the content include KIF and FIPA SL. The communicative intention or simply the *performative* indicates what the sender intends to achieve by sending the message to the receiver. The performative can be REQUEST, if the sender wants the receiver to perform an action,(e.g. in our case the facility agents request the synchronization agent to update the central treatment history repository on their behalf). INFORM, if the sender wants the receiver to be aware a fact (e.g. the KMSA-KRA Medical System Agent), QUERY_IF, if the sender wants to know whether or not a given condition holds, CFP (call for proposal), PROPOSE, ACCEPT_PROPOSAL, REJECT_PROPOSAL, if the sender and receiver are engaged in a negotiation. The content i.e. the actual information included in the message (i.e. the action to be performed in a REQUEST message, the fact that the sender wants to disclose in an INFORM message).

2.3 Methodologies for MAS Engineering

According to Sycara (1998), there are two technical hurdles to the extensive use of MAS, beginning with lack of a proven methodology for structuring systems into MAS and secondly, there is no ultimate tool for MAS engineering in the industry. Few selected methodologies are analysed below:

2.31 SADAAM Methodology

Software Agent Development An Agile, Methodology. SADAAM is an agent development methodology that utilises methods borrowed from diverse Agile Development methods, like the ones found in Extreme Programming (XP) to facilitate the development and implementation of software agents into MAS (Beck, 2003). According to Amor (2004), the approach used in SADAAM in agent development lays focus towards delivery of working code rather than the documentation and a detailed upfront design. SADAAM is an ADP that provides the core agile agent development process which consists of four main phases: Design, Test-Driven Implementation, Release and Review which are applied iteratively until the product is finished. The ADP methodology supports a bottom-up approach that increases flexibility and enables the development team to focus on the rapid delivery of working code, and hence give room to respond quickly to the changes in system requirements.

Advantages of SADAAM is that it promotes iterative, incremental, testing and development of agent-based systems; and injecting flexibility into the development cycle, therefore greatly increases productivity.

Disadvantage of SADAAM includes adoption of a minimal approach to the requirements capture, thus there is the danger of requirements creep or instability.

2.32 Gaia Methodology

According to Kinny et al.(1997), the Gaia methodology models both the macro (social) aspect and the micro (agent internals) aspect of the MAS. Gaia takes the view that a system can be seen as a society or an organization of agents. The methodology is applied after the requirements are gathered and specified, and covers the analysis and design phases. Gaia methodology is intended to allow an analyst to go systematically from a statement of requirements to a design that is sufficiently detailed which can be implemented directly. The requirements capture phase is independent of the paradigm used for analysis and design. In applying Gaia, the analyst moves from abstract to increasingly concrete concepts. Each successive move introduces greater implementation bias, and shrinks the space of possible systems that could be implemented to satisfy the original requirements statement. The key concepts in Gaia are roles, which have associated with them responsibilities, permissions, activities, and protocols. Roles can interact with other roles.

2.33 MaSE Methodology

The MaSE Analysis stage includes the Capturing of Goals where goals are identified and put into goal hierarchy form then use cases are used to elicit the overall system behaviour and sequence diagrams illustrating how activities relate to each other are also drawn. The final step constitutes refining of Roles and the output is the Role Model which describes the roles in the system. The Design Phase follows where the Agent Classes are created with the Agent Class Diagram as the corresponding output. The agent class diagram shows the agents and the roles they play within the system. Then the conversation diagrams are constructed, they depict communication between agents as defined by use cases. The Agent classes are assembled together to define the system behaviour. System design forms the final stage where deployment diagrams are created to indicate agents in the model. According to Wooldridge (1999), there have been several proposed methodologies for analyzing, designing, and building multi-agent systems. The majority of these are based on existing object oriented or knowledge-based methodologies.

MASE Advantages:

- a) MaSE has extensive tool support in the form of *agentTool* and also provides automated support for transforming analysis models into design constructs.
- b) The main advantage of MaSE over previous methodologies is its scope and completeness.
- c) MaSE does not dictate any particular implementation platform.

MASE Disadvantages :

- a) MASE assumes that the system being designed is closed and that all external interfaces are encapsulated by an agent that participates in the system communication protocols.
- b) The methodology does not consider dynamic systems where agents can be created, destroyed, or moved during execution.
- c) Inter-agent conversations are assumed to be one-to-one, as opposed to multicast. However, substituting a series of point-to-point messages can be used to fulfil the requirement for multicast.
- d) MaSE is not very good in testing and debugging but the agents will be tested using the graphical testing and debugging tool provided by JADE platform.

2.34 MESSAGE Methodology

Methodology for Engineering Systems of Software AGENTS methodology covers MAS analysis and design and is designed for use in mainstream software engineering departments. UML concepts are used to model MESSAGE entities at a detailed (or micro) level.

2.35 MESSAGE Concepts

Most of the MESSAGE knowledge level entity concepts fall into three main categories namely *ConcreteEntity*, *Activity*, and *MentalStateEntity*. According to (Dardenne et.al,1993). The main types of *ConcreteEntity* are defined below;

Agent: An Agent is an atomic autonomous entity that is capable of performing some (potentially) useful function. This functional capability is captured as the agent's services.

Organisation: An Organisation is a group of Agents that work together towards a common purpose or goal. It is a virtual entity in the sense that the system has no individual computational entity corresponding to an organisation and its services are provided and purpose achieved collectively by its constituent agents.

Role: The Role describes the external characteristics of an Agent in a particular context. An Agent may be capable of playing several roles, and multiple Agents may be able to play the same Role. Roles can also be used as indirect references to Agents.

2.36 Types of Activities

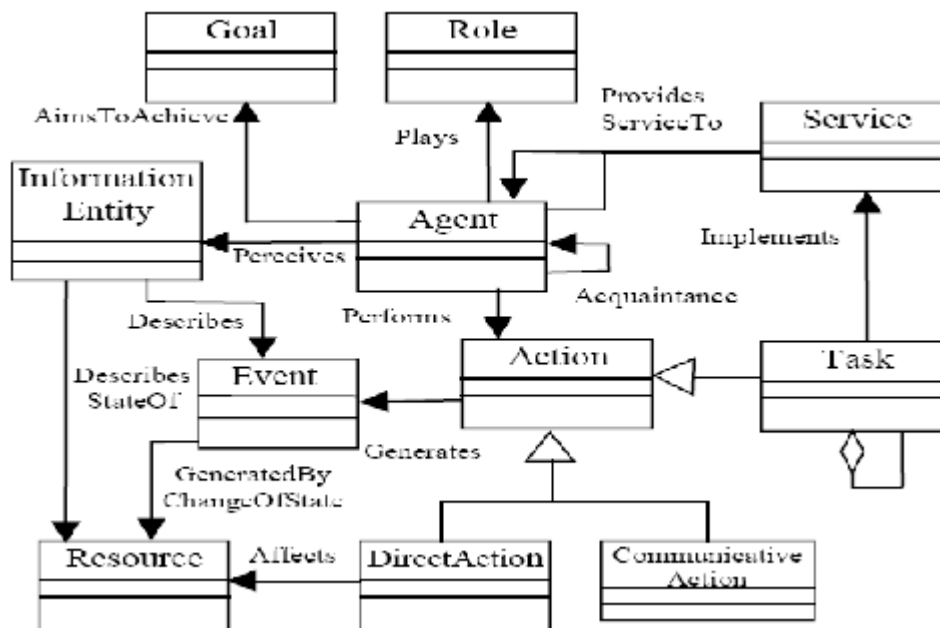
Task: A task has a set of pairs of Situations describing pre- and post-conditions. If the Task is performed when a precondition is valid, then one can expect the associated post condition to hold when the Task is completed. Tasks are State Machines, so that e.g. UML activity diagrams can be used to show temporal dependencies of sub-tasks.

Interaction and Interaction Protocol: The MESSAGE concept of Interaction borrows heavily from the Gaia methodology (Wooldridge, 1999). An Interaction has more than one participant, and a purpose which the participants collectively must aim to achieve. An Interaction Protocol defines a pattern of Message exchange associated with an Interaction. The internal architecture of an agent typically is based on one of several models derived from cognitive psychology. MESSAGE is intended to be applicable to a variety of agent cognitive architectures.

Goal: A Goal associates an Agent with a certain Situation. Some Goals are intrinsic to the agent's identity, and are derived from its purpose. These persist throughout the life of the Agent. Others are transient tactical Goals. It is often useful to express the purpose in terms of a utility function that associates 'goodness values' with Situations. The target situation of the Goal is then the one that is estimated to maximise utility.

Information Entity, this means an object encapsulating a chunk of information and Message. The agent-oriented concept of Message differs from the object-oriented one in a number of respects. In UML, a Message is a causal link in a chain of behaviour, indicating that an Action performed by one object triggers an Action by another object. In MESSAGE, a Message is an object communicated between Agents. Transmission of a Message takes finite time and requires an Action to be performed by the Sender and also the receiver. The attributes of a Message specify the sender, receiver, a speech act (categorising the Message in terms of the intent of the sender) and the content (an Information Entity).

Figure 2 : MESSAGE Concepts diagram



2.4 ANALYSIS MODEL VIEWS

An analysis model is a complex network of inter-related classes and instances derived from concepts defined in the MESSAGE/UML meta model. MESSAGE defines a number of views that focus on overlapping sub-sets of entity and relationship concepts that include;

Organisation view (OV) – This shows Concrete Entities (Agents, Organisations, Roles, Resources) in the system and its environment and the finer relationships between them. An acquaintance relationship indicates the existence of at least one Interaction involving the entities in the model.

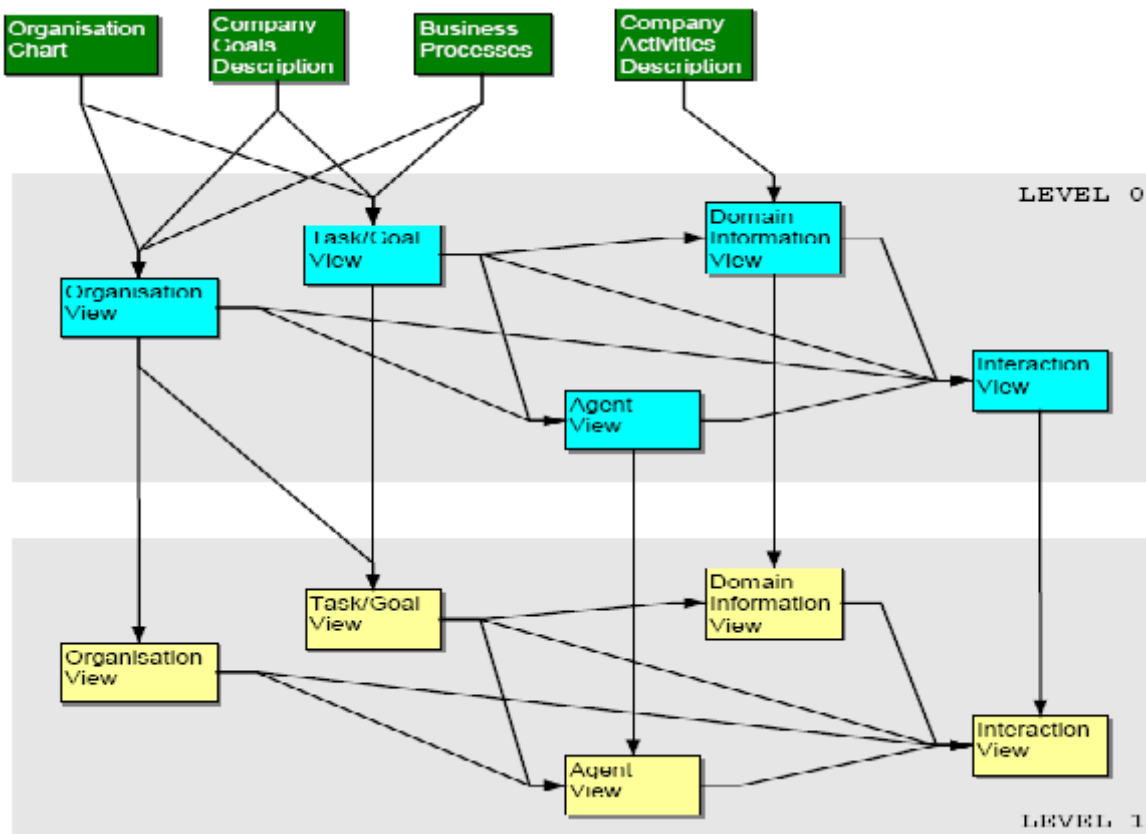
Goal/Task view (GTV) – It shows the Goals, Tasks, Situations and the dependencies among them. Goals and Tasks both have attributes of type Situation, so that they can be linked by logical dependencies to form graphs that show e.g. decomposition of high-level Goals into sub-goals, and how Tasks can be performed to achieve Goals.

Agent/Role view (AV) – Focuses on the individual Agents and Roles. For each agent or role it uses schemata supported by diagrams to its characteristics such as what Goals it is responsible for, what events it needs to sense, what resources it controls, what Tasks it knows how to perform, 'behaviour rules'.

Interaction view (IV) – For each interaction among the agents or roles, it must indicate the initiator, the collaborators, the motivator (generally a goal the initiator is responsible for), the relevant information supplied/achieved by each participant, the events that trigger the interaction, other relevant effects of the interaction (e.g. an agent becomes responsible for a new goal).

Domain view (DV) – This shows the domain specific concepts and relations that are relevant for the system under development (e.g. for a system dealing with data integration, then this view will show concepts like hospital, treatment, invoice and patient).

Figure 3 : MESSAGE Analysis Activity diagram



2.5 The Analysis Process

The purpose of Analysis is to produce a model of the system to be developed and its environment, that is agreed between the researcher and the consumer and other stakeholders. It aids communication between the system developer and the customer, and provides a basis from which design can proceed with confidence. The analysis models are produced by stepwise refinement.

2.51 Refinement Approach:

The top level of decomposition is referred to as level 0. This initial level is concerned with defining the system to be developed with respect to its stakeholders and environment. The system is viewed as a set of organisations that interact with resources, actors, or other organisations. Actors may be human users or other existing agents. At level 0 the modelling process starts building the Organisation and the Goal/Task views. These views then act as inputs to creating the Agent/Role and the Domain Views. Finally the Interaction view is built using input from the other models. The level 0 model gives an overall view of the system, its environment, and its global functionality. The granularity of level 0 focuses on the identification of entities, and their relationships according to the meta model. In level 1 the structure and the behaviour of entities such as organisation, agents, tasks, goals domain entities are defined.

2.52 Analysis Refinement strategies

The agents needed for achieving the goals appear naturally during the refinement process. Then co-operation, possible conflicts and conflict resolution may be analysed. Agent centred approaches focus on the identification of agents needed for providing the system functionality. The most suitable organisation is identified according to system requirements. Interaction oriented approaches suggest progressive refinement of interaction scenarios which characterise the internal and external behaviour of the organisation and agents.

The goal or task decomposition approaches are based on functional decomposition where System roles, goals and tasks are systematically analysed in order to determine the resolution conditions, problem solving methods, decomposition and failure treatment. Task preconditions, task structures, task output and task post-condition may determine what Domain Entities are needed. Goals and tasks must be performed by agents playing certain roles.

CHAPTER THREE: METHODOLOGY

3.0 Introduction

The transformation of the research problem into a working prototype or model has been solved by creating a virtual environment and hosting four virtual machines which have been configured with windows and FOSS Operating system. RDBMs have been installed (MS-SQL, Oracle and Mysql) and a web server platform configured (PHP and Apache Tomcat),that will be used by the medical service providers to authenticate members of staff and their dependants. The agent decomposition process and assignment of the defined tasks to the respective agents has been illustrated and to determine the number of agents to be used, consideration has been made depending on the number of participating entities

The milestones that have been followed include the following:

- Requirements gathering
- MESSAGE analysis and design
- System implementation ,testing and evaluation
- Documentation of all the work done

3.1 Requirements gathering

This has been done through reviewing documentation, journals and interviewing the medical system users at KRA. Also selected facility/hospitals were visited and a questionnaire used to elicit the situation on the ground.

3.2 Sources of data and its relevancy

The Medical facilities send or provide to KRA the hard copy of the bills or invoices for settlement. To facilitate this research, electronic copies of the same have been provided by selected medical providers in the form of data dumps as well as replicated database consisting of the relevant key information that contain treatment records for KRA related patients only. This is where the local agents will be searching and retrieving the required information and updating the status of the same once an invoice has been paid. KRA medical department did provide access to staff and beneficiary data in the format normally sent to the service providers.

3.3 MESSAGE methodology in System development

The main focus of MESSAGE is on the phase of analysis of agent-based systems, based on five analysis models classified as follows:

- The *Organization Model* captures the overall structure and behaviour of a group of agents, and the external organization, working together to reach common goals
- The *Goal/Task Model* defines the goals of the composite system (the agent system and its environment) and their decomposition into sub goals.
- The *Agent Model* consists of a set of individual agents and roles. The relationship between a role and agent is defined analogously to that between an interface and an objectClass: a role describes the external characteristics of an agent in a particular context
- The *Domain Model* functions as a repository of relevant information about the problem domain. The conceptualization of the specific domain is taken to be a mixture of;
- *Object-oriented approach* (by which all entities in the domain are classified in classes, and each class groups all entities with a common structure) and *Relational* (by which a number of relations describe the mutual relationships between the entities belonging to the different classes).
- The *Interaction Model* is concerned with capturing the way in which agents (or roles) exchange information with each another (as well as with their environment).

The analysis, design and implementation is carried out in the next chapter (chapter 4).

3.4 System implementation, testing and Validation.

The IDE selected is Netbeans given that it is freely available and has adequate API ready for use. JADE frameworks was used to derive the java agents that communicate by exchanging messages as defined or governed by the FIPPA standards. Unit, system and regression testing was the approach we used bounded by instances of sanitized real data (to conceal the sensitive patient information) that had been derived from the systems being integrated. In validating the web interface, we used the real employee staff IDs since it is what is presented to the facilities during treatment, and they are to confirm if patient is a beneficiary of the KRA medical scheme.

3.5 System Documentation

The documents produced at every stage of system development have been bundled into technical and user based documentation together generating this report write up.

CHAPTER FOUR: ANALYSIS, DESIGN AND IMPLEMENTATION

4.1 Introduction

KRAMAS has been developed for data integration at the database level. Other mechanisms of integration are possible as discussed in chapter two. The user requirements are defined as what the user wishes to derive from the system. They include: KRA medical section to be able to know who has been treated and at what hospital. On the other hand, medical services providers should be updated whenever a certain invoice has been paid, also they need to have an efficient way of validating stuff when they seek medical attention.

4.11 System Analysis requirements

These are the specifications which include:

- Description of the Coordination model design
- The input types and number into the processing and the output (s).
- The operation that system agents performs for each input of data.
- The nature of output per each input that is directed to the system.

KRAMAS has been developed using five (5) agent programs which are the processing components of the system where some agents take in data, typecast or format before invoking the next process .Only the interface agent has a GUI, the others work in the background once started but are monitored and managed using the AMS.

4.12 Coordination model design

This coordination design illustrates the interaction between the system agents and how they relate to each other. Every facility or hospital has its agent which retrieves the relevant data from its source and relays the same to the synchronization agent SYNCA which then stores the received data permanently into a relational database. The CPM module has configuration items such as the frequency of sending data for the facility agents (polling parameters), the number of trials the agents should retry their respective transaction before scheduling the transaction as a task to be executed later by the agent which has been assigned that task.

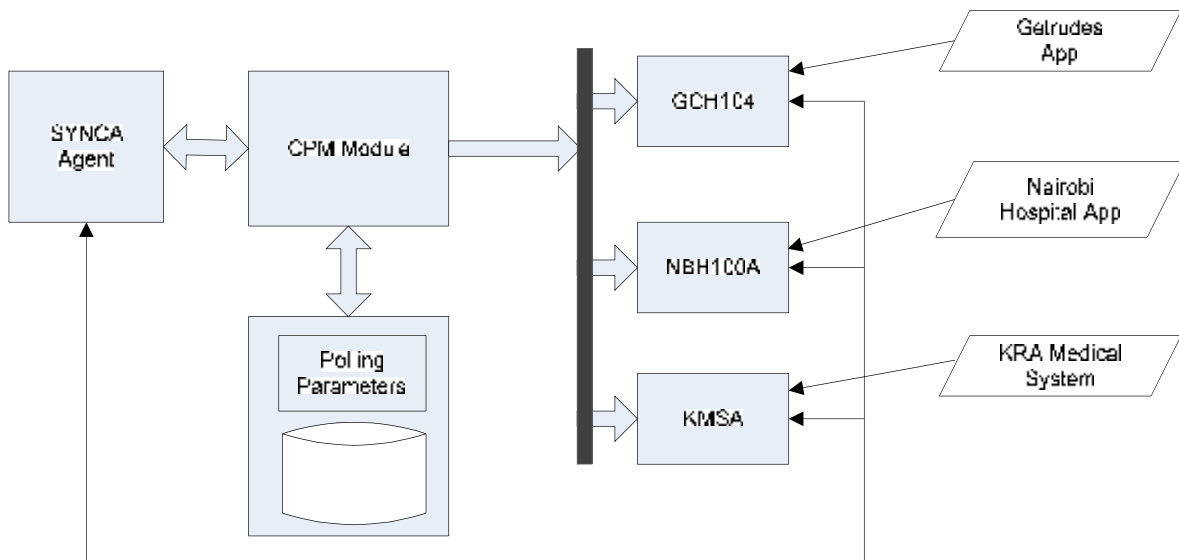
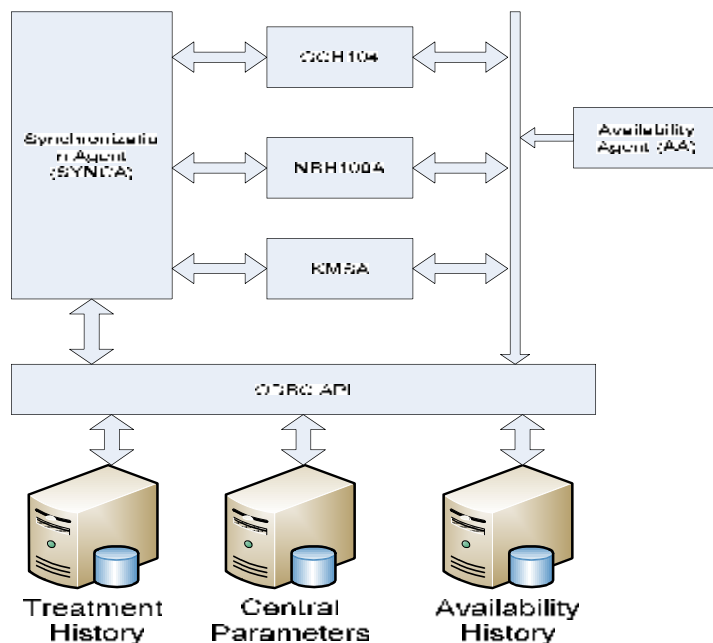


Figure 4 : Coordination Model

4.13 Agent Data Communication Design

The agents are able to access the CPM – Central Parameter Management database which is critical for their decision making process and they do this using the defined ODBC APIs. The Availability Agent, AA monitors the uptime of the other registered agents and registers in the availability database for the sake of capacity planning needs. Only the SYNCA has R/W access to the treatment history thereby ensuring data integrity of the relevant records from the sources.

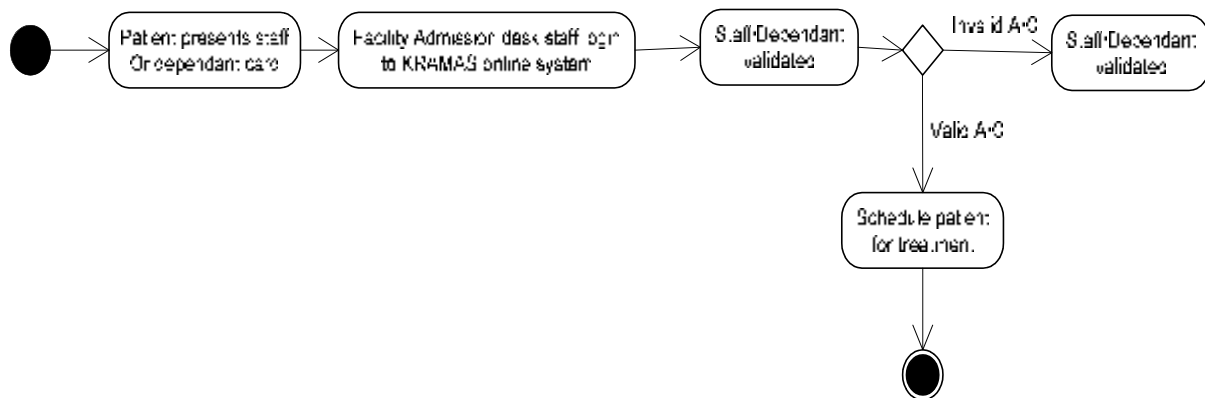


4.14 Activity and Interaction Diagrams

The sequence of activities involving a member of staff or beneficiary while seeking the medical services and the involvement of the KRA medical system is as shown below.

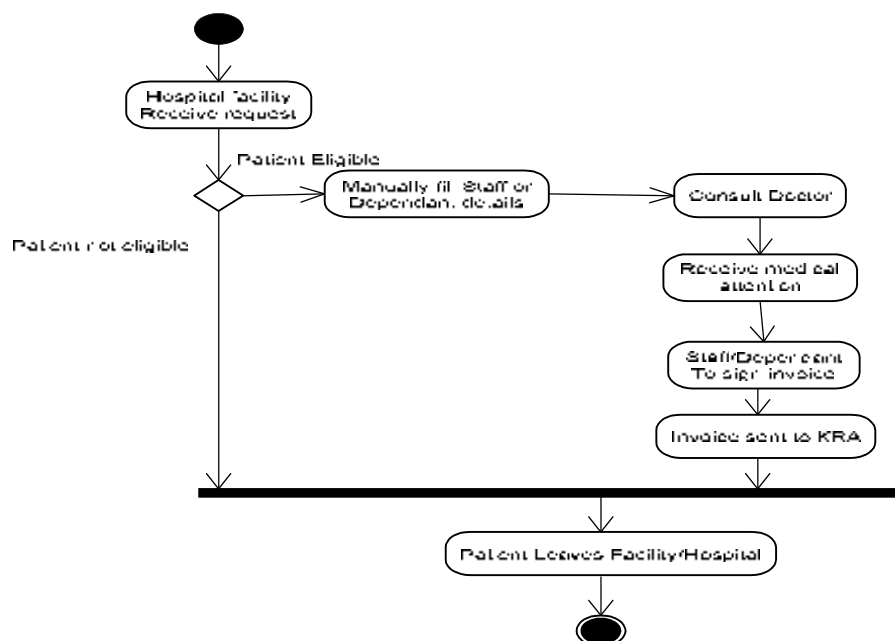
4.15 Staff/Patient accessing medical services

This process stipulates the various steps that are currently executed when the patient is seeking medical attention.



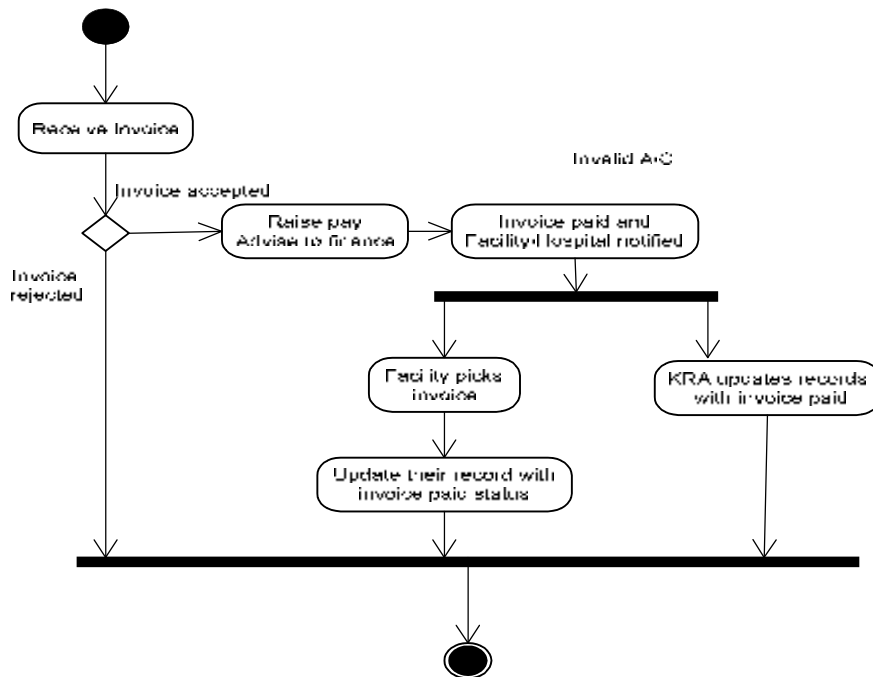
4.16 Treat Patient (Staff/Dependant)

The patient visits the facility, receives the required medical services and gets discharged or gets admitted. Soon after discharge, the responsible agent at that facility will pick each and every copy of all patients treated who only and only belong to KRA medical scheme. Each patient being treated at a given facility is associated with a certain organization, and this organization is represented by a unique ID and this is how the facility agents are able to know or differentiate which records to transmit or ignore given that the facility uses a single system for all its patients.



4.17 Invoice payment to Facility/Medical Service providers

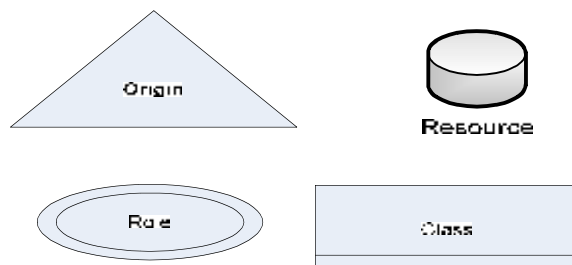
At this point the invoices from the facilities have already been received and the finance personnel just confirms from the system if the physical invoices indeed exist in the system and proceed to pay the same if the outcome is true else payment is halted (implying fraudulent invoice or one in error).

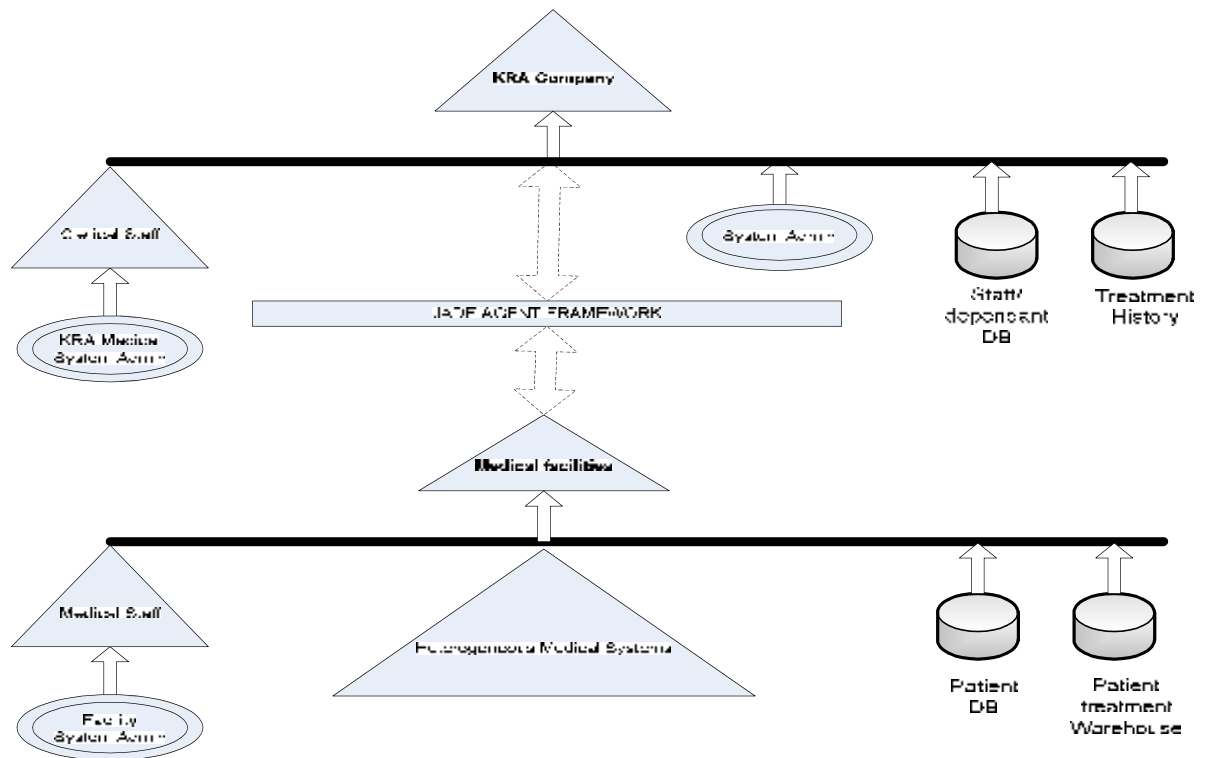


4.2 Analysis using the MESSAGE concept

The Structural Relationship between the entities involved in data integration is illustrated below with the following symbols or notations been used in the Analysis and Design work.

MESSAGE notations used:





4.3 Agent Task and Responsibility Analysis

The following are agents that comprise the model, the tasks or the services to be done, goals to be achieved as well as the collaborators.

a) **SYNCA (Synchronization agent)**- This has the duties

- i. Consult the CPM for polling frequency to use.
- ii. Receive INSERT, UPDATE and DELETE instructions from other agents
- iii. Check if the operation is permissible
- iv. Update the relevant record as per the primary key
- v. Inform the sender of that instruction that it has been made (else try three times and if still no success then mark it as uncompleted task to be done later)
- vi. Sleep and wait for new instruction.

Main Goal: Update the central repository

Sub goal: Notify the instruction sender agents of transaction status

Collaborators: Synchronization Database

- ✓ MH100A
- ✓ NBH100A
- ✓ GCH104A
- ✓ AA

b) KMSA (KRA Medical System Agent) The duties include:

- i. Consult the CPM for polling frequency to use.
- ii. Check the KRA medical system log tables (staff/beneficiary) for updates based on INSERT, UPDATE or DELETE operations.
- iii. If updates are found then check the operation type and format the data
- iv. Send the instruction request to the SYNCA to update the record based on the primary key.
- v. Wait for an ACK message from the SYNCA to signify task completed.
- vi. Get the next set of data meeting the criteria specified in step 1.

Main Goal: Have staff/dependant updates relayed

Collaborators: KRA medical system data source

- ✓ SYNCA
- ✓ AA

c) AA (Monitoring and availability Agent) The duties include:

- i. Consult the CPM for polling frequency to be used.
- ii. Send a broadcast message to all the agents in the system
- iii. Wait for a response from each Agent
- iv. Mark agent as non responsive after three(3) trials
- v. Repeat from step 1 above

Main Goal: Check uptime of system agents and record in a database

Collaborators: Synchronization Database

- ✓ SYNCA
- ✓ KMSA
- ✓ MH100A
- ✓ GCH104A
- ✓ NBH100A
- ✓ AA

d) n Facility Agents

- i. If it is a new facility created /accredited by KRA, then its data structure must be modified to reflect the data source architecture.
- ii. Else if it's a branch of an existing Facility/Hospital then an existing agent class is used and new branch name given to the agent during runtime.

- iii. Its CPM parameters are configured by the admin where the agent can now consult it for the polling frequency it is going to use.
- iv. Check the KRA medical system log tables (staff/beneficiary) for updates based on INSERT, UPDATE or DELETE operations.
- v. If updates are found then check the operation type and format the data
- vi. Send the instruction request to the SYNCA to update the record based on the primary key.
- vii. Wait for an ACK message from the SYNCA to signify task completed.
- viii. Get the next set of data meeting the criteria specified in step 1.

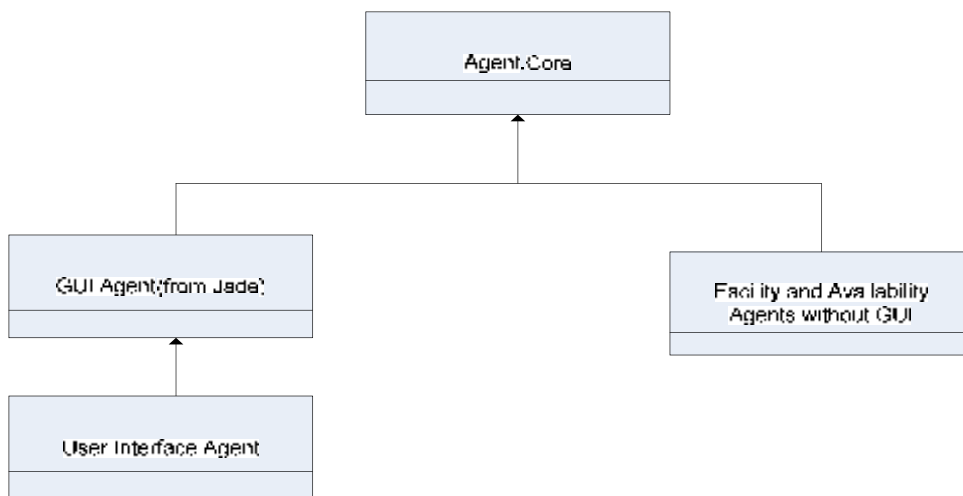
Main Goal: Have staff/dependant updates relayed

Collaborators: KRA medical system data source

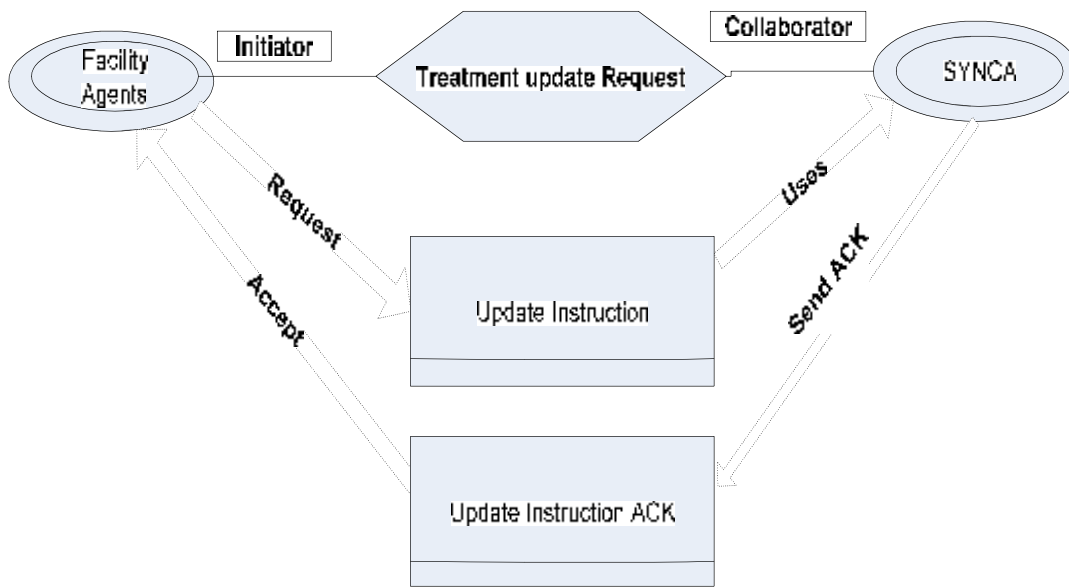
✓ SYNCA

✓ AA

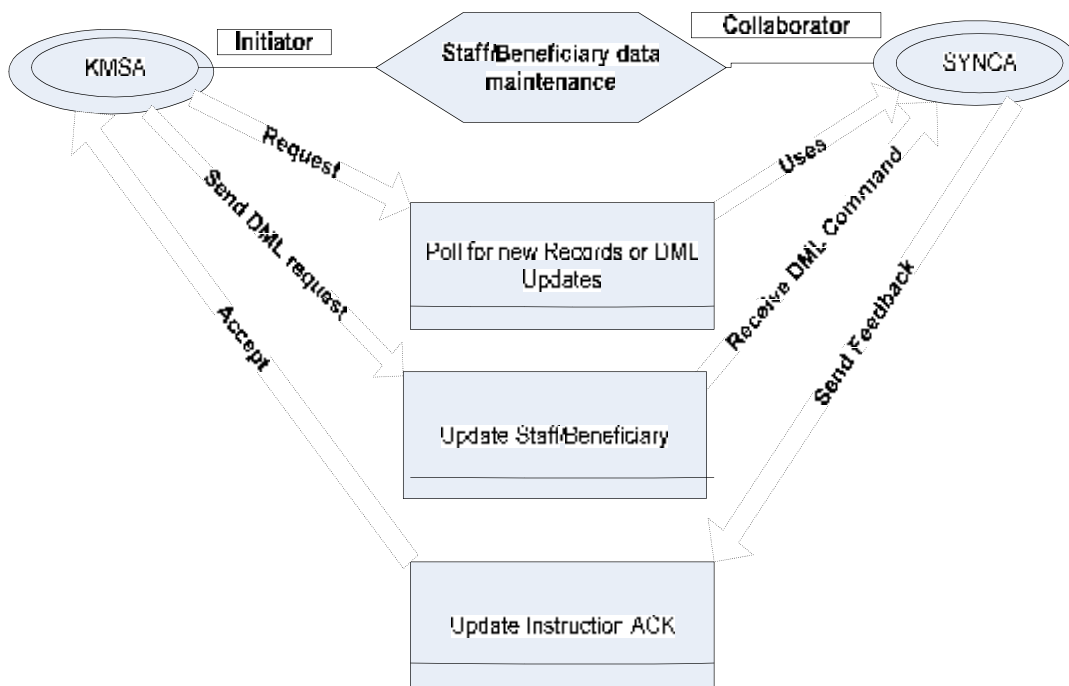
4.4 General class, Interaction and Sequence Diagrams



4.41 Global Interaction diagram between Facility Agents and SYNCA

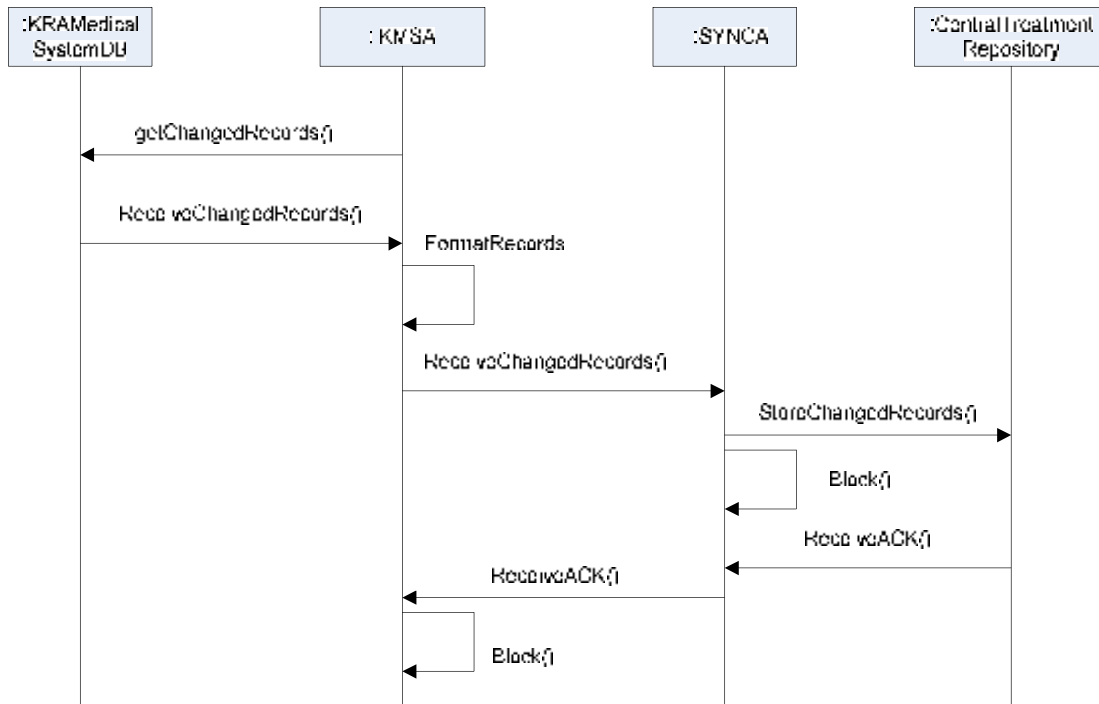


4.42 Interaction diagram between KMSA and SYNCA

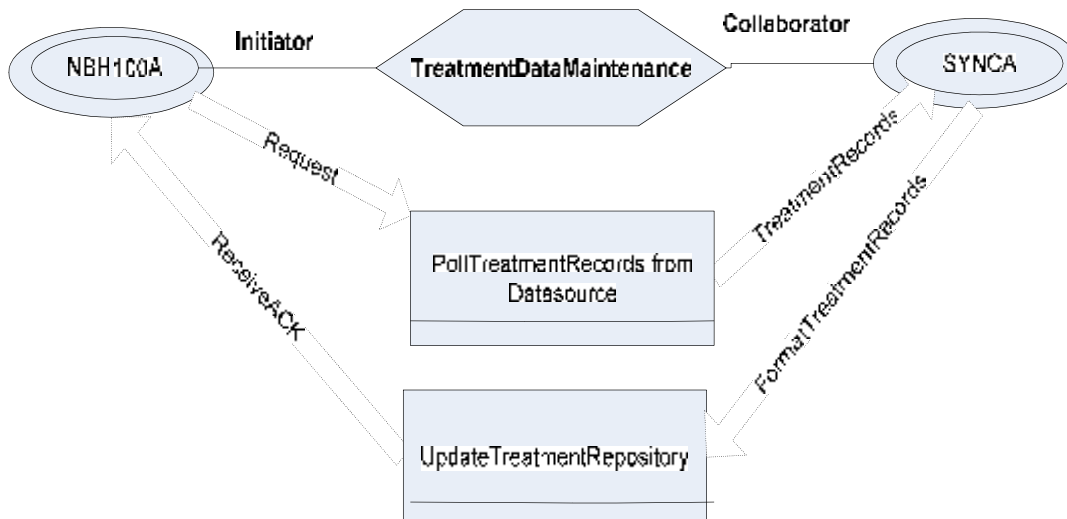


S

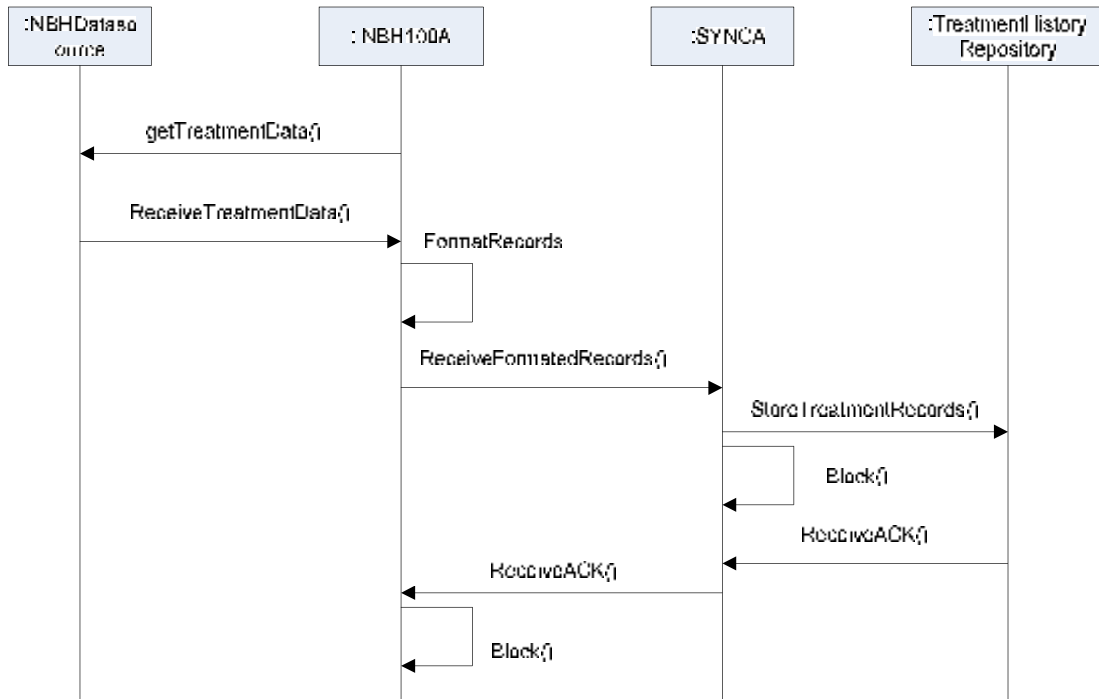
4.43 Sequence diagram between KMSA and SYNCA



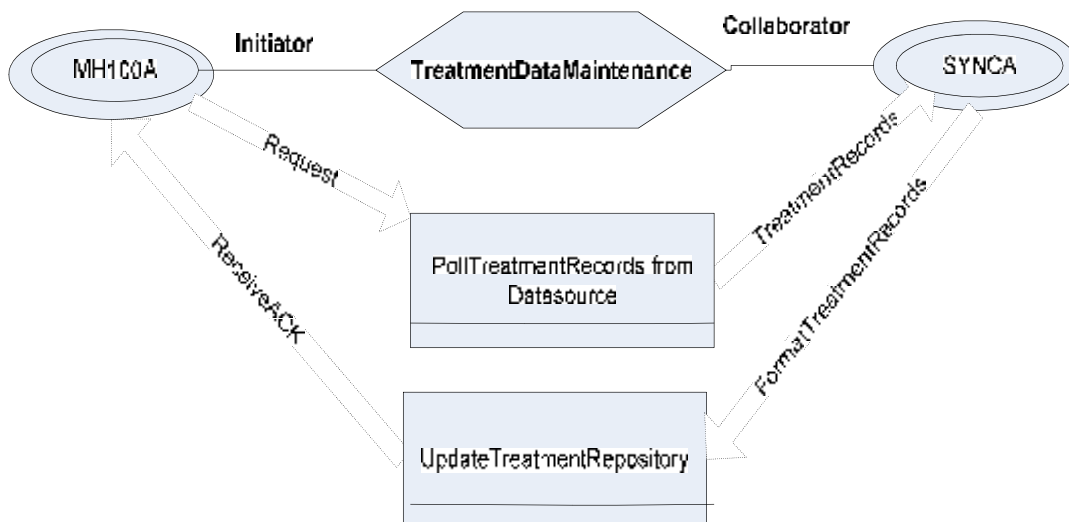
4.44 Interaction diagram between NBH100A and SYNCA



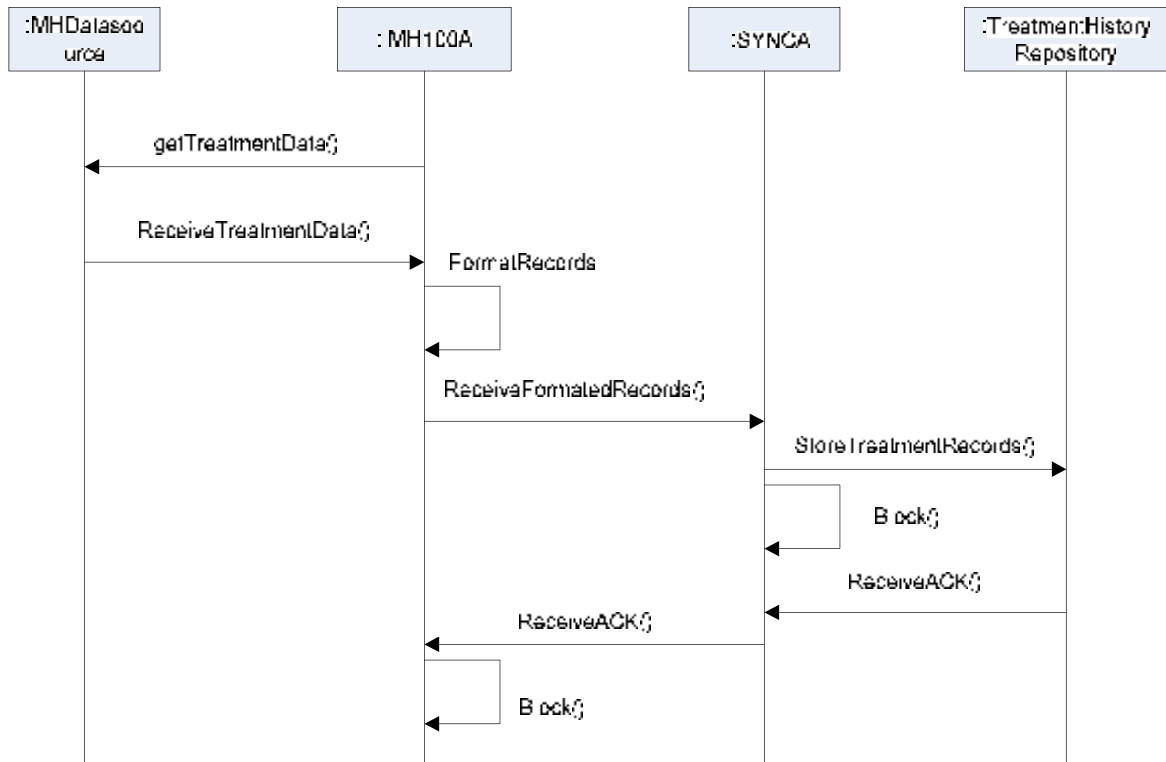
4.45 Sequence diagram between NBH100A and SYNCA



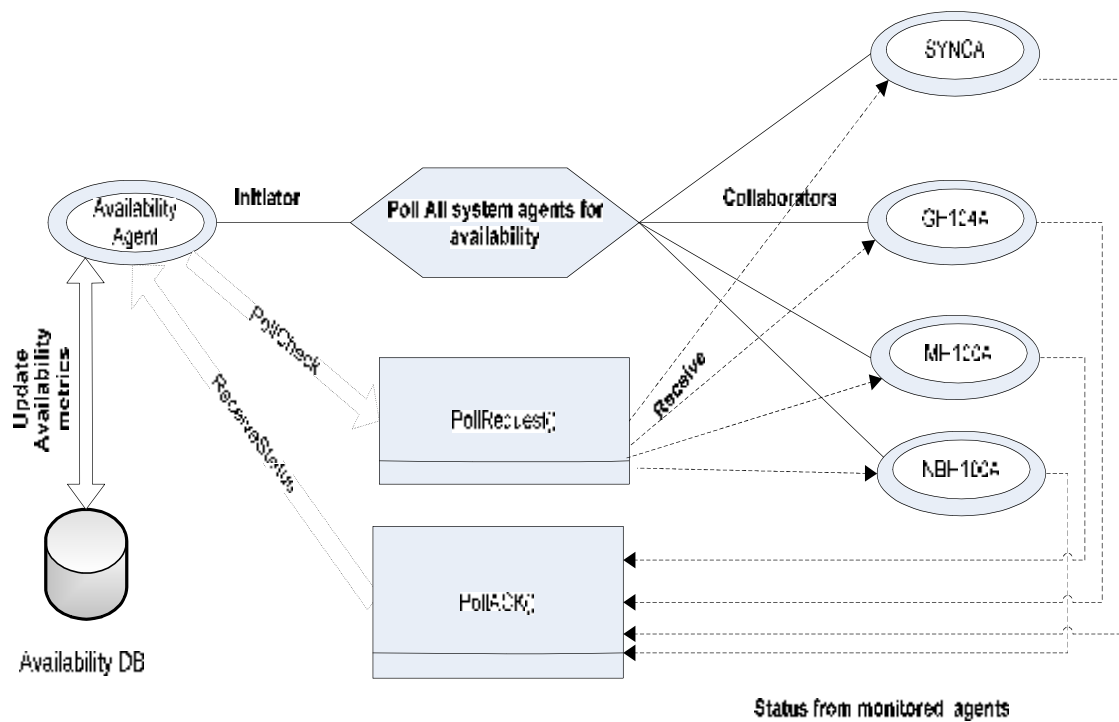
4.46 Interaction diagram between MH100A and SYNCA



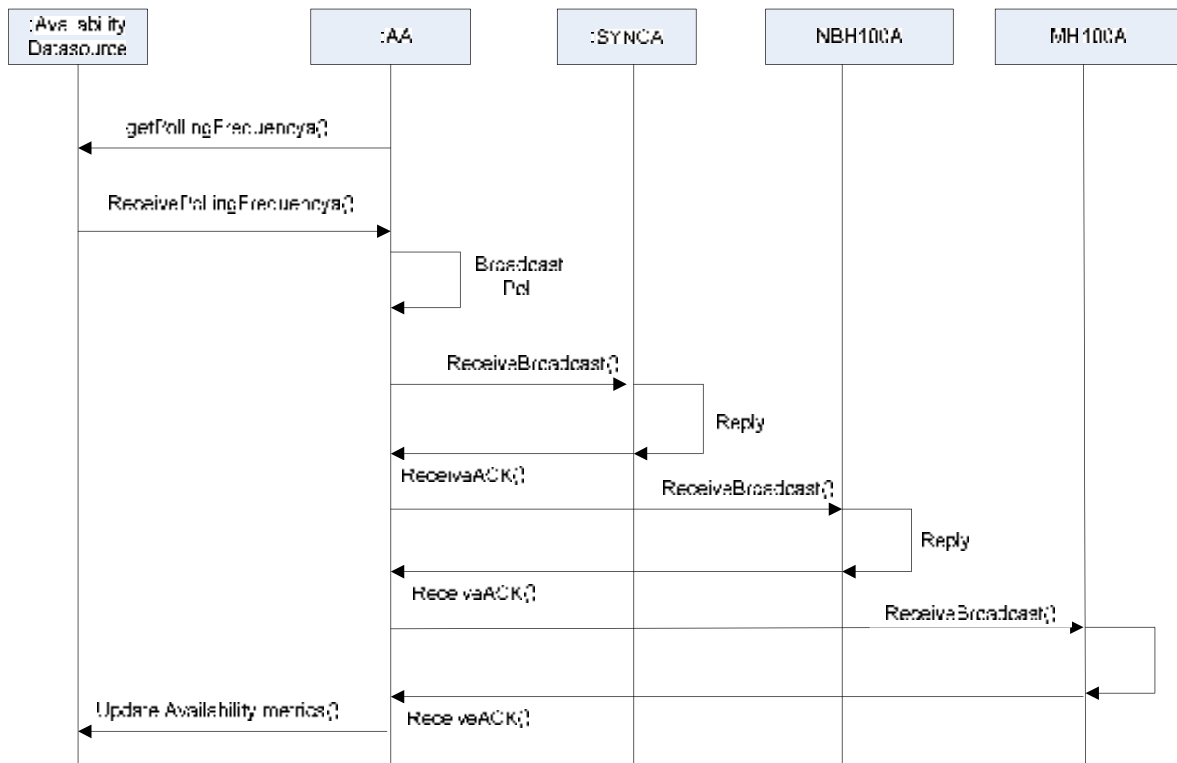
4.47 Sequence diagram between MHH100A and SYNCA



4.48 Interaction diagram between AA, SYNCA, GH104A, MH100A and NBH100A



4.49 Sequence diagram between AA, SYNCA, NBH100A and MHH100A



4.5 Prototype implementation, testing and evaluation

4.51 Virtualization and Web services

The Host PC runs Windows XP 32 bit, onto which VMware workstation 32bit has been installed. The four virtual machines and a virtual LAN are setup and configured with static IP addresses on a class C network addressing scheme 192.168.0.0, mask /24 .The web server is based on Apache 2.x and PHP 5.X as scripting language and housing a MySQL 5.0 database server. The database gets update from the KRA medical system which is based on MsSQL Server, and only the required fields are replicated.

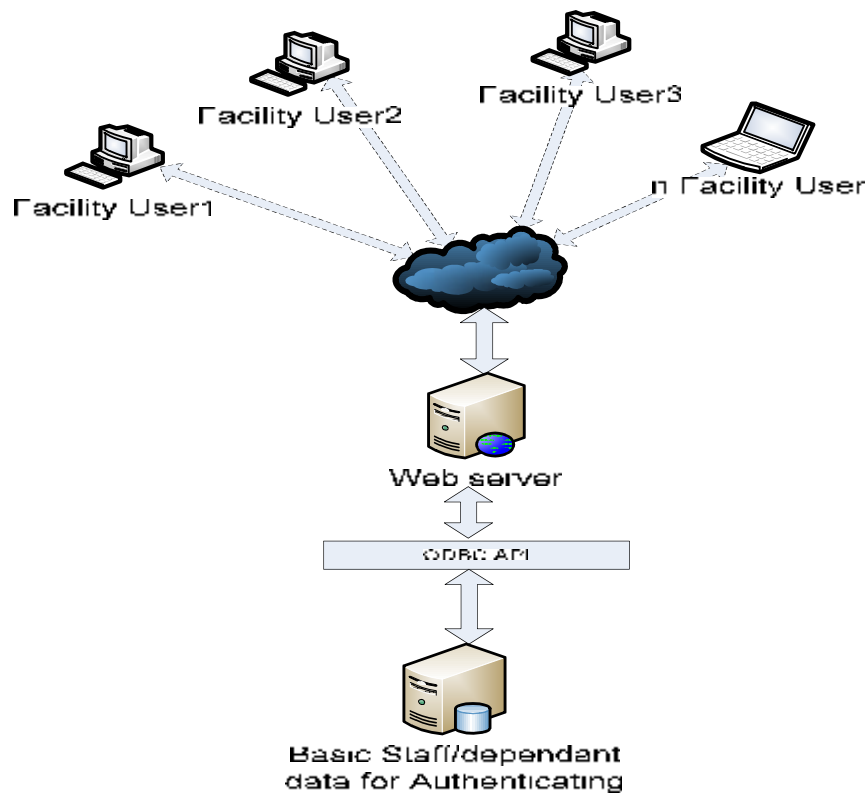


Figure 5 : Online Validation Architecture

4.52 Virtual environment design

This step is necessary for enabling testing the model over a distributed environment. The following steps were used in creating the model.

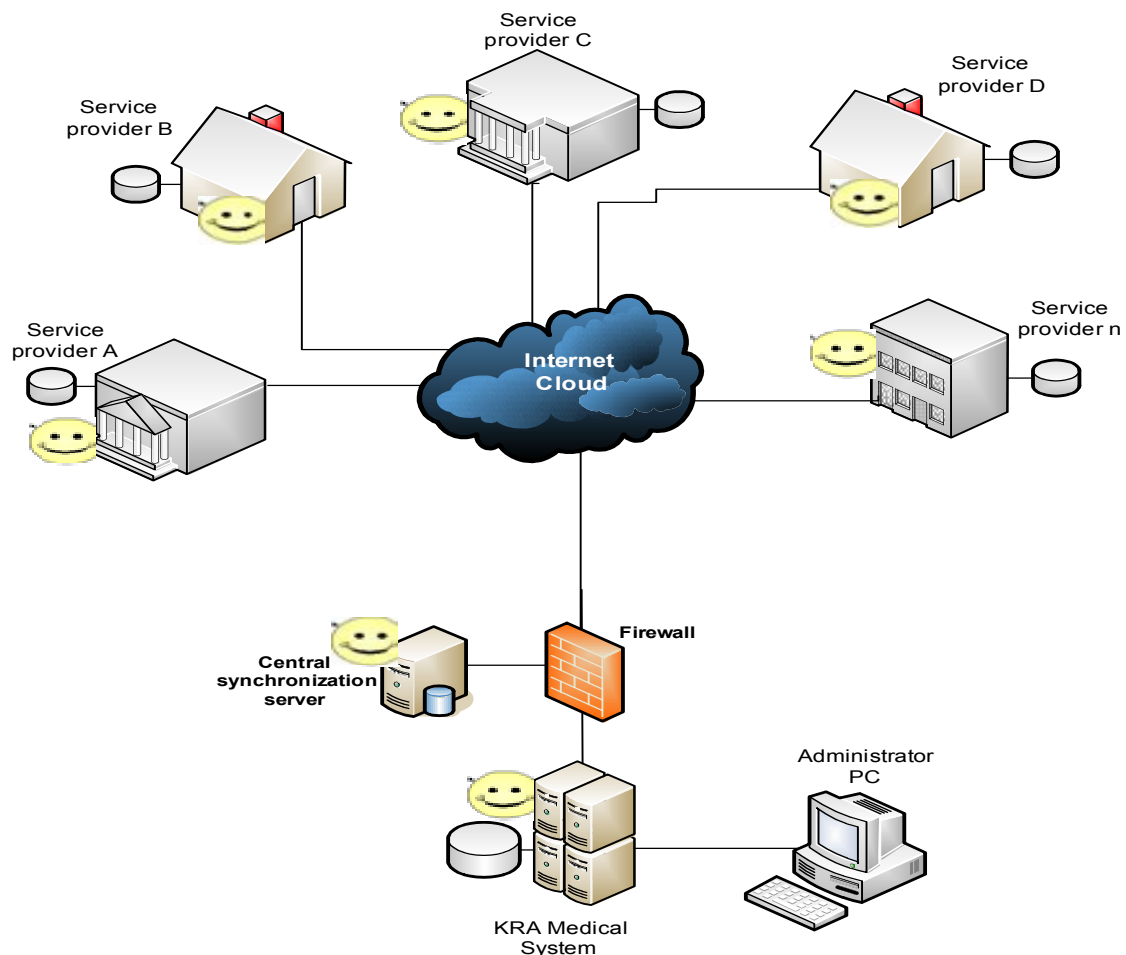
- a) VMware workstation was installed on top of the host operating system (Windows XP). Then four VMs are created
- b) Each facility has an agent responsible for picking up the treatment records and relaying to the synchronization agent for historical purposes. This means that only a single JVM is needed per host.
- c) Communications between JVMs is done automatically in JADE using RMI. The agent container is an RMI server object that manages a set of agents. Each agent gets one thread. The agent container handles the routing of messages.
- d) The agent platform provides a GUI for the remote management, monitoring and controlling of the status of agents. This GUI is implemented as an agent, the Remote Monitoring Agent (RMA).

The *Microsoft Loopback adapter* has been configured to ensure that the host PC can communicate with a virtual network environment where network access is not available.

4.53 Testing and Evaluation

This was carried out using data from the hospitals. INSERT, UPDATE and DELETE data manipulation commands (DML) are used to manage the new and old data. Specifically, the treatment history from the facility agents to the synchronization server (housing the SYNCA) is considered as new data /records, hence the instruction to SYNCA must be an INSERT DML command. The data from the KRA medical system (housing the KMSA) to the SYNCA can take any form of the above DML operation. INSERT (for example for new staff members/dependants), UPDATE (for example the routine maintenance operations such as data corrections, addition of spouse and dependants or filling prior data omissions).

Figure 6 : Infrastructure Diagram



The treatment warehouse (repository with all those treated from the facilities) is only updated by the SYNCA (synchronization agent). It has information such as invoice no and amount, staff/beneficiary full names, nature of treatment, invoicing facility. When the facility agents wish

to send data to SYNCA, they first consult the CPM (Central Parameter Management) to learn about the polling frequency each should use.

The tables below shows the data definitions for selected facility/medical service provider data sources.

Table 1: Mater Hospital data source

<i>serialno</i>	<i>Invoice No</i>	<i>Invoice Amount</i>	<i>Invoice eDate</i>	<i>Invoice Status</i>	<i>InvDateSent</i>	<i>patientId</i>	<i>PayerAccount</i>	<i>Payer Name</i>	<i>Staff fref</i>	<i>Dependantref</i>	<i>Firstname</i>	<i>Middlename</i>	<i>Lastname</i>
14	BKX42223	21,500.00		Not Sent		4322118	M3009	KRA	6602	400016	James		Waweru
18	BKX43254	7,800.00		Not Sent		3345321	M3009	KRA	8201	400021	Winfred		Melisa

Table 2: Nairobi Hospital data source

<i>Recno</i>	<i>NBHNo</i>	<i>VoucherNo</i>	<i>Voucher amount</i>	<i>Voucher date</i>	<i>Voucher Status</i>	<i>Voucher Datesent</i>	<i>CorporateID</i>	<i>CorporateName</i>	<i>Staff Id</i>	<i>DependantId</i>	<i>First name</i>	<i>middlename</i>	<i>lastname</i>
2	246732	OPR90003	7,920.20	27/03/2013			N1020	Kenya Revenue Authority	7721		Caroline	Onyango	Wacuka
4	489345	OPR90028	6,724.00	16/04/2013			N1020	Kenya Revenue Authority	6602		James		Waweru

Table 3: Getrude Hospital data source

<i>RecId</i>	<i>Patientno</i>	<i>memberref</i>	<i>Beneficiary</i>	<i>InvoiceRef</i>	<i>Amountdue</i>	<i>Invoice Date</i>	<i>surname</i>	<i>Firstname</i>	<i>Lastname</i>	<i>Company</i>	<i>CompanyRef</i>
16	421323	8880	400020	426932	6,800.00		Kioko	Titus	Mwania	KRA	6342

Table 4: Agent Responsibilities

NO	AGENT NAME	TASK TO DO	AGENT RUN MODE
1	KMSA	<p>When it first starts, it communicates with SYNCA agent to ensure that the total count of records in the staff/beneficiary object (at the KRA medical system server) equals that found at integration server, which houses the SYNCA agent.</p> <p>If the two datasets contain the same number of records, what follows after is to monitor INSERTS,UPDATES and DELITIONS from the master(staff/beneficiary) and inform the SYNCA agent to update the same in the dependent table. Data picked is based on the current date and if the status field is 'pending' .</p>	24/7 (but also depending on the parameters sent by the Admin).
2	SYNCA	<p>It should be the first agent to start in the platform.</p> <p>It is responsible for receiving requests from the facility agents and updating the treatment history residing at the Synchronization server. It uses filters to inform the next course of action i.e. if the message is from KMSA then it is considered to be an INSERT,UPDATES or DELITION from the master (staff/beneficiary) data source and this needs to be reflected in the staff/dependant repository since it is the source the facilities use to authenticate if the patients belong to the KRA scheme.</p>	24/7 (but also depending on the parameters sent by the Admin).
3	AA	The Availability/Monitoring agent runs continuously to check and log the uptime of all other agents for the review by the auditor. It should try three times to contact a non responsive agent before declaring it dead/out of service.	24/7 (running mode can also be based on parameters set by the Admin).
4	GCH104A	Getrudes Children Hospital Agents task include formatting/ typecasting and relay a message to the SYNCA agent requesting it to update the treatment history.	24/7 (but also depending on the parameters sent by the Admin).
5	MH100A	Mater Hospital Agent routinely picks the treatment records from the said Facility, does the formatting /typecasting and then relays a message to the SYNCA agent requesting it to update the treatment history with the records it found.	24/7 (but also depending on the parameters sent by the Admin).
6	NBH100A	Nairobi Hospital Agents task include formatting /typecasting and then relaying a message to the SYNCA agent requesting it to update the treatment history.	24/7 (but also depending on the parameters sent by the Admin).
8	nA	nFacility agents can be included into the infrastructure since the design is very modular	24/7 as above.

4.6 System Resources Required

The execution of the project will need resources such as hardware, software, service and stationery.

4.61 Hardware Requirement: This will include a Laptop computer (the least of 4GB RAM, 2.5Ghz Duo core CPU), Wireless modem, 10 GB Flash disk and empty DVD for backup measures.

4.62 Software Requirement: This will include a Laptop computer (the least of 4GB RAM, 2.5Ghz Duo core CPU), Wireless modem, 10 GB Flash disk and empty DVD for backup measures.

- a) Microsoft Windows XP OS and Microsoft Windows Office
- b) VM Ware, the virtualization Environment
- c) JDK/JRE – Java Development Kit/Java Runtime Environment
- d) FOSS – Free Open Source Software i.e. Ubuntu and Centos 5.8
- e) Database Management System i.e. Oracle , Mysql and MS SQL Server 2005
- f) Apache Tom cat web Server and PHP 5.x
- g) Net Beans IDE 6.9.1
- h) Java enabled web browser
- i) Jasper and crystal Reporter

4.63 Agents implementation

The agent classes as defined in the analysis chapter were implemented using java and the JADE framework. The agents are started and tested in unit to see that they can exchange messages in a controlled manner. All the system agents are then tested to gauge if objectives are being attained.

4.64 Database implementation

This entailed installing MS SQL server, Oracle, Mysql and Postgresql, which are the most common databases in use today, which posed as the heterogeneous datasources.

4.65 Virtual LAN implementation

This was configured in VMware and a virtual network configured in the planned subnet. Virtual machines were then added into the virtual network each with an IP address in 192.168.0.0/24 network.

4.66 Summary of Prototype Implementation

The implementation consisted of the heterogeneous databases representing various facilities/hospitals, web server, the front end that was developed in PHP with a Mysql

database, Jade agents including KMSA,GCH104A,SYNCA,NBH100A and MH100A. In addition we had a virtual machine installed with Jasper and Crystal reports applications. The SQLyog database access tool was used for manipulating related backend data sources for Mysql data sources while Toad was used for Oracle and MS SQL server data sources.

CHAPTER FIVE: PROTOTYPE TESTING AND VALIDATION

5.1 Introduction

Testing was carried out to identify logical bugs and to conform that input is transformed into output as planned.

- ✓ *Tools used:* Remote Monitoring Agent (RMA), Sniffer Agent, database analysis tools
- ✓ *Experiment objective:* To verify that the prototype meets the system objectives.

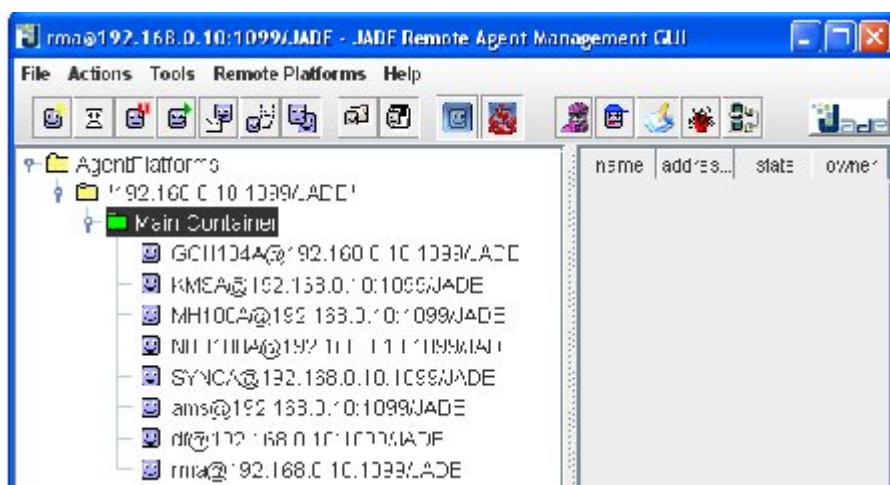
5.2 Results Analysis (Input, Processing and Output)

The analysis of the research project objectives is as shown below including explanations on how each objective was fulfilled. The testing steps are highlighted below:

We first start the JADE agents SYNCA, KMSA, GCH104A, NBH100A and MH100A to ensure that the facilities can interact. The Sniffer agent is also activated so that we can monitor message exchange between the agents. The polling parameters need be set by configuring using the CPM We then introducing a staff member or beneficiary into the scheme via INSERT DML command at the kra_medical database, and then validate the same over the web to confirm that KMSA actually receives the updates and post to the kra_central database.UPDATE DML command can also be utilised to edit the staff member or beneficiary data already contained in kra_medical database. And the same updates should be reflected at the kra_central database and be viewable over the web where applicable, i.e if an account is disabled.

5.21 How System Objectives were achieved

We start by activating the System agents (KMSA, SYNCA, NBH100A, MH100A and GCH104A)



The screen shot below shows the system agents in action polling the data sources for new records that may not have been transmitted. The polling duration is variable defined or configured at the CPM.

```

MH100A Polling for new records from Matter Hospital in : 2000 Seconds
NDH100A Polling for new records from Nairobi Hospital in : 2540 Seconds
MH100A Polling for new records from Matter Hospital in : 2000 Seconds
KMSA Polling for new/updates from Medical System in : 2555 Seconds
NDH100A Polling for new records from Nairobi Hospital in : 2540 Seconds
GCH104A Polling for new records from Getrude's Hospital in : 2888 Seconds
MH100A Polling for new records from Matter Hospital in : 2000 Seconds
KMSA Polling for new/updates from Medical System in : 2555 Seconds
NEH100A Polling for new records from Nairobi Hospital in : 2540 Seconds
MH100A Polling for new records from Matter Hospital in : 2000 Seconds
GCH104A Polling for new records from Getrude's Hospital in : 2888 Seconds
KMSA Polling for new/updates from Medical System in : 2555 Seconds
NDH100A Polling for new records from Nairobi Hospital in : 2540 Seconds
MH100A Polling for new records from Matter Hospital in : 2000 Seconds
GCH104A Polling for new records from Getrude's Hospital in : 2000 Seconds
MH100A Polling for new records from Matter Hospital in : 2000 Seconds
KMSA Polling for new/updates from Medical System in : 2555 Seconds

```

5.22 Introducing a Staff/dependant into the scheme

The scripts below illustrate the introduction of Staff or the dependant into the system , however it is the function of the responsible application that is meant to execute these DML commands and not to be done directly into the database.

```
insert into staff values (7070,'James', 'Mutembei','Nteere','Male','1979-04-03','P.O Box 91675','Meru',0720063432,'Active');
```

```
insert into beneficiary values (400032,'Tanya','Nkatha', 'Nteere','2013-01-12','Female',7070,'Daughter','Active');
```

StaffID	StaffName	StaffMn...	StaffLn...	StaffGen...	StaffDob	StaffBoxNo	StaffTown	StaffMobile	StaffStatus
5500	Martha	Njuku	Kiuku	Female	12/6/1984 12:00:00 AM	F.O Box 2545	Gilguri	721122111	Active
5580	Suzan	Wabene	Makau	Female	3/15/1985 12:00:00 AM	F.O Box 45678	Karatina	772254799	Active
6830	Christina	Kosisi	Mugano	Female	7/11/1982 12:00:00 AM	F.O Box 12273	Malaba	722755132	Current
0201	Raphael	Mwaniki	Njeru	Male	6/5/1977 12:00:00 AM	F.O Box 45670	Moyale	777271177	Active
6633	James	Mwangangi	Kariaka	Male	4/6/1969 12:00:00 AM	F.O Box 55763	Muribasa	722452457	Active
5252	Michael	Kashigi	Kituku	Male	3/11/1977 12:00:00 AM	F.O Box 23337	Machakos	733737373	Active
7999	Moses	Magenha	Othman	Male	6/5/1979 12:00:00 AM	F.O Box 00722	Kisii	722747171	Active
7970	Franklin	Keane	Misigo	Male	3/12/1981 12:00:00 AM	F.O Box 64953	Webuye	733375555	Active
7878	Charles	Kipchumba	Fono	Male	6/5/1972 12:00:00 AM	F.O Box 52344	Eldoret	72137423	Active
7710	Andrew	Wamoo	Mwazungu	Male	7/12/1973 12:00:00 AM	F.O Box 64475	Narok	723135511	Active
0310	Abel	Mwanja	Kimondo	Male	4/11/1960 12:00:00 AM	F.O Box 010402	Machakos	777777777	Active
6540	Franklin	Sesio	Eel	Male	11/5/1965 12:00:00 AM	F.O Box 12468	Eldoret	733375553	Active
7070	James	Mutembei	Nteere	Male	4/3/1979 12:00:00 AM	F.O Box 91675	Meru	720053432	Active

It is shown above that the record whose **StaffID** is **7070** has been posted into the `kra_medical` database, however the same record does not reflect in the synchronization server if the KMSA agent is not active (which is responsible for updating records from `kra_medical` DB to `kra_central`

DB, which is queried by the facilities/Hospitals during the validation process). The first instance shows the record has been inserted into the DB then updates on the same record can be seen under the *Operation* column and the date when the operation was done is also shown as captured in the log table.

StaffID	StaffNo.	StaffNo.	StaffName	StaffGender	StaffPhone	StaffEmail	StaffAddress	StaffMobile	StaffStatus	Operation	OperDate	OperTime	OperStatus	OperTime
590	Sasa	Waleke	Mesau	Female	9091156	F.O.Ex:49078	Chigusa	73294759	Active	Update	2016/10/20 15:28:47	Send	2016/10/20 15:00:13 AM	
591	Kenke	Mher	Arin	Female	1101156	F.O.Ex:2384	Chimungu	6111311	Active	Update	2016/10/20 15:30:47	Send	2016/10/20 15:00:17 AM	
7978	Alak	Mwale	Kwaku	Male	4010156	F.O.Ex:610402	Makulu	73300082	Active	Insert	2016/10/20 15:35:47	Send	2016/10/20 15:00:59 AM	
810	Ahar	Mwale	Kwaku	Male	1101156	F.O.Ex:610402	Makulu	73300082	Active	Insert	2016/10/20 15:37:47	Send	2016/10/20 15:01:01 AM	
603	Keshae	Mwandi	Mw	Female	6919156	F.O.Ex:49060	Mwale	6339060	Active	Update	2016/10/20 15:41:47	Send	2016/10/20 15:01:04 AM	
6583	James	Mwale	Garale	Male	4019156	F.O.Ex:38750	Mwale	73292467	Active	Update	2016/10/20 15:50:47	Send	2016/10/20 15:01:17 AM	
711	Chadine	Wes	Migoma	Female	6011156	F.O.Ex:4700	Mwale	6710157	Inactive	Update	2016/10/20 15:57:47	Send	2016/10/20 15:01:19 AM	
590	Sasa	Waleke	Mesau	Female	9091156	F.O.Ex:49078	Chigusa	73294759	Active	Update	2016/10/20 16:01:47	Send	2016/10/20 15:00:13 AM	
7970	Kwaku	Migoma	Arin	Male	6019156	F.O.Ex:49078	Arin	73294759	Active	Update	2016/10/20 16:01:47	Send	2016/10/20 15:01:01 AM	
603	Charles	Wanyama	Rongo	Male	6919156	F.O.Ex:49060	Edoek	6339060	Active	Delete	2016/10/20 16:02:47	Send	2016/10/20 15:01:04 AM	
7978	Charles	Wanyama	Rongo	Male	6919156	F.O.Ex:49060	Edoek	73294759	Active	Insert	2016/10/20 16:03:47	Send	2016/10/20 15:00:59 AM	
591	Kenke	Mher	Arin	Female	1101156	F.O.Ex:2384	Chimungu	6111311	Active	Update	2016/10/20 16:04:47	Send	2016/10/20 15:01:07 AM	
590	Mwale	Mwale	Gw	Female	1010156	F.O.Ex:2385	Gwale	7312210	Active	Update	2016/10/20 16:05:47	Send	2016/10/20 15:00:57 AM	
590	Kenke	Mher	Arin	Female	1010156	F.O.Ex:2385	Chimungu	7312210	Active	Update	2016/10/20 16:06:47	Send	2016/10/20 15:00:57 AM	
590	Mwale	Mwale	Gw	Female	1010156	F.O.Ex:2385	Chimungu	7312210	Active	Update	2016/10/20 16:07:47	Send	2016/10/20 15:01:00 AM	
603	Keshae	Mwandi	Mw	Female	6919156	F.O.Ex:49060	Mwale	73292467	Active	Update	2016/10/20 16:08:47	Send	2016/10/20 15:01:05 AM	
711	Andrew	Wes	Migoma	Male	6011156	F.O.Ex:4700	Mwale	6710157	Active	Insert	2016/10/20 16:09:47	Send	2016/10/20 15:01:07 AM	
770	Mwale	Waleke	Mwale	Male	7020156	F.O.Ex:6448	Mwale	73292467	Active	Update	2016/10/20 16:10:47	Send	2016/10/20 15:00:59 AM	
590	Sasa	Waleke	Mesau	Female	9091156	F.O.Ex:49078	Chigusa	73294759	Active	Update	2016/10/20 16:11:47	Send	2016/10/20 15:00:59 AM	
603	Keshae	Mwandi	Mw	Female	6919156	F.O.Ex:49060	Mwale	73292467	Active	Update	2016/10/20 16:12:47	Send	2016/10/20 15:00:59 AM	
6940	Franklin	Wes	Wes	Male	1101156	F.O.Ex:2385	Edoek	73292467	Active	Insert	2016/10/20 16:13:47	Send	2016/10/20 15:01:06 AM	
711	Andrew	Wes	Migoma	Male	6011156	F.O.Ex:4700	Mwale	6710157	Active	Update	2016/10/20 16:14:47	Send	2016/10/20 15:01:07 AM	
770	Mwale	Waleke	Mwale	Male	7020156	F.O.Ex:6448	Mwale	73292467	Active	Update	2016/10/20 16:15:47	Send	2016/10/20 15:00:59 AM	
7970	James	Mwale	Wes	Male	4019156	F.O.Ex:38750	Arin	73294759	Active	Insert	2016/10/20 16:16:47	Send	2016/10/20 15:01:17 AM	
7970	James	Mwale	Wes	Male	4019156	F.O.Ex:38750	Arin	73294759	Disabled	Update	2016/10/20 16:17:47	Send	2016/10/20 15:00:59 AM	
7970	James	Mwale	Wes	Male	4019156	F.O.Ex:38750	Arin	73294759	Active	Update	2016/10/20 16:18:47	Send	2016/10/20 15:00:59 AM	

The record 400032 which is associated with staff 7070 (the KRA staff member) is shown below as being posted in the database and below the said table the corresponding entry as captured in the log stable showing transaction details as captured by the triggers.

BeneficiaryID	BeneficiaryName	BeneficiaryPhone	BeneficiaryEmail	BeneficiaryDOL	BeneficiaryGender	Staff ID	BeneficiaryType	BeneficiaryStatus
400007	David	Wanyama	Kwale	2011/19871210	Male	7070	Spouse	Active
400008	Claire	Wanyama	Rongo	8/20/2009 12:00:00	Female	7999	Daughter	Active
400009	Martin	Kwale	Kwale	2/25/2006 12:00:00	Male	4011	Spouse	Active
400020	Dore	Kwale	Mwale	8/19/2007 12:00:00	Female	5699	Daughter	Disabled
400021	Abraham	Mwale	Mwale	6/16/2010 12:00:00	Male	6262	Spouse	Active
400022	Eedule	Mwale	Kwale	7/22/1984 12:00:00	Female	3801	Spouse	Active
400023	Kwale	Kwale	Rongo	8/20/2002 12:00:00	Female	6262	Daughter	Active
400027	Dore	Mwale	Mwale	6/16/2010 12:00:00	Female	7970	Daughter	Active
400029	Christopher	Mwale	Kwale	11/3/2005 12:00:00	Male	7070	Spouse	Active
400026	Amy	Mwale	Mwale	4/9/2009 12:00:00	Female	5292	Spouse	Active
400028	Enan	Mwale	Mwale	11/6/2008 12:00:00	Male	5262	Spouse	Active
400029	Rosemary	Mwale	Kwale	4/7/2011 12:00:00	Female	5000	Daughter	Active
400030	Grace	Mwale	Mwale	6/1/2012 12:00:00	Female	3810	Daughter	Active
400001	Miriam	Kwale	Kwale	01/01/2010 12:00:00	Female	7999	Daughter	Active
400032	Tanya	Kwale	Wes	11/2/2013 12:00:00	Female	7070	Daughter	Active

The log table entry below shows the type of operation done (INSERT) as well as the time it was done, on 7/25/2013 9.33.09 AM and the operator was *mas*, and that the record was sent or transferred to kra_central database on 7/25/2013 11.08.26 AM.

Benef...	Benefic...	Benefic...	Benefic...	Benefic...	Benef...	Oper...	Oper...	Operati...	Operati...	Operati...	Operati...	Operati...
40024	Mikem	Kanongo	Niguru	11/03/2005	Male	7954	Son	Active	Insert	mas	7/25/2013 9:33:09 AM	7/25/2013 11:08:26 AM
40025	Jenna	Suswa	Staki	01/02/2019	Female	3211	Daughter	Active	Update	mas	7/25/2013 11:08:26 AM	7/25/2013 11:08:26 AM
40026	Mikem	Kanongo	Niguru	11/03/2005	Male	7954	Son	Active	Update	mas	7/25/2013 11:08:26 AM	7/25/2013 11:08:26 AM
40027	Mikem	Kanongo	Niguru	11/03/2005	Male	7954	Son	Active	Update	mas	7/25/2013 11:08:26 AM	7/25/2013 11:08:26 AM
40028	Mikem	Kanongo	Niguru	11/03/2005	Male	7954	Son	Active	Delete	mas	7/25/2013 11:08:26 AM	7/25/2013 11:08:26 AM
40029	Mikem	Kanongo	Stallins	11/03/2005	Male	7954	Son	Active	Insert	mas	7/25/2013 11:08:26 AM	7/25/2013 11:08:26 AM
40030	Mikem	Kanongo	Niguru	11/03/2005	Male	7954	Son	Active	Insert	mas	7/25/2013 11:08:26 AM	7/25/2013 11:08:26 AM
40031	Mikem	Mwai	Niguru	11/03/2005	Male	7954	Son	Active	Insert	mas	7/25/2013 11:08:26 AM	7/25/2013 11:08:26 AM
40032	Philippe	Mwai	Niguru	11/03/2005	Male	7954	Son	Active	Insert	mas	7/25/2013 11:08:26 AM	7/25/2013 11:08:26 AM
40033	Audrey	Waguru	Njugus	4/9/2009	Female	3222	Son	Disabled	Insert	mas	7/25/2013 11:08:26 AM	7/25/2013 11:08:26 AM
40034	Jhar	Luoba	Nobuzi	11/03/2005	Male	3222	Son	Active	Insert	mas	7/25/2013 11:08:26 AM	7/25/2013 11:08:26 AM
40035	Rosemary	Wamuzi	Nagutho	4/7/2011	Female	3300	Daughter	Active	Insert	mas	7/25/2013 11:08:26 AM	7/25/2013 11:08:26 AM
40036	Abraham	Mudine	Nobuzi	01/03/2010	Male	3222	Son	Active	Update	mas	7/25/2013 11:08:26 AM	7/25/2013 11:08:26 AM
40037	Rosemary	Wamuzi	Nagutho	4/7/2011	Female	3300	Daughter	Disabled	Update	mas	7/25/2013 11:08:26 AM	7/25/2013 11:08:26 AM
40038	Dore	Mudine	Mwondo	01/03/2012	Female	3300	Daughter	Active	Update	mas	7/25/2013 11:08:26 AM	7/25/2013 11:08:26 AM
40039	Shahin	Enay	Enayr	8/20/2002	Female	3222	Daughter	Active	Insert	mas	7/25/2013 11:08:26 AM	7/25/2013 11:08:26 AM
40040	Miriam	Kisumu	Muguri	3/18/2010	Female	3222	Daughter	Active	Insert	mas	7/25/2013 11:08:26 AM	7/25/2013 11:08:26 AM
40041	Miriam	Kisumu	Njiri	3/18/2010	Female	3222	Daughter	Active	Insert	mas	7/25/2013 11:08:26 AM	7/25/2013 11:08:26 AM
40042	Philippe	Mwai	Enayr	11/03/2005	Male	3222	Son	Active	Insert	mas	7/25/2013 11:08:26 AM	7/25/2013 11:08:26 AM
40043	Jenna	Suswa	Staki	01/02/2019	Female	3211	Daughter	Active	Update	mas	7/25/2013 11:08:26 AM	7/25/2013 11:08:26 AM
40044	Audrey	Waguru	Njugus	4/9/2009	Female	3222	Son	Active	Update	mas	7/25/2013 11:08:26 AM	7/25/2013 11:08:26 AM
40045	Yenir	Kadi	Ndertu	4/23/2006	Male	3211	Son	Active	Update	mas	7/25/2013 11:08:26 AM	7/25/2013 11:08:26 AM
40046	Rosemary	Wamuzi	Nagutho	4/7/2011	Female	3300	Daughter	Active	Update	mas	7/25/2013 11:08:26 AM	7/25/2013 11:08:26 AM
40047	Rosemary	Wamuzi	Nagutho	4/7/2011	Female	3300	Daughter	Active	Update	mas	7/25/2013 11:08:26 AM	7/25/2013 11:08:26 AM
40048	Ther	Mudine	Nobuzi	11/03/2010	Female	3222	Daughter	Active	Insert	mas	7/25/2013 11:08:26 AM	7/25/2013 11:08:26 AM

At the synchronization server, the database (mysql) which the facilities/Hospitals access has been updated with both records (staff and beneficiary) which were captured in the Medical database system (based on MS SQL). The responsible agent that does this updates is the KMSA agent.

StaffID	Staff name	Staff name	Staff name	Staff dob	Staff gender	Staff home	Staff phone	Staff mobile	Staff status
5338	M. Auel	Kassiku	Eladua	1977-03-11	Male	P.O. Box 23384	Kisumu	788787878	Active
5500	Martha	Kakko	Kakko	1964-11-26	Female	P.O. Box 114	Chungu	721221111	Active
5580	Suzan	Kacko	Kacko	1965-03-15	Female	P.O. Box 45578	Kisumu	772254700	Active
6540	Franklin	Besilo	Besilo	1965-11-26	Male	P.O. Box 13468	El Dorado	799376593	Active
6600	John	Phongor	Kwenge	1964-11-11	Male	P.O. Box 1111	Phongor	722464111	Active
6830	Christine	Kasilo	Mugere	1963-07-10	Female	P.O. Box 47778	Kisumu	782758432	Disabled
7000	Rose	Kutebui	Otebo	1973-04-26	Female	P.O. Box 51575	Kisumu	780355432	Active
7011	Andrew	Kwenge	Phongor	1964-11-11	Male	P.O. Box 4444	Kisumu	722464111	Active
7378	Charles	Kwenge	Besilo	1973-06-26	Male	P.O. Box 53344	El Dorado	721374333	Active
7970	Franklin	Kasilo	Bischof	1961-07-18	Male	P.O. Box 04000	Kisumu	790070555	Active
7990	Walter	Phongor	Phongor	1973-07-10	Male	P.O. Box 43032	Kisumu	722464111	Active
8200	Regina	Kasilo	Otebo	1965-06-26	Female	P.O. Box 45380	Kisumu	782357777	Active
0010	Abadi	Kwenge	Kwenge	1963-04-11	Male	P.O. Box 11444	Kisumu	721000522	Active

The Beneficiary record from the kra_medical database is now reflected as Dependant record in the Synchronization server, that houses kra_central database as shown in the snap shot below.

Dependant ID	DependantName	DependantEthnic	DependantSex	DependantDOB	DependantGender	State	DependantType	DependantStatus
<input type="checkbox"/>	400007 David	Kisumu	Male	1997-07-21	Male	8833 Sec	Spouse	Active
<input type="checkbox"/>	400008 Lucas	Kisumu	Male	2007-08-20	Female	7677 Daughters	Spouse	Active
<input type="checkbox"/>	400009 Kevin	Kisumu	Female	2000-04-20	Male	0201 Sec	Spouse	Active
<input type="checkbox"/>	400010 Irene	Nakuru	Female	2007-07-19	Female	0777 Daughters	Disabled	Active
<input type="checkbox"/>	400021 Akasha	Nakuru	Female	2000-06-15	Male	2252 Sec	Spouse	Active
<input type="checkbox"/>	400022 Beatrice	Nakuru	Female	1984-07-22	Female	8201 Spouse	Spouse	Active
<input type="checkbox"/>	400023 Esab	Nakuru	Female	2000-08-20	Female	2252 Daughters	Spouse	Active
<input type="checkbox"/>	400024 Kristoferson	Nakuru	Male	2005-11-08	Male	7677 Sec	Spouse	Active
<input type="checkbox"/>	400025 Andy	Nakuru	Female	2000-04-09	Male	0252 Sec	Spouse	Document
<input type="checkbox"/>	400027 Dennis	Nakuru	Female	2009-07-16	Female	0577 Daughters	Spouse	Active
<input type="checkbox"/>	400028 Eileen	Nakuru	Female	2000-11-05	Male	2252 Sec	Spouse	Active
<input type="checkbox"/>	400029 Rosemary	Nakuru	Female	2000-04-07	Female	8833 Daughters	Spouse	Active
<input type="checkbox"/>	400030 Grace	Nakuru	Female	2000-06-01	Female	8833 Daughters	Spouse	Active
<input type="checkbox"/>	400031 Elizabeth	Nakuru	Female	2000-08-18	Female	7677 Daughters	Spouse	Active
<input type="checkbox"/>	400032 Danya	Nakuru	Female	2000-01-18	Female	7677 Daughters	Spouse	Active

5.23 Validating or Authenticating KRA Staff

Now that the staff/dependant has been introduced to Authenticating staff/dependants using the online service after KRA staff or beneficiary has gone to a facility/hospital to seek medical attention.

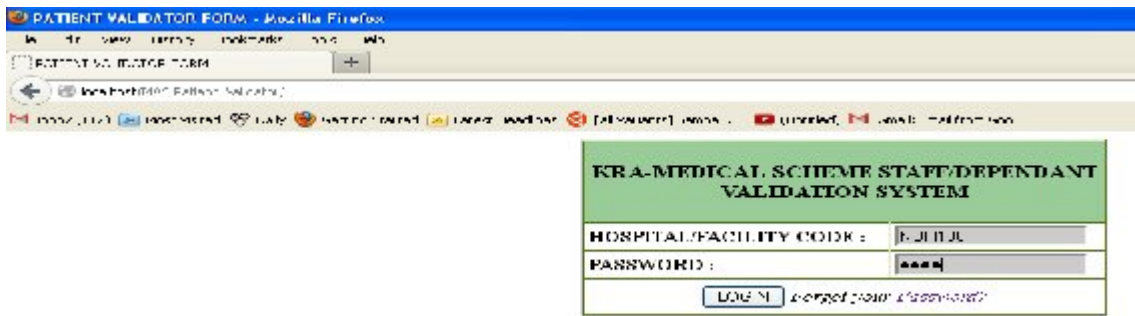
Inputs to the interface which the users at the facility/hospital Administration office

The screen shot below shows the interface validating to ensure that input details are supplied by the facility/Hospital users during staff/dependants validation process.



The Hospital/Facility name we choose to use is: *NBH100* (Nairobi Hospital), while the Password is managed at the KRA side, new passwords are generated automatically as defined in the KRA password rotation policy and relayed via email.

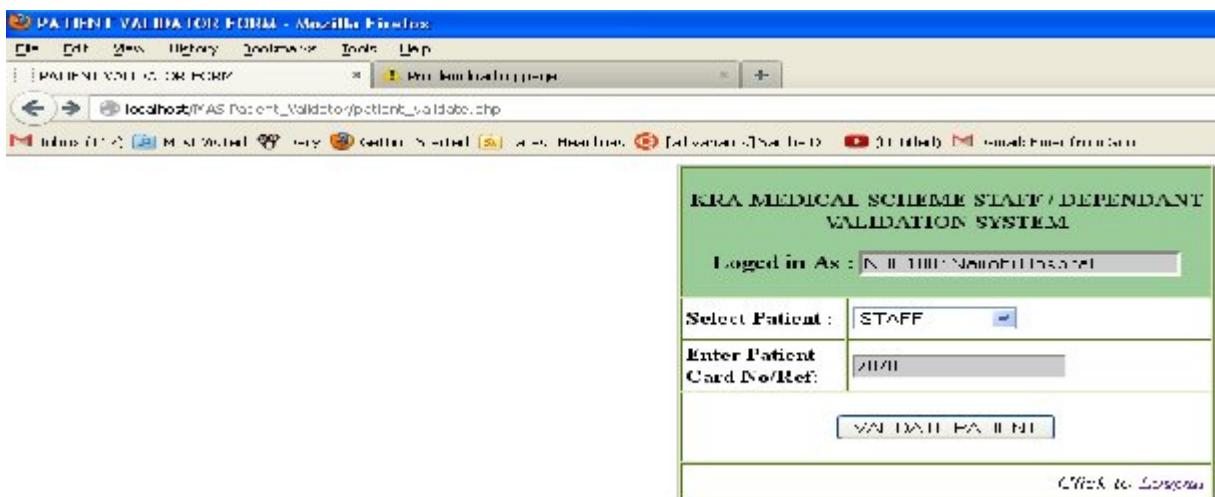
The required data has now been inserted into the capturing input boxes and the facility/Hospital user can now login.



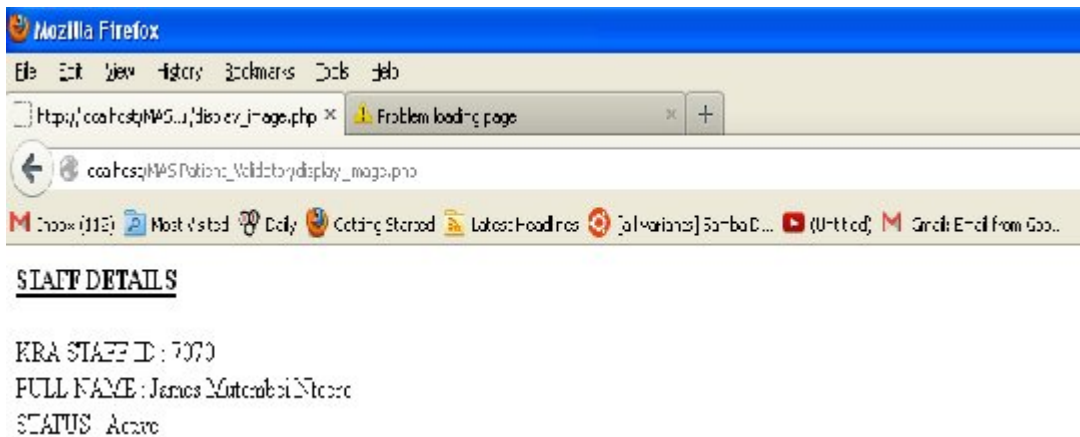
The facility/Hospital user has successfully logged in and the *Select Patient* option will determine the kind of patient being dealt with.



Validating the Staff ID: 7070 to see if he is an employee of KRA as well as the status of his account. (if eligible for treatment).



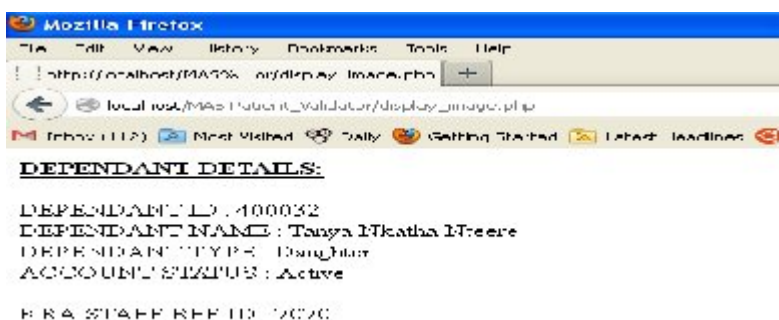
The queried details (if the system is accessed from Nairobi Hospital) of the staff member with ID 7070 is shown below.



5.24 Validating or Authenticating Dependant



The results of the above patient (dependant) being validated to know whether he/she is related to any staff member.



5.25 Sending of invoices from Facility/Hospital to KRA

Since the patient (i.e. the dependant 400032) has been validated and account status found to be Active then treatment process can commence and the following invoice is raised at Nairobi Hospital and the same is relayed to the Central Server (that hosts the invoice database) hosted at KRA.

a) Sample DML command to the Database that would be done by the Nairobi Hospital Medical System.

```
insert into billing values (347444,'OPR916545',34643.90,getdate(),'NotSent','N1020','Kenya Revenue Authority','7070','Staff','Nteere','James','Mutembei');
```

The screen shot below shows the SYNCA (Synchronization Agent) receiving a request message from NBH100A (Nairobi Hospital Agent)to update the invoice table (residing at the synchronization server) in KRA. After SYNCA has performed the request, it informs the requester (NBH100A) that the process has been performed successfully, which then updates the Voucher Status from Not Sent to Sent.

```
NBH100A Polling for new records from Nairobi Hospital in : 2540 Seconds
847444 OPR916545 34643.90 Oct-Sent 2013-07-25 17:26:54.2 Kenya Revenue Authority 7070 Staff James Nteere Mutembei

->SYNCA: NBH100A[52.168.0.10:1099/TCP],Sent the Query :INSERT INTO invoice (InvoiceNo,VoucherNo),Invoice_VoucherAmount,FacilityCo
NBH100A Successfully Connected to KRA Central Server...
GCH104A Polling for new records from Getrude's Hospital in : 2388 Seconds
```

b) Sample DML command to the Database that would be done by Getrude’s Children Hospital Medical System.

```
insert into charges values
(494462,'GA083299',11452.25,getdate(),'NotSent','7070','400032','Tanya','Nkatha','Nteere','5050',
'Kenya Revenue Authority');
```

The screen shot below shows the SYNCA (Synchronization Agent) receiving a request message from GCH104A (Getrude’s Children Hospital Agent) to update the invoice table (residing at the synchronization server) in KRA. After SYNCA has performed the request, it informs the requester (GCH104A) that the process has been performed successfully, which then updates the Invoice status from Not Sent to Sent.

```
NBH100A Polling for new records from Nairobi Hospital in : 2540 Seconds
NBH100A Polling for new records from Matter Hospital in : 3000 Seconds
GCH104A Polling for new records from Getrude's Hospital in : 2358 Seconds
494462 GA083299 11452.25 Oct-26 2013-07-26 17:31:44.06 7070 400032 Mwanze Tanya Nkatha 5050 Kenya Revenue Authority

->SYNCA: GCH104A[52.168.0.10:1099/TCP],Sent the Query :INSERT INTO invoice (InvoiceNo,VoucherNo),Invoice_VoucherAmount,FacilityCode,ResponsibleAgent,Invoice_Stat
GCH104A Successfully Connected to KRA Central Server...
GCH104A Successfully Executed the request on the DB...
```

a) Sample DML command to the Database that would be done by the Matter Hospital Medical System.

insert into invoice values

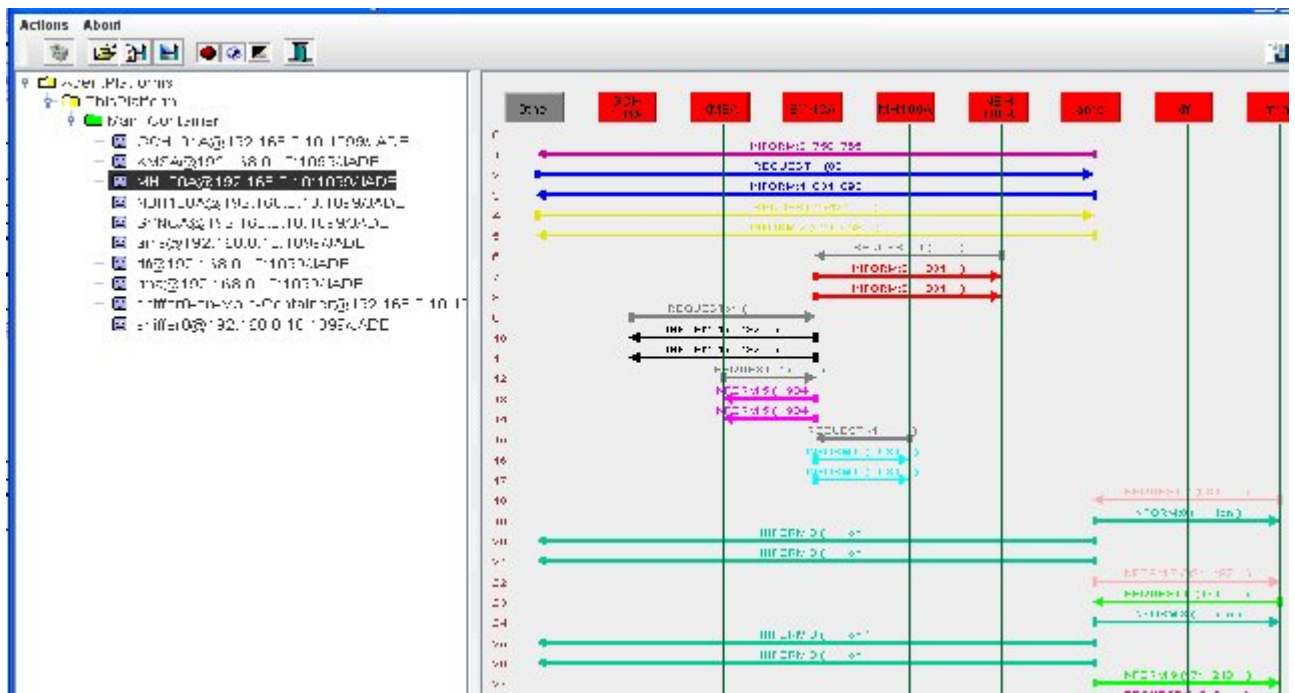
```
('k08228','12834.80',getdate(),'NotSent','871476','M3009','KRA','7070','400032','Tanya','Nkatha','Nteere');
```

The screen shot below shows the SYNCA (Synchronization Agent) receiving a request message from MH100A (Matter Hospital Agent) to update the invoice table (residing at the synchronization server) in KRA. After SYNCA has performed the request, it informs the requester (MH100A) that the process has been performed successfully, which then updates the Invoice status from Not Sent to Delivered.

```
GCHC04 Polling for new records from Gertrude's Hospital in : 2863 Seconds
MH100A Polling for new records from Mater Hospital in : 2000 Seconds
k08228          12834.8 NotSent 871476 M3009 KRA 7070 400032 Tanya Nkatha Nteere

->SYNCA: MH100A@192.168.0.10:1095/JADE,Send the Query :SELECT INTO invoice (InvoiceNo,VoucherNo, FacilityCode, &
MH100A Successfully Connected to Kra_Central Server...
MH100A Successfully Updated the Invoice from Mater Hospital...
```

5.26 System Agents captured using the Sniffer Agent.



5.27 Shutting down agent platform and agent deregistration

```

MH100A Polling for new records from Matter Hospital in : 2000 Seconds
GCH104A Polling for new records from Getrude's Hospital in : 2000 Seconds
GCH104A Polling for new records from Getrude's Hospital in : 2888 Seconds
GCH104A Polling for new records from Getrude's Hospital in : 2888 Seconds
GCH104A-Agent GCH104A@192.168.0.10:1099/JADE Terminating.
Jul 26, 2013 7:16:23 PM jade.core.messaging.MessageManager shutdown
TNF0: MessageManager shutting down
Jul 26, 2013 7:16:23 PM jade.core.RuntimeImpl run
INFO: JADE is closing down now.
MH100A Agent MH100A@192.168.0.10:1099/JADE Terminating.
NBH100A-Agent NBH100A@192.168.0.10:1099/JADE Terminating.
KMSA-Agent KMSA@192.168.0.10:1099/JADE Terminating.
SYNUA-agent SYNUA@192.168.0.10:1099/JADE Terminating.
BUILD SUCCESSFUL (total time: 127 minutes 30 seconds)

```

The responsible agents will find the invoices, convert/format into the destination type and then transmit bounded by the polling frequency. Three records are shown below as having been delivered successfully for the staff member 7070 and his dependant 400032. The staff was treated at Nairobi hospital while the dependant 400032 was treated at Getrude's (GCH104) and later at Matter Hospital (MH100), with other relevant invoice details as shown.

InvoiceNo_You	Invoice_Number	FacilityCode	ResponsibleAgent	Invoice_Timestamp	PatientRefID	StaffRef	Dependant	FirstName	MiddleName	LastName
400032	11451.50	GCH104	GCH104A	2013-07-26 17:11:44	454602	7070	400032	Tanya	Wendy	Kibera
400032	12874.80	MH100	MH100A	2013-07-26 18:59:35	454602	7070	400032	Tanya	Wendy	Kibera
400032	34641.50	NBH100	NBH100A	2013-07-26 17:09:04	400032	7070	Staff	Steele	Mitchel	James

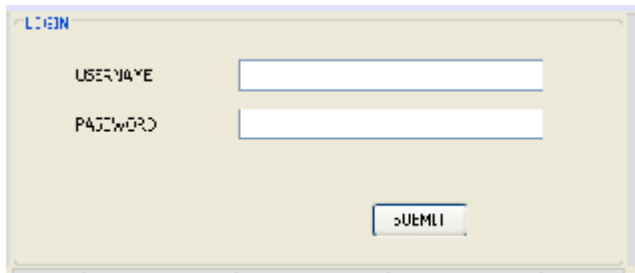
Screen shot below shows all the invoices combined as they are received from various Facilities/Hospitals, including the three selected above for Staff 7070.

InvoiceNo_You	Invoice_Number	FacilityCode	ResponsibleAgent	Invoice_Timestamp	PatientRefID	StaffRef	Dependant	FirstName	MiddleName	LastName
61987684	73333.37	GCH104	GCH104A	2013-07-17 13:33:15	493574	5858	400032	Ashika	Nabuta	Muchina
400032	24771.45	MH100	MH100A	2013-07-25 08:40:45	704210	7970	400032	Verne	Russa	Satana
400032	3238453.00	MH100	MH100A	2013-07-15 04:03:43	453333	8201	400032	Eccard	Wendy	M'engo
400032	9135.50	MH100	MH100A	2013-07-17 13:59:59	644446	5858	400032	Elisa	Lusida	Hoburu
400032	4416.74	MH100	MH100A	2013-07-17 13:16:13	644446	5858	400032	Erny	Mitchel	Hoburu
400032	12955.35	MH100	MH100A	2013-07-15 11:30:52	643366	7999	400032	Ernst	Wendy	Higuna
400032	12334.30	MH100	MH100A	2013-07-25 13:25:35	871476	7070	400032	Tanya	Mitchel	Wende
400032	1166.10	MH100	MH100A	2013-07-17 13:16:13	74384	6030	400032	Kennedy	Wendy	Mgatha
400032	9133.30	MH100	MH100A	2013-07-15 03:51:03	871443	8310	400032	Grace	Mitchel	Kirundo
400032	7455.00	MH100	MH100A	2013-07-15 01:13:17	873443	5380	400032	Rahab	Zawadi	Pouye
400032	4432.55	MH100	MH100A	2013-07-17 13:16:13	644446	5858	400032	Erny	Mitchel	Hoburu
400032	17987.30	MH100	MH100A	2013-07-17 13:53:35	567675	5858	400032	Mirskal	Muchina	Mugane
400032	30000.20	MH100	MH100A	2013-07-15 03:14:10	4296407	7721	400032	Frank	Delvin	Muchina
400032	3445.21	MH100	MH100A	2013-07-17 13:16:13	346332	8201	Staff	Erny	Wendy	Mgatha
400032	5335.35	MH100	MH100A	2013-05-13 03:03:00	246782	8201	Self	Frank	Delvin	Mgatha
400032	7045.00	MH100	MH100A	2013-05-13 03:03:00	246792	0201	Self	Frank	Delvin	Mgatha
400032	5553.70	MH100	MH100A	2013-05-13 03:47:40	489345	6830	400032	Erny	Wendy	Mgatha
400032	21373.37	MH100	MH100A	2013-07-15 13:53:35	566991	5858	400032	Elisa	Muchina	Muchina
400032	3444.24	MH100	MH100A	2013-07-15 13:53:35	774222	7999	400032	Karim	Mgatha	Mgatha
400032	34543.00	MH100	MH100A	2013-07-25 17:25:54	347444	7070	Staff	Erny	Wendy	Mgatha

5.3 System access and Usage

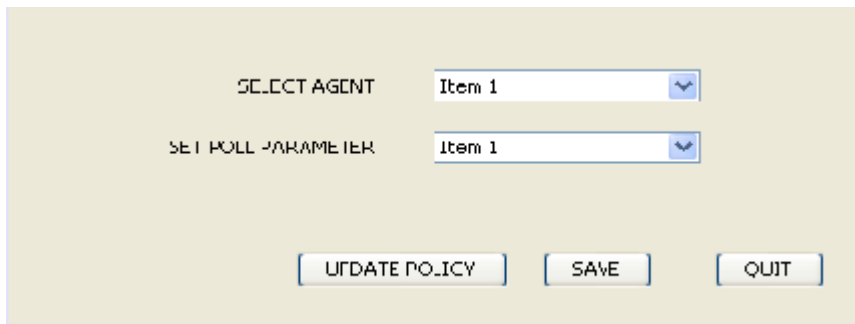
The various interfaces or mechanisms used to access the KRAMAN are explained below;

5.31 The login interface



5.32 CPM (Central Parameter Management) interface

The interface below is used to set the parameters which the agents need to use while interacting with the other agents;



5.33 User Management interface.

The administrator of the system will need an interface for adding the authorised Facilities as well as the users who may login into the system.

5.4 Sample output screen shots

The screen shot below shows a collection of all invoices collected from the facilities. In the manual system this output can only be realised after the invoices have been captured into the KRA Medical system from the physical invoices.

InvoiceNo	You	Invoice_VoucherNo	FacilityCode	Responsible	Invoice_VoucherDate	PatientRefID	StaffRef	Dependant	First Name	MiddleName	Last Name
<input type="checkbox"/>	8632222	009.20	60104	Facility	2013-07-17 10:00:15	481211	1111	0000	Paphael	Joseph	Wesiu
<input type="checkbox"/>	8632222	009.20	60104	Facility	2013-07-17 10:00:15	481211	1111	400091	Harmon	Samson	Boqua
<input type="checkbox"/>	8632222	009.20	60104	Facility	2013-07-17 10:00:15	481211	1111	400021	Abraham	Thomas	Huchina
<input type="checkbox"/>	8707345	0083.45	KEF100	Facility	2013-07-15 10:40:45	5545518	1111	400027	Devine	Samuel	Stard
<input type="checkbox"/>	8707333	0083453.00	KEF100	Facility	2013-07-15 10:40:43	5545518	1111	400022	Francis	Thomas	Wengo
<input type="checkbox"/>	8707333	0083.54	KEF100	Facility	2013-07-15 10:40:50	5545518	1111	400028	Evian	Thomas	Hoburu
<input type="checkbox"/>	8707332	0083.54	KEF100	Facility	2013-07-15 10:40:50	5545518	1111	400028	Evian	Thomas	Koburu
<input type="checkbox"/>	8707301	13456.85	KEF100	Facility	2013-07-15 11:39:53	5545518	1111	400024	Kristopher	Edo	Kiyona
<input type="checkbox"/>	8707343	0083.10	KEF100	Facility	2013-07-15 10:40:53	5545518	1111	400029	Rosemary	Thomas	Wyo Ju
<input checked="" type="checkbox"/>	8707313	0083.90	KEF100	Facility	2013-07-15 10:40:53	5545518	1111	400030	Grace	Thomas	Kiyonko
<input type="checkbox"/>	8707355	0083.00	KEF100	Facility	2013-07-15 10:40:51	5545518	1111	400029	Pahab	Samuel	Ponze
<input type="checkbox"/>	8707355	0083.00	KEF100	Facility	2013-07-15 10:40:51	5545518	1111	400020	Edan	Thomas	Hoburu
<input type="checkbox"/>	8707344	1507.20	KEF100	Facility	2013-07-15 10:40:49	5545518	1111	400021	Abraham	Thomas	Hupane
<input type="checkbox"/>	8707344	11100.20	KEF100	Facility	2013-07-15 10:40:49	5545518	1111	400010	Yadah	Thomas	Huchina
<input type="checkbox"/>	8707305	0083.20	KEF100	Facility	2013-07-15 10:40:45	5545518	1111	500001	Samuel	Thomas	Wengo
<input type="checkbox"/>	8707305	0083.46	KEF100	Facility	2013-07-15 10:40:45	5545518	1111	500001	Samuel	Thomas	Wengo
<input type="checkbox"/>	8707304	0083.20	KEF100	Facility	2013-07-15 10:40:45	5545518	1111	500001	Samuel	Thomas	Wengo
<input type="checkbox"/>	8707303	0083.00	KEF100	Facility	2013-07-15 10:40:45	5545518	1111	400010	Onyango	Thomas	Wawuka
<input type="checkbox"/>	8707355	0083.87	KEF100	Facility	2013-07-15 10:40:45	5545518	1111	400021	Hoburu	Thomas	Hoburu
<input type="checkbox"/>	8707332	0083.24	KEF100	Facility	2013-07-15 10:40:45	5545518	1111	400031	Karuna	Thomas	Kiyonko

The screen shot below shows the transaction activities that have been carried out on the staff table as captured by the triggers. This is the mechanism used to know what data has changed in the relevant table of the KRA Medical system.

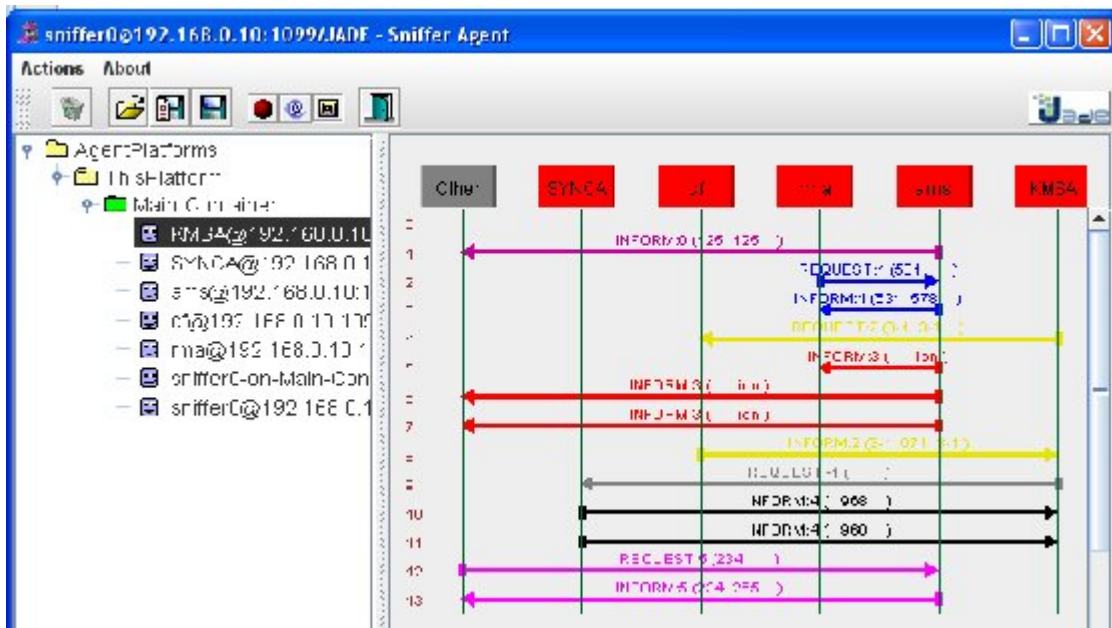
StaffId	StaffFname	StaffMname	StaffLname	StaffGender	StaffDob	StaffJob	StaffTown	StaffMobile	StaffSL	Operation	Opera	OperationDate	Default...	DefaultDate	
37	ESAC	Suzan	Wazene	Makau	Female	965-06-3...	P.O.Box 30304	Kitwaka	77326799	Active	Update	mas	2013-07-18 08:03:11.27	Bank	2013-07-18 08:03:11.27
38	ESAC	Moses	Hagenbe	Ujoro	Male	1973-06-01...	P.O.Box 30322	Kit	723346170	Active	Update	mas	2013-07-18 08:03:11.70	Bank	2013-07-18 08:03:11.70
39	ESAC	Charles	Pipukula	Female	Male	1972-06-05...	P.O.Box 30324	Eldoret	72335738	Active	Delete	sa	2013-07-18 08:03:09.00	Bank	2013-07-18 08:03:09.00
40	ESAC	Charles	Kipchumba	Female	Male	1972-06-05...	P.O.Box 30324	Eldoret	72335723	Active	Insert	sa	2013-07-18 08:03:09.00	Bank	2013-07-18 08:03:09.00
41	ESAC	Mahie	Hibata	Female	Female	1984-12-01...	P.O.Box 2945	Eldoret	72122111	Active	Update	sa	2013-07-17 14:53:00.00	Bank	2013-07-17 14:53:00.00
42	ESAC	Mahie	Hibata	Female	Female	1984-12-01...	P.O.Box 2945	Eldoret	72122111	Active	Update	sa	2013-07-17 14:53:00.00	Bank	2013-07-17 14:53:00.00
43	ESAC	Mahie	Hibata	Female	Female	1984-12-01...	P.O.Box 2945	Eldoret	72122111	Active	Update	sa	2013-07-17 14:53:00.00	Bank	2013-07-17 14:53:00.00
44	ESAC	Mahie	Hibata	Female	Female	1984-12-01...	P.O.Box 2945	Eldoret	72122111	Active	Update	sa	2013-07-17 14:53:00.00	Bank	2013-07-17 14:53:00.00
45	ESAC	Pauline	Hwangi	Female	Male	1953-06-03...	P.O.Box 45700	Nairobi	72286777	Active	Update	mas	2013-07-17 09:45:43.77	Bank	2013-07-17 09:45:43.77
46	ESAC	Antony	Wamwa	Male	Male	1973-07-12...	P.O.Box 31415	Nairobi	72283614	Active	Insert	mas	2013-07-17 09:36:07.79	Bank	2013-07-17 09:36:07.79
47	ESAC	Antony	Wamwa	Male	Male	1973-07-12...	P.O.Box 31415	Nairobi	72283614	Active	Update	mas	2013-07-17 09:36:07.79	Bank	2013-07-17 09:36:07.79

The screen shot below shows the captured operations that have been carried out on the Beneficiary table of the KRA Medical System.

Benefit	Beneficiary	Beneficiary	Beneficiary	Beneficiary	Beneficiary	Beneficiary	Beneficiary	Beneficiary	Beneficiary	Beneficiary	Beneficiary	Beneficiary	Beneficiary	Beneficiary
45	40001	Gischa	Mbari	Miguna	2006-1-03-00-00-0000	Male	7999	Son	Active	Update	mas	2013-07-17 09:36:07.79	Sent	2013-07-17 09:36:07.79
46	40002	Aicy	Wagui	Hbugua	2006-4-03-00-00-0000	Male	5252	Son	Default	Insert	mas	2013-07-16 03:38:50.750	Sent	2013-07-16 03:38:50.750
47	40003	Brian	Lukia	Hobuu	2006-1-05-00-00-0000	Male	5252	Son	Active	Insert	mas	2013-07-16 03:38:48.000	Sent	2013-07-16 03:38:48.000
48	40004	Fosaman	Wanau	Kigoto	2011-04-07-00-00-0000	Female	6330	Daughter	Active	Insert	mas	2013-07-16 03:38:48.000	Sent	2013-07-16 03:38:48.000
49	40005	Abraham	Puchoc	Hobuu	2011-06-15-00-00-0000	Male	5252	Son	Active	Update	mas	2013-07-16 03:38:48.000	Sent	2013-07-16 03:38:48.000
50	40006	Fosaman	Wanau	Kigoto	2011-04-07-00-00-0000	Female	6330	Daughter	Default	Update	mas	2013-07-16 03:38:48.000	Sent	2013-07-16 03:38:48.000
51	40007	Grace	Mure	Kirando	2011-06-01-00-00-0000	Female	8310	Daughter	Active	Insert	mas	2013-07-16 03:38:52.550	Sent	2013-07-16 03:38:52.550
52	40008	Fened	Kanyi	Fenge	2006-08-20-00-00-0000	Female	5252	Daughter	Active	Update	mas	2013-07-16 03:38:45.740	Sent	2013-07-16 03:38:45.740
53	40009	Miriam	Karen	Hogoa	2011-03-13-00-00-0000	Female	7999	Daughter	Active	Insert	sa	2013-07-16 03:38:40.000	Sent	2013-07-16 03:38:40.000
54	40010	Miriam	Karen	Kok	2011-03-13-00-00-0000	Female	7999	Daughter	Active	Update	mas	2013-07-16 03:38:45.740	Sent	2013-07-16 03:38:45.740
55	40011	Georgette	Khar	Karen	1971-07-17-00-00-0000	Male	6484	Son	Active	Insert	mas	2013-07-16 03:38:45.740	Sent	2013-07-16 03:38:45.740

5.5 KRAMAS System Monitoring

The following agent sniffer shows interaction between two agents, the KMSA and SYNCA. REQUEST and INFORM FIPA ACL messages are seen to be exchanged.



5.6 User friendly Crystal Report Outputs

The following screen short highlights a formatted report indicating the patients treated per certain facility or hospital. The consumer of this report would be the KRA medical system administrators who to have a global picture of what liabilities or expected bills to be settled before the physical invoices are received.

KRAMAS INVOICES REPORT PER FACILITY										
Invoice Voucher#	Invoice Amount	Facility Code	Responsible Agent	Invoice Voucher Date	Patient RefID	Staff Ref	Respond and Ref	First Name	Middle Name	Last Name
GCH104 Getode Children Hospital										
GAC0282E	8 843.20	GCH104	GCHIC4A	7/17/2013 11:38:15PM	421233	E201	Self	Raphael	Mwaniki	Njeru
GAC69056	8 766.24	GCH104	GCHIC4A	7/17/2013 11:30:16PM	492299	7999	400021	Minium	Karanu	Mogoa
GAC6957E	14 843.00	GCH104	GCHIC4A	8/5/2013 5:05:06PM	492456	E201	400017	Martin	Kulhi	Ndantu
GAC83230	11 452.25	GCH104	GCHIC4A	7/26/2013 5:51:44PM	494462	7070	400022	Tanya	Nkatna	Nteere
GAC87634	72 333.87	GCH104	GCHIC4A	7/17/2013 11:38:16PM	493574	E262	400021	Acranam	Mubulu	Muchira
MH100 Mater Hospital										
KC8367E	12 400.85	MH100	MH103A	8/5/2013 4:53:54PM	455533	E201	400022	Beatrice	Waweru	Njenga
KC8455E	7 456.00	MH100	MH103A	7/16/2013 11:13:17AM	873445	5580	400023	Rahab	Kangi	Ronge
KC8456E	8 432.66	MH100	MH103A	7/17/2013 2:01:26PM	614446	E262	400028	Brian	Lecula	Mubulu
K092349	17 987.20	MH100	MH103A	7/17/2013 11:54:35PM	567575	E252	400021	Acranam	Muchira	Mugana
K0923498	68 888.20	MH100	MH103A	7/16/2013 3:14:13AM	4236737	7721	400010	Fridah	Ochieng	Muchira
KC83412	9 123.30	MH100	MH103A	7/16/2013 8:51:03AM	871443	E310	400020	Grace	Muthu	Kwendo
KC8234E	8 355.10	MH100	MH103A	7/16/2013 11:28:32AM	74345	6830	400020	Rosemary	Wambui	Ngitho

NDH100		Nairobi Hospital									
CFR560339	3,344.74	NE-H100	NE-100A	7/16/2013	1:57:45PM	534255	7936	400031	Carina	Muganyizi	Kenneth
CFR560346	3,872.87	NE-H100	NE-111A	7/16/2013	1:41:46PM	563351	8050	411121	Mutitu	Mudhwa	Abraham
CFR560368	5,553.00	NE-H100	NE-111A	6/21/2013	1:47:49PM	483545	683L	400010	Onyango	Cecilia	Wacuka
CFR560364	7,345.31	NE-H100	NE-111A	6/18/2013	10:00:00AM	543752	8201	Self	Mwari	Rehnae	Njeru
CFR560361	3,343.43	NE-H100	NE-100A	3/18/2013	10:00:00AM	543752	8201	Self	Mwari	Rehnae	Njeru
CFR560365	3,745.20	NE-H100	NE-100A	7/16/2013	1:50:13PM	513752	8201	Staff	Mwari	Rehnae	Njeru
CFR5615015	34,610.80	NE-H100	NE-100A	7/20/2013	6:26:54PM	577411	7070	Staff	Neera	Mute nibai	Jamas

CHAPTER SIX: CONCLUSION

6.1 Results and output

A multi agent approach in data integration was successfully implemented and the prototype demo done. The user acceptance was done at KRA medical section, where the users of were able to view the treatment records/invoices even before the hard copy versions arrived for vetting and subsequent capturing into the system. The system has reduced the risks of paying duplicate invoices.

KRAMAS has enabled data sharing between the accredited medical service providers and KRA. This integration prototype is also modular in that it is easier to add new agents which will represent new facilities or hospitals. Since the prototype is a data integration model, another output of most importance is to ensure consistent invoice and staff/beneficiary databases.

6.2 Research Objectives Achievement

These have been met in the following ways:

Project objective: *To design, implement and evaluate an integrated data model between KRA medical system and its medical service providers.* This has been achieved using the prototype as demonstrated.

System objectives:

- a) *To enable the service providers authenticate or identify all KRA members of staff or beneficiary by querying an online service.* This has been achieved whereby it was possible to introduce a member of staff or beneficiary into the KRA medical systems and then the medical service providers were able to check the same while in the comfort of their premises via an online service i.e. before according treatment or consultation services to a patient who claims to be associated with KRA.
- b) *The KRA medical system should be notified by service provider systems about the invoices as they are generated.* The invoices being generated by the Medical service providers to KRA medical system contain treatment information such as date of treatment, name of facility, amount due, invoice date and the incurred cost among other details(See appendices for sample). With this data it was therefore possible to know the number of patients that have been treated per day and from which facility.
- c) *To develop, evaluate and test the proposed model.* The test criteria and results have included, refer to the testing and Validation section for more details.

6.3 Relationship to previous work

Data integration has previously been done using the following mechanisms:

- a) *Mobile agents* that move from data repository to another, where the collector agents accumulated data collected along the itinerary.
- b) *Replication mechanisms* This is where the schemas are similar between the source and destination data source. It could not work in our case because the systems are diverse and owned by private entities.
- c) *Middleware integration*, this include SOA and web services, but this method could not be used since we are tackling integration at the data source level and not at the application level where the above are characterised.

6.4 Agent based integration value to an organization

There is *availability of timely information* which is reliable and hence could be used for decision making both at the facility level and also at KRA. Also The model provides a *data warehouse* which KRA medical section decision makers could consult at any time using a client applications or a database access tool such as toad or crystal reports. Lastly the *risk of paying duplicate invoices is reduced* since the status of each invoice can be tracked from the time it is generated at the facility to the time its paid/honoured at KRA.

6.5 Major findings & Research Contribution

The research work has displayed the complexities exhibited in the integration of heterogeneous legacy systems and how they can be solved using database triggers and JADE agents while working at the data source level. The agents must be kept active and should start automatically when the hosting system starts. It was noted that opening say the *kra_medical* database objects like beneficiary or staff tables and editing a certain record, resulted to lost updates because the triggers did not fire to capture the updates meaning the agents did not participate in the process hence *kra_central* database remained inconsistent with the *kra_medical* database.

However if the prototype is adopted, this model would assist the accredited medical service providers in authenticating staff and dependants from KRA and on the other side there would be better planning and decision making process at KRA.

6.6 Recommendation to future researchers

- ✓ A mechanism to prevent lost updates between *kra_medical* and *kra_central* databases should be enforced. A simple work around to this vulnerability can be achieved by routine

synchronization of the participating databases at pre defined times of the day so that the two ends remain consistent.

- ✓ SMS alerting mechanism should be included such when the Authority receives updates that a staff or dependant has been treated, then similar message should be sent to the patient or guardian as information.
- ✓ The password management process can be implemented in such a way that the users at the hospitals/facilities can be managing their own passwords guided by the set well defined policies and guidelines.
- ✓ The validation process should include the pictures of the staff or beneficiary in addition to the medical or staffID card.

REFERENCES

- 1) Andrea Cali , Diego Calvanese , Giuseppe De Giacomo ,Maurizio Lenzerini, "*Accessing Data Integration Systems through Conceptual Schemas*," no 270-284, November 27-30, 2001.
- 2) Amor, M., Fuentes, L. and Vallecillo, A., "*Bridging the Gap between AO Design and Implementation Using MDA*" , AOSE 2004. LNCS 3382, pp 93-108.
- 3) Anastasios Gounaris, Rizos Sakellariou, Norman W.Paton, Alvaro A Fernandes. "*Distributed and parallel Databases*," Volume: 19 no 2-3, May 2006.
- 4) Beck, K., "*Test-Driven Development by Example.*", Addison Wesley, 2003.
- 5) Barret D., L. Clarke, P. Tarr, and A. Wise. "*A framework for event-based software integration*". Volume: 12, 5(4),1996.
- 6) Clifton C, M. Kantarcioglu, A. Doan, G. Schadow, J.Vaidya,. "*Privacy-Preserving Data Integration and Sharing*," no 19-26,June 13, 2004.
- 7) Cohen W. W. "*Data integration using similarity joins and a word- based information representation language*," Volume: 18 no 288-321, 2000.
- 8) Dardenne , A., van Lamsweerde, A. and Fickas, S. "*Goal-Directed Requirements Acquisition*". Science of Computer Programming Vol. 20, North Holland, 1993,3-50.
- 9) Fabio B, Giovanni C, Tiziana T , "*Jade Programmer's Guide*",08-April-2010.
- 10) Huang (2002) .*Ontology-based Land Use Information Service on the Semantic Web*. Available at <http://www.ucgis.org/summer03/studentpapers/yuxiahuang.pdf>.
- 11) Kinny D. and M. Georgeff. "*Modelling and design of multi-agent systems*". pages 1–5. Springer-Verlag: Berlin,Germany, 1997.
- 12) McDonald, J.T., Talbert, M.L., DeLoach, S.A. "*Heterogeneous Database Integration Using Agent Oriented Information Systems*". Proceedings of the International Conference on Artificial Intelligence (2000).
- 13) Odell, J.FIPA-The Foundation for Intelligent Physician Agents. Retrieved, read at from <http://www.fipa.org>, 2007.
- 14) Sycara, K. P., Multiagent Systems. "*Analysis and design using MaSE and agent Tool*". In Proceedings of the 12th Midwest Artificial Intelligence and Cognitive Science Conference (MAICS 2001), 2001.
- 15) Wikipedia, "*Service-oriented architecture*" [Online]. Available at: http://en.wikipedia.org/wiki/Service-oriented_architecture [Accessed January 2013].
- 16) Wooldridge, M., Jennings, N., & Kinny, D. 1999."*A Methodology for Agent-Oriented Analysis andDesign*". to be presented at Agents'99, Seattle WA,May 1999.

- 17) Wooldridge, M. *An Introduction to MultiAgent Systems* (1st ed.). Chichester, England: John Wiley & Sons Ltd. 2002)
- 18) Ziegler P and K. R. Dittrich. " *Three Decades of Data Integration- All Problems Solved?* " no 3-12, August 22-27, 2004.
- 19) "BizTalk Server" [Online]. Available at:
http://en.wikipedia.org/wiki/Microsoft_BizTalk_Server
- 20) "Enterprise replicator" [Online]. Available at: <http://enterprise.replicator.daffodilsw.com/>

APPENDICIES

APPENDIX 1: KRAMAS UAT QUESTIONNAIRE

Date.....

Researcher: Name: Raphael Njeru

Organization from: Kenya Revenue Authority

Target Organization(s): Selected Facilities/hospitals that KRA has Medical Services provision contract for its employees and dependants.

Purpose: As part of my academic requirement for attaining an MSc in Computer Science at the University of Nairobi, I have proposed a framework for data integration between KRA and medical service providers. Therefore I wish to collect user views regarding the proposed system and the current scenario on the ground.

Section A

1.Name of your Hospital/Facility.....Branch.....

2. What is your role ? (Cancel out) Doctor/Nurse/Admin/Data clerk

3. Select your age bracket from the scale below:

18-24[] 25-32[] 33-37 [] 38-41 [] 45-50 [] 52-54 [] > 55 []

Section B

Please complete the following questionnaire with specific regard to the above purpose, by placing

	Strongly agree	Agree	Uncertain/ not applicable	Disagree	Strongly disagr
1. Are you are able to authenticate/validate accurately KRA staff/dependants before treatment?	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
2. Do you receive timely and accurate staff/dependant updates from KRA medical section	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
3. A soft copy invoice and other bills incurred by patients affiliated to KRA should be sent in real/near real time	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
4. Introduction of an integrated platform for data sharing will improve your work	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>

a CROSS {X} in the appropriate box.

APPEDIX 2 : SAMPLE CODE (Synchronization Agent- SYNCA)

```
package KRAMED;
import jade.content.*;
import jade.content.lang.*;
import jade.content.lang.sl.*;
import jade.content.onto.basic.*;
import jade.util.leap.*;
import jade.core.Agent;
import jade.core.AID;
import jade.lang.acl.ACLMessage;
import java.sql.DriverManager;
import java.sql.Connection;
import java.sql.ResultSet;
import java.sql.Statement;
import jade.domain.*;
import jade.core.behaviours.CyclicBehaviour;
import jade.domain.FIPAAgentManagement.DFAgentDescription;
import jade.domain.FIPAAgentManagement.ServiceDescription;

public class SYNCA extends Agent
{
    AID AA = new AID("AA@192.168.0.10:1099/JADE",AID.ISGUID);
    AID SYNCA = new AID("SYNCA@192.168.0.10:1099/JADE",AID.ISGUID);
    AID MH100A = new AID("MH100A@192.168.0.10:1099/JADE",AID.ISGUID);
    AID KMSA = new AID("KMSA@192.168.0.10:1099/JADE",AID.ISGUID);
    AID NBH100A = new AID("NBH100A@192.168.0.10:1099/JADE",AID.ISGUID);
    AID GCH104A = new AID("GCH104A@192.168.0.10:1099/JADE",AID.ISGUID);

    protected void setup()
    {
        DFAgentDescription dfd= new DFAgentDescription();
        dfd.setName(getAID());
        ServiceDescription sd =new ServiceDescription();
        sd.setType("KRA_MAS");
        sd.setName(getName());
        dfd.addServices(sd);
        try
        {
            DFService.register(this, dfd) ;
        }
        catch (FIPAException e)
        {
            System.err.println(getLocalName()+"Registration with DF Failure Reason :"+ e.getMessage());
            doDelete();
        }
        System.out.println("SYNCA Started...."+this.getLocalName());
        addBehaviour(new ReceiveTreatmentRecords());

        //implementation of ReceiveTreatmentRecords()
    }
    protected void takeDown() {
        // Deregister from the yellow pages
        try {
            DFService.deregister(this);
        }
        catch (FIPAException fe) {
            fe.printStackTrace();
        }
        // Printout a dismissal message
        System.out.println("SYNCA-agent "+getAID().getName()+" Terminating.");
    }
}
```

```

    }
private class ReceiveTreatmentRecords extends CyclicBehaviour
{
    //define variables
    public void action()
    {
        ACLMessage msg = myAgent.receive();
        if (msg != null) //if there is a message(s)
        {

            switch (msg.getSender().getName())
            {
                case "KMSA@192.168.0.10:1099/JADE":
                {
                    String SqlFromKMSA = msg.getContent();
                    System.out.println( "\n->" + myAgent.getLocalName() + ": " + msg.getSender().getName()+
",Sent the Query :"+ SqlFromKMSA);

                    Connection conn = null;
                    String url = "jdbc:mysql://192.168.0.10:3306/";
                    String dbName = "kra_central";
                    String driver = "com.mysql.jdbc.Driver";
                    String username = "root";
                    String password = "mysql";
                    try
                    {
                        Class.forName(driver).newInstance();
                        conn = DriverManager.getConnection(url+dbName,username,password);
                        Statement smt = conn.createStatement();
                        // ResultSet rst=smt.executeQuery(InsertQuery);
                        System.out.println("KMSA Successfully Connected to Kra_Central Server...");

                        //get the string message and perform the instruction on the DB as requested
                        smt.executeUpdate(SqlFromKMSA);

                        System.out.println("KMSA Successfully Executed the request in the DB...");

                    }
                    catch (Exception e){e.printStackTrace();}

                    //to reply to the sender

                    ACLMessage reply = msg.createReply();
                    reply.setPerformative(ACLMessage.INFORM);
                    reply.setContent("Record updates sent to the DB...");
                    reply.addReceiver(msg.getSender() );
                    send(reply);
                    break;
                }//end case for "KMSA@192.168.0.10:1099/JADE":

                case "MH100A@192.168.0.10:1099/JADE": //From Matter Hospital
                {

                    String SqlFromMH100A = msg.getContent();
                    System.out.println( "\n->" + myAgent.getLocalName() + ": " + msg.getSender().getName()+
",Sent the Query :"+ SqlFromMH100A);

                    Connection conn = null;
                    String url = "jdbc:mysql://192.168.0.10:3306/";
                    String dbName = "kra_central";

```

```

String driver = "com.mysql.jdbc.Driver";
String username = "root";
String password = "mysql";
try
{
    Class.forName(driver).newInstance();
    conn = DriverManager.getConnection(url+dbName,username,password);
    Statement smt = conn.createStatement();
    // ResultSet rst=smt.executeQuery(InsertQuery);
    System.out.println("MH100A Successfully Connected to Kra_Central Server...");

    //get the string message and perform the instruction on the DB as requested
    smt.executeUpdate(SqlFromMH100A);

    System.out.println("MH100A Successfully Updated the Invoice from Matter Hospital...");

}
catch (Exception e){e.printStackTrace();}

//to reply to the sender

ACLMessage reply = msg.createReply();
reply.setPerformative(ACLMessage.INFORM);
reply.setContent("Record updates sent to the DB...");
reply.addReceiver(msg.getSender() );
send(reply);
break;
} //end case for "KMSA@192.168.0.10:1099/JADE":

case "NBH100A@192.168.0.10:1099/JADE":
{
    String SqlFromNBH100A = msg.getContent();
    System.out.println( "\n->" + myAgent.getLocalName() + ": " + msg.getSender().getName()+
",Sent the Query :"+ SqlFromNBH100A);
    Connection conn = null;
    String url = "jdbc:mysql://192.168.0.10:3306/";
    String dbName = "kra_central";
    String driver = "com.mysql.jdbc.Driver";
    String username = "root";
    String password = "mysql";
    try
    {
        Class.forName(driver).newInstance();
        conn = DriverManager.getConnection(url+dbName,username,password);
        Statement smt = conn.createStatement();
        // ResultSet rst=smt.executeQuery(InsertQuery);
        System.out.println("NBH100A Successfully Connected to Kra_Central Server...");

        //get the string message and perform the instruction on the DB as requested
        smt.executeUpdate(SqlFromNBH100A);

        System.out.println("NBH100A Successfully Executed the request in the DB...");

        //Update this Record to Sent status from Pending if it is an INSERT,
        //Else just update it to reflect changes

    }
    catch (Exception e){e.printStackTrace();}

//to reply to the sender

```

```

        ACLMessage reply = msg.createReply();
        reply.setPerformative(ACLMessage.INFORM);
        reply.setContent("Record updates sent to the DB...");
        reply.addReceiver(msg.getSender() );
        send(reply);
        break;
    }//end case for "NBH100A@192.168.0.10:1099/JADE":

        case "GCH104A@192.168.0.10:1099/JADE":
        {

            String SqlFromGCH104A = msg.getContent();
            System.out.println( "\n->" + myAgent.getLocalName() + ": " + msg.getSender().getName()+
",Sent the Query :"+ SqlFromGCH104A);
            Connection conn = null;
            String url = "jdbc:mysql://192.168.0.10:3306/";
            String dbName = "kra_central";
            String driver = "com.mysql.jdbc.Driver";
            String username = "root";
            String password = "mysql";
            try
            {
                Class.forName(driver).newInstance();
                conn = DriverManager.getConnection(url+dbName,username,password);
                Statement smt = conn.createStatement();
                // ResultSet rst=smt.executeQuery(InsertQuery);
                System.out.println("GCH104A Successfully Connected to Kra_Central Server...");

                //get the string message and perform the instruction on the DB as requested
                smt.executeUpdate(SqlFromGCH104A);

                System.out.println("GCH104A Successfully Executed the request in the DB...");

            }
            catch (Exception e){e.printStackTrace();}

            //to reply to the sender
            ACLMessage reply = msg.createReply();
            reply.setPerformative(ACLMessage.INFORM);
            reply.setContent("Record updates sent to the DB...");
            reply.addReceiver(msg.getSender() );
            send(reply);
            break;
        }//end case for "NBH100A@192.168.0.10:1099/JADE":
    }//End of switch
}
else {block();}
}
}
} // End of class definition

```