ADDIS ABABA UNIVERSITY

SCHOOL OF GRADUATE STUDIES

COLLEGE OF NATURAL SCIENCES

DEPARTMENT OF COMPUTER SCIENCE

Context Aware Tourist Information and Recommendation System using iQR Code

Hassen Ibrahim

A THESIS SUBMITTED TO THE SCHOOL OF GRADUATE STUDIES OF ADDIS ABABA UNIVERSITY IN PARTIAL FULFILMENT FOR THE DEGREE OF MASTERS OF SCIENCE IN COMPUTER SCIENCE

February 2015

ADDIS ABABA UNIVERSITY
SCHOOL OF GRADUATE STUDIES
COLLEGE OF NATURAL SCIENCES
DEPARTMENT OF COMPUTER SCIENCE

Context Aware Tourist Information and Recommendation System using iQR code

Hassen Ibrahim

ADVISOR: Dejene Ejigu (PhD)

Signature of the board of Examiners for approval:

Name	**Signature**
1. Dejene Ejigu (PhD)	_____
2. _____	_____
3. _____	_____

Dedication

To my Parents

Acknowledgements

First and Foremost I would like to thank creature of the universe, almighty God, Allah for giving me the wisdom and strength I need to fulfill my duty. *Alhamdulillah!*

I would like to thank my Advisor Dr. Dejene Ejigu for his valuable comments, support, and guidance throughout this thesis work. Without his encouragement and constant guidance, I could not have finished this thesis. *Thank you Dr.!*

It is a great pleasure to get an opportunity to thank my families for supporting and teaching me what is important in life. *Thank you My Families!*

Finally, I would like to thank my colleagues and friends for their encouragement and support throughout this thesis work.

Table of Contents

List of Tables

List of Figures

List of Algorithms

List of Acronyms and Abbreviations

ADT	Android Development Tool
API	Application Program Interface
ANN	Artificial Neural Network
CATIS	Context Aware Tourist Information System
CAMIS	Context Aware Museum Information System
CL	Confidence Level
CM	CoMment
IDE	Integrated Development Environment
MAC	Media Access Control
NME	National Museum of Ethiopia
NF	Negative Factor
NTO	National Tour Operation
OWL	Web Ontology Language
POI	Point of Interest
QR code	Quick Response code
RDF	Resource Description Framework
RDFS	Schema for RDF
RSSI	Received Signal Identification
SDK	Software Development Kit
SPARQL	SPARQL Protocol and RDF Query Language
SSID	Service Set Identification
TIC	Tourist Information Center
UDDI	Universal Description, Discovery and Integration
UK	United Kingdom
URL	Uniform Resource Locator
VISIT	Virtual Intelligent System for Informing Tourists
XML	Extensible Markup Language
Zxing	Zebra Crossing

Abstract

Advancements in mobile computing and communication technologies have led to the emergence of a concept pervasive (ubiquitous) computing. In pervasive computing, ubiquitous environment that combines devices and sensors with intelligent software and wireless technologies is created to improve quality of life. Ubiquitous computing is applicable in different domains. Among this, tourism domain is the one that will be suitable for providing ubiquitous access on the basis of customization because, tourists will come to visit certain tourist attractions by holding different technologies and they will have no or little information about a tourist attraction.

In a tourism environment providing organized information to visitors about tourist attraction is a challenging task. To handle this problem QR or iQR codes are being used as an ideal solution. However, the information provided using QR or iQR codes doesn't consider the context of tourists.

In order to solve such problem, this research work proposed a context based tourist information and recommendation system using iQR code that can support tourists while visiting certain tourist attraction. We have mapped private tourist context (such as identity, location, device and nearest member/s) with public information about Point of Interest (POI) encoded on iQR code to provide context based information about a POI. We have used iQR codes for storing public information about POI due to its high data storage capacity and flexibility compared to QR codes. To minimize scanning overhead by all members in a group visit the proposed system considers nearest member/s to deliver appropriate service. Recommendation of POIs is based on a hybrid approach. It uses personal rating history, current context and others rating to recommend unvisited tourist attractions. In order to show the usability of the system it is implemented on National Museum of Ethiopia (NME).

Keywords:

Pervasive/ubiquitous Computing, Context, Context Awareness, QR code, iQR code

Chapter 1: Introduction

1.1 Overview

Ubiquitous computing creates a saturated environment with computing and communication capabilities integrated with users. Its main essence is to assist day to day activities of human beings by using different available technologies around the environment with no or minimal distraction.

It is characterized by interaction transparency, automated capture of experience and context-awareness [1]. Interaction transparency means it hides the availability of pervasive technologies from users. Automated capture of experience means that application can remember when, where, and why something was done and make it available for future use. Context-awareness capability allows applications to be aware of the environment in which they are operating on.

Context aware computing is a mobile computing paradigm in which applications can discover and take advantage of contextual information (such as user location, time and day, nearby people and devices and users activity) [2]. Context aware applications respond to changes in the environment in an intelligent manner in order to enhance the user experience. Context aware applications should be enhanced mobile applications because: i) user changes frequently subject to the user mobility behavior and ii) the need for context aware behavior is great in a mobile environment [3].

The common application domain for providing ubiquitous access to users is the tourism domain [4] and the best way for delivering those context aware tourism services is by using user's mobile phones since 80% of the world population uses mobile phones for their daily activities, that means there are more than 5 billion phones worldwide [5]. Among these 1.08 billion of phones are Smartphone. According to ethio telecom report more than 17 million mobile users are available in Ethiopia [6]. So providing a context aware tourist guide system using mobile devices will create a good experience to retrieve and deliver information to users.

Additionally, there are two dimensional (2D) barcodes called Quick Response (QR) codes, as shown in Figure 1.1, which can be scanned by range of devices (like Smartphone) and deliver

information to users with very little or no overhead cost [7]. QR codes can hold approximately 7000 numeric characters. They emerged in response to the need of codes capable of storing more information, more character types, and that could be printed in a smaller space. The creator intended the code to allow its contents to be decoded at high speed [8]. Unlike the older one-dimensional barcode that was designed to be mechanically scanned by a narrow beam of light to extract data, the QR code is detected as a 2-dimensional digital image by a semiconductor image sensor and is then digitally analyzed by a programmed processor and sent to users' phone.

Compared with one-dimensional bar codes or other two-dimensional codes, QR codes can hold a larger amount of data in a smaller space. In addition, an advanced error-correction method and other unique characteristics allow QR Code to be read more reliably than other codes [9].

Those QR codes are currently being used for tourist guide systems [10, 11, 12]. In the systems QR codes will be placed on separate locations and whenever tourists scan the codes by using their Smartphones they will be redirected to a web page to get information about a given historical place. However, their main drawback is the systems do not include context (personal, mobile, location, weather, etc.) when providing information to tourists. In addition, even if Ethiopia is one of the few countries that have highest number of tourist destinations (like historical places and heritages), QR codes are not used in any of those historical places to deliver information easily to tourists.

Also, Denso wave incorporated developed an extended version of QR code called iQR code [13] as shown in Figure 1.2, which can store more data and can be printed as a rectangular code. The new 2D barcode was created at the request of the industrial market for a code with a higher data capacity, printable in a smaller space and working in a rectangular format. iQR code has an 80% higher data storage capacity, and for the equivalent data is 30% smaller than model 2 type of QR code. Also the new code can be printed in a rectangular format for printing in narrow space or cylindrical products. So if we use this code in a tourism environment we can store more information and easily present it to tourists.

Figure 1.2: Rectangular iQR code created in different size

Our work deals with a context aware system for tourist guide with emphasis of iQR codes as a source of information. In the system we build a context aware information service to tourists from iQR code data and private tourist contexts. Those contexts include personal, time, location and device. It also supports group tour service in order to support social interaction of group of tourists. The group tour service includes: i) a recommendation of point of interests, and ii) group information system by identifying a member near to a person that scan the iQR code.

1.2 Motivation

Ubiquitous computing offers new opportunities and challenges in terms of time-aware, location-aware, device-aware and personalized services which can be achieved by using customization [4]. One of the application domains particularly suited for providing ubiquitous access on the basis of customization is the tourism domain. Information can be provided to tourists at preparatory state and/or at the time of visit by personalizing information about Point of Interests (POIs).

Different hardware and software technologies are used to deliver information about a tourist attraction including RFID tags, NFC tags, QR codes, and iQR codes. Among these iQR codes

can be an ideal solution to provide organized information to tourists for a single tourist attraction because, they are easier and cheaper to get, they do not require special equipment to access the information (i.e., if a tourist has Smartphone and barcode decoding software, he/she can easily access the information encoded on iQR codes), and they are more flexible.

The main motivation to do the research work is the growing use of QR codes in a tourism environment and the capability of iQR code to store large amount of information than the traditional QR code. However, that information is not provided by considering contexts of tourists. Additionally, in Ethiopia even if there are a lot of tourist attractions it is hard to find a tour guide that guide tourists in a given tourist environment.

Tourists may visit certain tourist attraction in a group with their families, friends, classmates, etc., either for minimizing cost and/or for enjoying collectively. So there should be a service that can support group tour. For example in a group visit if a group leader scans iQR code of a given heritage the information should be delivered to nearest member/s of the group. The proposed system considers group of visitors have the same preference with different hardware technologies hold by them.

1.3 Statement of the Problem

When developing context aware human centered system the context should be used to say "the 'right' information, at the 'right' time, in the 'right' place, in the 'right' way, to the 'right' person" [14]. In order to support the above principle researches [28, 29, 31, 32] for tourism guide have been done. Those researches use personal, mobile, location and time contexts to provide efficient service to tourists. Currently QR codes are being used in tourism to provide location based information to tourists [10, 11, 12]. However, their main drawback is they only provide static information to users, i.e., those systems aren't aware of context of the environment, users and also devices. For example, a tourist who comes to visit certain historical area, he/she may be interested only on the architecture of the building but not the history of the building. So the information provided by QR codes should be aware of the context of users and the environment.

In most context aware tourist application social context information, i.e., the relationship existing among a group of tourists is seldom taken in to account [15]. For example, when group of

tourists are visiting certain historical area if a group member scans iQR code of certain building the information should be delivered to devices of all nearest members without scanning the code one by one or if they want to have individual visit and/or break and want to meet together after some time, assisting to the nearest free group member should consider group members context.

So, in this research work, in order better serve users' (tourists') need in a tourism environment, we added private context information to public iQR code data and present those information according to users context. In addition, we exploit social context information among a group of visitors to assist group visit. It also performs a hybrid recommendation system by combining the rating of other tourists with tourists' previous rating.

1.4 Objective

General Objective

The general objective of this research is to model and develop iQR code based context aware tourist information and recommendation system that supports individual and group tour.

Specific Objectives

Specific objectives of this research are:

- ✓ A critical analysis of related works
- ✓ Analyze social and personal context parameters in a tourism environment
- ✓ Design a new architecture for context based information system from iQR code data
- ✓ Build a hybrid context based recommendation system
- ✓ Build a prototype of the system

1.5 Methodology and Tools

To achieve general and specific objectives of the research different methodologies are employed. These are:-

a. Literature Review

We will conduct an extensive review of important literatures in order to have a deep understanding in the area of pervasive computing and iQR code. Detailed analysis of related works will be made to identify problem areas and to come up with a best solution for context aware tourist information service by using iQR code.

b. Interview and observation

For designing the proposed system and developing a prototype we will perform onsite observation and interview with staff members of National Museum of Ethiopia (NME).

c. Prototype development

We have identified the necessary tools and technologies to develop the prototype. Then we have implemented and demonstrated the prototype. In addition, we have tested our system by creating possible scenarios.

1.6 Scope and Limitations

The scope of this work is to design and build a context aware individual or group tour information system using iQR code on smartphones. It also includes group information and recommendation service. Implementation will be limited to developing a prototype. The prototype for the designed system is tested only on Smartphone that run android OS.

1.7 Application of Results

Companies, institutions and organizations are using either QR codes or iQR codes for tourist information system, for providing product information to users, for assisting students in a library, etc. so the result of this research work will have a huge application on the above areas because it adds the necessary context information on iQR code data and present to users. Specially, tourist centers like historical places, art galleries and museums will be benefited hugely on the result of this research work.

1.8 Thesis Organization

The remaining part of this document is organized as follows. Chapter 2 reviews concepts related to pervasive computing and iQR codes. It discusses about context, context awareness and context aware systems. Context acquisition, modeling and reasoning concepts and techniques are also presented. In addition, it describes about QR codes and iQR codes and their characteristics, types of QR codes and the importance of using iQR code and their implementation area.

Chapter 3 reviews research works related to this work. Works done in a tourism environment and iQR codes are considered. The architecture of each work is described in detail. Finally, it shows how iQR codes can be used to deliver organized information to visitors for each tourist attraction.

The fourth Chapter discusses details of the proposed system. How context is acquired, modeling entities and reasoning is described. It shows how to decode iQR code information and map with the retrieved context information.

In Chapter 5 implementation of the proposed system is presented. It discusses possible scenarios in a context aware museum environment. Finally, we have evaluated the prototype by using scenarios described in this Chapter.

The last Chapter deals with conclusion of our study and future works. It summarizes the study and describes contribution of our research work. It provides potential research areas that can be considered as a continuation for this work.

Chapter 2: Literature Review

This Chapter deals with the review of important concepts related to pervasive computing to make our proposed work understandable by readers. It discusses basic concepts about pervasive computing, context, context aware systems and their elements, context acquisition, modeling and reasoning. In addition, it describes about QR code and iQR code, and their implementation areas.

2.1 Pervasive Computing

Modern computing environment is classified in to three major computer eras Such as: the Mainframe era, the era of Personal Computers (PC) and the era of ubiquitous (pervasive) computing. In the first era a powerful single large time-shared computer called mainframe computer was available, which was owned by an organization and used by many people at the same time. Second, the era of PC, is characterized by a personal computer primarily owned and used by one person, and dedicated to him/her. The third era, ubiquitous computing, representative of the present time, is the era of small networked portable computer products in the form of smart phones, Personal Digital Assistants (PDAs), smart watches, wearable technologies and additional embedded computers built into many of the devices we own resulting in a world in which each person owns and uses many computers. Each era has resulted in progressively a larger number of computers to be integrated into everyday life [44].

The term ubiquitous computing is first coined two decades ago by Mark Weiser, who was a chief technology officer of Xerox's Palo Alto Research Center in 1991. He described the term ubiquitous computing as, "the most profound technologies are those that disappear, they weave themselves into the fabric of everyday life until they are indistinguishable from it" [16]. The essence of that vision was the creation of environment saturated with computing and communication capabilities, yet gracefully integrated with human users.

When the idea was first raised the necessary hardware technology needed to achieve did not exist, However currently when we see different technologies and software architectures being developed ubiquitous computing is no longer a vision; it is rapidly becoming a reality.

2.2 Context, Context Awareness and Concepts

2.2.1 Context

Context is a powerful and longstanding concept in human-computer interaction. Interaction with computation can be by explicit acts of communication (e.g., pointing to a menu item) and/or implicit (e.g., default settings). Context can be used to interpret explicit acts, making communication much more efficient. Thus, by carefully embedding computing into the context of our lived activities, it can serve us with minimal effort on our part. Communication can be not only effortless, but also naturally fit with our ongoing activities. Pushing this further, the actions we take are not even felt to be communication acts at all; rather, we are just engaged in normal activities; and the computation becomes invisible [44].

After the concept context aware computing is coined various scholars give different definitions to the term *context.* Free online Dictionary of computing define context as "that which surrounds, and gives meaning to, something else" [17]. This definition can be specialized to the application at hand. Whether that "something" is an assertion in logic, a person, an utterance, or a computer system, with a suitable definition for "meaning," the intuition captured by the word context serves its purpose. Schilit *et al.* [18] defines context from the perspective of distributed and mobile computing, where a person is "something," and context refers to information about a person's proximate environment, such as location and identities of nearby people and objects. More appropriate and well known definition is given by authors Dey and Abowd [19]. In their work the term context is defined as: "Context is any information that can be used to characterize the situation of an entity. An entity is a person, place, or object that is considered relevant to the interaction between a user and an application, including the user and applications themselves." There has been much work in identifying what such information can be, the structure of the information, how to represent such information, and how to exploit context in specific applications. Contexts can include information such as location of person or objects, time, execution state of applications, computational resources, network bandwidth, activity, user intentions, user emotions [20], and conditions of the environment [19].

2.2.2 Context-Aware System

A context-aware system is described as a system which uses contexts to provide relevant information and/or services to the user, where relevancy depends on the user's task [19]. It was pioneered by XEROX PARC and Olivetti Research Ltd. Context aware computing involves many individual elements:

1. **Sensor Technology** - Numerous hardware devices must be equipped with the capability to collect information that will form part of the context of the system. These devices should be relatively inexpensive and readily available, and should hopefully require a minimum amount of configuration and management. They must also be capable of transmitting information to some central location, or else communicating with nearby devices.
2. **Context Model** - A model of the context information must be formulated to provide a resource for applications to avail of.
3. **Decision Systems** - Once the context model has been formed, elements of the system must make decisions based on the available information. These decisions can be made either at application level based on the information available to it, or can be made centrally and then disseminated to the individual agents in the system.
4. **Application Support** - Application programmers need to be aware of context information, since, this information will fundamentally change how they work. Instead of being driven primarily by explicit user input as they have been in the past, applications will begin to 'act for themselves', but must do so in a way that attempts to adhere to the law of least surprise for the user. Ideally, user interaction with a context-aware application should be simpler and more productive than with a traditional application.

A context-aware system must be capable of mimicking a human's ability to recognize and exploit implicit information in the environment, in order to advance the operations of its functionalities [21]. Although identifying and deducing human activity is a challenge, it is critical that context-aware applications should operate by conveying the appropriate information to the right place at the right time through inferring the user's intention. To accomplish this objective context aware systems must [21]:

- Gather information from the environment or the user's situation.
- Translate this information into the appropriate format.
- Combine context information to generate a higher context. A higher context is context information that is derived as a result of the merger of other context information.
- Automatically take action based on the retrieved information.
- Make the information accessible to the user, immediately, in the future, or when it is required, to enhance and aid in the completion of the user's task.

One goal of context-aware computing is to acquire and utilize information about the context of a device to provide services that are appropriate to the particular people, place, time, events, etc. For example, a cell phone will always vibrate and never beeps in a concert, if the system can know the location of the cell phone and the concert schedule. However, this is more than a question of gathering more contextual information about complex situations. More information is not necessarily more helpful. Further, gathering information about our activities intrudes our privacy. Context information is useful only when it can be usefully interpreted, and it must be treated with sensitivity. Moreover, a context aware system focuses on the contextual information manipulated by a certain application [21].

2.2.3 Challenges in Context Aware System

Implementing context-aware system requires addressing many issues [22]:

- How does the system represent context internally? How do we combine this information with the system and application state? Where should the system store context locally, on the network, or both? What are the relevant data structures and algorithms?
- How frequently does the system need to consult contextual information? What is the overhead of considering context? What techniques can we use to keep this overhead low?
- What are the minimal services that an environment must provide to make context awareness feasible? What are reasonable fallback positions if an environment does not provide such services?
- What are the relative merits of different location-sensing technologies? Under what circumstances should we use one and not another? Should we treat location information

just like any other contextual information, or should we handle it differently? Is historical context useful?

- How frequently does the system need to be updated on context changes?
    - How often to poll?
    - How often to change behavior?
- What sensors and sensor infrastructures are necessary?
    - What is the fallback condition?
- How to sense location information?
    - Technical details
    - History of location?

2.3 Context Acquisition and Modeling

2.3.1 Context Acquisition

Context acquisition is an essential component when developing context-aware systems. It obtains context data from diverse context sources comprising of sensor networks designed to sense and retrieve raw context data that confronts to a low-level data model. The sensed contextual information is inferred to achieve high level understanding of the situation.

Some context information can be given to the context-aware system explicitly, such as a user's name or age; other context information can be obtained through the use of sensors. Many types of sensor are already commonly in existence and can provide primitive physical information such as light, heat and pressure readings. Other types of context such as facial recognition rely on fairly simple sensors such as cameras, but require considerable processing such as image recognition in order to make use of the information obtained.

Location and identity are the most frequently sensed pieces of context. For example, Active Badges [23] emit infrared signals which give a rough location and ID. Location is an important element of context information. Many different approaches have been taken to determining the location of agents within a context-aware system. GPS, Infrared and radio signals have all been explored. Many context-aware systems that have been produced are only aware of location

information. While these applications are useful, location is only one element of the wider context information. Schilit *et al.* discusses this fact in [18].

> *"Sensors are not always 100% accurate or reliable, particularly if they are disposable. The information gathering system must be tolerant of sensor failure, and any information gathered from sensors must be subjected to sanity checks to help verify its correctness. Sensor fusion is one method of avoiding this difficulty*."

Sensor fusion means aggregating the results of different sensors together to produce a reasonable approximation of the state of the system. This means that some sensors can fail or give erroneous answers, but the system will still be able to determine the real state through the use of a voting mechanism. When considering the output of temperature sensors for example, it might be prudent either to simply average the results or alternatively to discard reported values that differ too greatly from what other sensors report. This method would avoid drastic measures being taken by the context-aware system to correct what it considers to be temperature variations but are actually simply the result of sensor failure [18].

2.3.2 Context Modeling

No model has been standardized for managing context, yet there is a necessity to design a model to optimize the benefits gained from employing context in the application. The management model should handle context in a reusable manner to permit context from one source to be exploited by many distinct applications that perform a variety of tasks [21]. The authors of context Awareness in Mobile Computing Environment [3] describe context modeling as a research area that tries to answer questions like:

1. Which are the most appropriate contextual information that can model well enough the specific context in a certain domain (e.g., epistemic field, such as mobility behavior of a user, network congestion control, early warning, and transportation traffic notification)?
2. Which are the relations among such pieces of information?
3. How can one take into account the information change, and, how can one react to such change, if necessary?

Hence, modeling context is a technique focusing on how to find and relate contextual information that better captures the observation of certain worlds of interest.

There are different context representations or modeling approaches varying in accompanying features. Mark-up-scheme model, object-oriented model, logic based model, object-relational model, and ontology-based model are some of them. Pervasive computing systems require context modeling approaches to fulfill different requirements. Some of the requirements are: distributed composition, partial validation, richness and quality of information, incompleteness and ambiguity, level of formality, and applicability to existing environments. Table 2.1 [24] shows the features to each of these pervasive computing requirements against different context data representation or modeling approaches. Among the list of context modeling techniques we have proposed ontology based approach because it facilitates context reasoning, reusability, derivation and consistency of knowledge.

Table 2.1: Context Modeling Approaches

	Approaches				
Requirement	**Mark-up Scheme**	**Graphical Model**	**OO Model**	**Logic Based**	**Ontology Based**
Distributed Composition	+	-	++	++	++
Partial Validation	++	-	+	-	++
Quality of Inforamtion	-	+	+	-	+
Incompleteness/Ambiguity	-	-	+	-	+
Level of Formatity	+	+	+	++	++
Applicability	++	+	+	-	+
Key: ++ Compressive + Partial - Limited or none					

2.3.3 Context Reasoning

Context reasoning is a process for inferring new context from previously defined contexts. It checks the consistency of context and deduces high level, implicit context from low-level, explicit context [25].

2.4 QR Code and iQR Code

2.4.1 QR code

A QR code that stands for "Quick Response," is a two-dimensional barcode first developed by Japanese company Denso Wave in 1994 apparently in order to track manufactured items [7]. It has the capability to encode plain text, URL, contact information, etc. The capability to store information in both vertical and horizontal directions allows it to hold more information than the traditional barcode.

A QR code consists of black modules (square dots) arranged in a square grid on a white background, which can be read by imaging devices like camera and then digitally processed [45]. Software prepares the captured image data until it can be algorithmically processed according to the QR code standard so the QR code content can be read. This process is assisted by the use of the Reed-Solomon error correction.

A QR code acts as a link embedded in the real world, integrating it with the virtual computer world. Currently they are being widely used for advertising campaigns, linking to company websites, contest sign-up pages and online menus [45]. A QR code with a website of Addis Ababa University encoded on it is shown in Figure 2.1.

Figure 2.1: Sample QR Code Linked With URL "http://www.aau.edu.et/"

2.4.1.1 Characteristics of QR Code

As discussed in [46, 47] the characteristics of QR codes are listed below:

a. High Capacity Encoding of Data

QR Code is capable of encoding 7,089 characters in one symbol which is several dozens to several hundred times more information than the conventional barcode which can store a maximum of approximately 20 digits. In addition, it can handle all types of data, such as numeric and alphabetic characters, Kanji, Kana, Hiragana, symbols, binary, and control codes.

b. Small Printout Size

Since QR Code carries information both horizontally and vertically, it is capable of encoding the same amount of data in approximately one-tenth the space of a traditional barcode.

c. Resistance to Dirt and Damaged Symbols

QR Code has error correction capability. Data can be restored even if the symbol is partially dirty or damaged. They have four error correction levels (7%, 15%, 25% and 30% per symbol area). So a maximum of 30% of codewords can be restored by QR code.

d. Readable from any Direction in 360^0

QR Code is capable of 360 degree (Omni-directional), high speed reading. QR Code accomplishes this task through position detection patterns located at the three corners of the symbol. These position detection patterns guarantee stable high-speed reading, circumventing the negative effects of background interference.

e. Structured Appending Feature

QR Code can be divided into multiple data areas. Conversely, information stored in multiple QR Code symbols can be reconstructed as a single data symbol. One data symbol can be divided into up to 16 symbols, which allow to print in a narrow area.

f. Kanji Kana Capacity

As a symbology developed in Japan, QR Code is capable of encoding JIS Level 1 and Level 2 Kanji character set. In case of Japanese, one full-width Kana or Kanji character is efficiently encoded in 13 bits, whereas other 2D symbologies would require 16 bits (2 bytes) for a single character allowing QR Code to hold more than 20% data than other 2D symbologies.

2.4.1.2 The QR Code Structure

QR Code is a matrix type symbol with a cell structure arranged in a square. It consists of the functionality patterns for making reading easy and the data area where the data is stored. QR Code model 2 has finder patterns, separator, alignment patterns, timing patterns, format information, the data area, error correction and Reminder Bits [47]. The structure of the code is show in Figure 2.2.

a. Finder Pattern

A finder pattern is used for detecting the position of the QR Code and for determining the correct orientation. It consists of three identical structures that are located in all corners of the QR Code except the bottom right one. Each pattern is based on a 3x3 matrix of black modules surrounded by white modules that are again surrounded by black modules. By arranging this pattern at the three corners of a symbol, the position, the size, and the angle of the symbol can be detected. This finder pattern consists of a structure which can be detected in all directions (360°).

b. Separator

The white separators have a width of one pixel and improve the capability of recognizing the finder patters as they separate them from the actual data.

c. Alignment Pattern

It is a pattern for correcting the distortion of the QR Code. It is highly effective for correcting nonlinear distortions. The central coordinate of the alignment pattern will be identified to correct the distortion of the symbol. For this purpose, a black isolated cell is placed in the alignment pattern to make it easier to detect the central coordinate of the alignment pattern.

d. Timing Pattern

A pattern for identifying the central coordinate of each cell in the QR Code with black and white patterns arranged alternately. It is used for correcting the central coordinate of the data cell when the symbol is distorted or when there is an error for the cell pitch. It is arranged both in vertical and horizontal directions.

e. Format Information

The formation information section consists of 15 bits next to the separators and stores information about the error correction level of the QR Code and the chosen masking pattern.

f. Data Area

The QR Code data is stored (encoded) in the data area. The data will be encoded into the binary numbers of '0' and '1' based on the encoding rule. The binary numbers of '0' and '1' will be converted into black and white cells and then will be arranged. The data area have Reed-Solomon codes incorporated for the stored data and the error correction functionality.

g. Error Correction

Similar to the data section, error correction codes are stored in 8 bit long codewords in the error correction section.

h. Reminder Bits (Quite Zone)

This section consists of empty bits if data and error correction bits cannot be divided into 8 bit codewords without remainder.

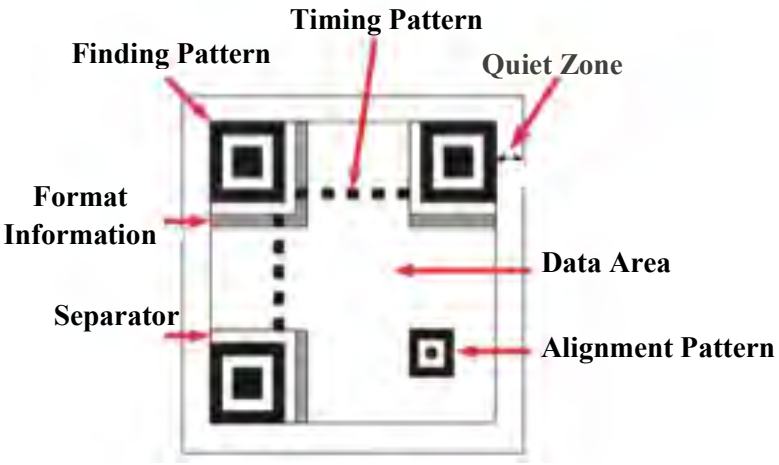

Figure 2.2: Structure of QR code

2.4.1.3 Error Correction Level of QR Code

Code employs the Reed-Solomon error correction algorithm to recover from the damaging part. There are four error detection levels of QR Code symbology ranges from L, M, Q and H[1]. Storage capacity may be relatively reduced when error correction level get higher.

- Level L is able to restore about 7 percentages of codeword.
- Level M is able to restore about 15 percentages of codeword.
- Level Q is able to restore about 25 percentages of codeword.
- Level H is able to restore about 30 percentages of codeword.

2.4.1.4 Types of QR Codes

1. QR code Model 1 and Model 2

Model 1 is the original QR Code, a code capable of coding 1,167 numerals with its maximum version being 14 (73 x 73 modules) [48]. Model 2 is a QR Code created by improving Model 1 so that this code can be read smoothly even if it is distorted in some way. QR Codes that are printed on a curved surface or whose reading images are distorted due to the reading angle can be read efficiently by referring to an alignment pattern embedded in them. This code can encode up to 7,089 numerals with its maximum version being 40 (177 x 177 modules) [48]. Figure 2.3 shows Model 1 and Model 2 QR codes

Figure 2.3: Model 1 and Model 2 QR Codes

[1] http://www.qrcode.com/en/about/error_correction.html

2. Micro QR Code

A major feature of Micro QR Code is it has only one position detection pattern, compared with regular QR Code that requires a certain amount of area because position detection patterns are located in three corners of a symbol [49]. Furthermore, QR Code requires at least a four-module wide margin around a symbol, whereas a two-module wide margin is enough for Micro QR Code. This configuration of Micro QR Code allows printing in areas even smaller than QR Code.

The amount of data that can be stored in a Micro QR Code is not so large (max. 35 numerals). Since this type can encode data more efficiently than the regular QR Code, its size does not have to be made much larger as the amount of data stored increases, unlike the case with the regular code.

There are 4 variations, M1 to M4, of this type of code. The maximum amount of data that can be encoded for the maximum version of this code, M4, is smaller than that encoded by version 1 of the QR Code.

3. iQR Code

iQR Code, as shown in Figure 2.4, is a matrix-type 2D code that allows an easy reading of its position and size [13]. This code allows a wide size range of codes from ones smaller than the traditional QR Code and Micro QR code to large ones that can store more data than these. This code can be printed as a rectangular code, turned-over code, black-and-white inversion code or dot pattern code (direct part marking) as well, allowing a wide range of applications in various areas.

Figure 2.4: iQR codes

4. SQRC

SQRC, as shown in Figure 2.5, is a type of QR Code equipped with reading restricting function. This can be used to store private information and manage company's internal information and the like However this functionality does not mean guaranteed securing of coded data [50].

Figure 2.5: SQRC

5. LogoQ

LogoQ, shown in Figure 2.6, is a new type of QR Code created to enhance visual recognizability by combining it with letters and pictures in full color [51].

Figure 2.6: LogoQ

2.4.1.5 How to Generate QR Code

Denso wave developed two free applications called QRdraw and QRmaker used for generating QR codes. In addition, there are free websites like Kaywa.com, GoQR.me and Zxing project on the Internet that allow users to generate QR code and there are open source codes for smart developers used to develop QR code generating software such as Zebra crossing (Zxing). In addition Google also has a tool for generating QR code.

Before generating a QR code two main points must be taken in to consideration. The first one is module size, in order to generate QR code that is easy to read the size of the module should be set by considering the capability of the printer for printing the QR code and the scanning device. Secondly we have to consider the coder area, in which the area of the QR code is determined at the end depending on its version, the module size and the margin size. Finally adding too much decoration on the code will distort the QR code and it minimizes readability.

2.4.1.6 How to View QR code Information on a Mobile Phone

Different applications are developed to decode QR code data on mobile phones. Mobile applications like QR Droid (for Android Devices), Optiscan (for iPhone/iOS), and QR Code Scanner (for Blackberry) can be used to scan and view QR code information.

2.4.1.7 Application of QR Code

1. Uses of QR Code in Museums and Cultural Institutions

Countries like Australia, USA, Japan, India, etc. are investing hugely to use QR codes on tourist attractions. Using 2D bar codes in a tourism environment allows them to deliver organized information to every tourist by linking the official website of the tourist center, Wikipedia page or YouTube video for minimize printing cost of visitor's guide, to advertize tourist attractions and share thoughts of visitors on social media with friends and families.

Power house museum, Sydney Australia[2] implements QR code early to provide additional information about exhibits by audio, videos, etc. project of the mattress factory, Pittsburgh Pennsylvania links the code to PDF documents, still images and YouTube hosted images to

[2] http://www.powerhousemuseum.com/

minimize the environmental impact of printing many visitor guides[3]. A project by Carinthia open Air museum[4], Carinthia Australia connects QR codes with wiki sites to provide more information about buildings and sites in multiple languages. Finally, Louisiana state museum, Baton Rouge Louisiana uses codes to link to the exhibits Facebook page, allowing visitors to discuss their experiences of Katrina while exploring the exhibit[5].

2. Uses of QR Codes in Health Centers

QR codes are being used largely in health centers. Hospitals in Japan, Hong Kong and Singapore have adopted QR code with patient information like name, ID number, Date of Birth, Sex and Bed number printed on patient wrist band to identify patients. In Australia QR codes are used for blood test process management. In this case each test tube will be marked with a code and collected blood is put in a test tube then the test tube will be placed on the tester and then the tester automatically examines the blood.

3. Uses of QR Codes in Factory

Factories create a code on their product that contain manufacturing and expiration date of products, a link to product certification which allow customers to confirm certification and products safety instruction and user manual to use the product safely by users.

4. Uses of QR Codes in Education

One of the common areas of using QR code is educational area. They are being used for interactive catalogue display at a library, used to label student assignment submission sheet and for experiment examination purpose. Context aware assessment using QR code [19], an extension of siette project, is developed to take outdoor test to Botany course by using QR code. In [18] the researchers propose how to use QR code to allow users get library instruction and help at a point of need by linking the code with electronic resources, instructional videos, and additional Internet information.

[3] http://www.mattress.org/
[4] http://freilichtmuseum-mariasaal.at/
[5] http://lsm.crt.state.la.us/

5. Uses of QR Codes in Business

QR codes are also used widely in business areas for advertising products to enhance company's income [47]. For example film makers attach a code on the DVD to link with the official film website. Super markets put a code on the price tag of items to be sold. Flyers, business cards and posters will be tagged with QR code to easily connect with organizations.

In addition to the above application areas they can also be used in other sectors such as agriculture, transportation, telecommunication, etc. [47].

In general, QR codes are a powerful tool for providing information to customers anywhere and anytime. They are easy to create, print, distribute and decoded. They can be easily used by small costs than other similar devices.

2.4.2 iQR code

As described in section 2.4.1.4, iQR code is an alternative to existing QR codes developed by Denso Wave. They can be created in square or rectangular formations; this is intended for situations where a rectangular barcode would otherwise be more appropriate, such as cylindrical objects. iQR codes can fit the same amount of information in 30% less space or hold 80% more information than the standard QR code. There are 61 versions of square iQR codes, and 15 versions of rectangular codes. For squares, the minimum size is 9x9 modules; rectangles have a minimum of 19x5 modules.

2.4.2.1 Features of iQR code

As discussed in [13] the characteristics of iQR codes are listed below:

1. High Rate of Information Lacking

iQR Code can hold a greater amount of information than the traditional QR Code. An iQR Code of the same size as an existing QR Code can hold 80% more information than the latter as shown in Figure 2.7. If the same amount is stored, an iQR Code can be made 30% smaller (compared to the regular QR Code) as shown in Figure 2.8.

Figure 2.7: iQR code hold 80% more information than Standard QR code with same size

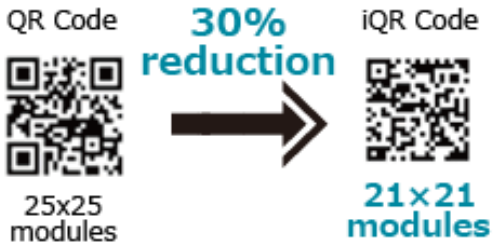

Figure 2.8: iQR code reduces size by 30% with the same information compared to QR code

2. Reduced Size

Using iQR Code, smaller-sized codes than the regular QR Code can be generated. The minimum size consists of 11 by 11 modules for the regular QR Code. On the other hand, it is 9 by 9 modules for the iQR Code. The area required for this code can be reduced by up to roughly 60% compared to the regular code.

3. Use of Rectangular Modules

iQR Code with rectangular modules, as well as square ones, can be generated. Thanks to this feature, it is possible to overwrite it with a space where a barcode is and to print it on cylindrical products while maintaining the code's readability, which is difficult for codes made up of square modules.

4. High Data Capacity

With the iQR Code, one can generate codes that can hold more information than possible with a QR Code. The number of characters that can be stored by the largest version of the QR Code (177 by 177 modules) is about 7,000. iQR Code, on the other hand, can hold about 40,000 characters in its largest version (422 by 422 modules) when the characters are all numerals.

5. High Restoration Capacity

With iQR, one can set a higher restoration capability than with the traditional QR Code. The error correction level is up to 30% of the entire code for the QR Code. On the other hand, this is increased to 50% with the iQR Code.

2.5 Summary

The main essence of ubiquitous computing is to support day to day activities of persons by using available technologies with minimal or no destruction. In case of tour guide system, for example, applications can use different devices and communication technologies to provide information about every tourist attraction. However, the information should be delivered by considering the context of users (tourists) and the environment. For example, applications should deliver services by considering the interest of tourists, and adapt the information by considering the device hold by tourists.

Context aware systems use context to provide relevant information and/or service to users. As defined by Abowd [1] context is information that can be used to characterize the situation of an entity. An entity is a person, place or object that is relevant to the interaction of the system and the user. In context aware systems necessary context information about an entity should be considered. These contexts can be classified in to four. These are: Identity, Location, Environment and Time. Identity context indicates about an entity that communicates with the system. Location context describes the position of user or an entity the user is interested on, Time context indicates the availability of specific service at a given time when the user enters to a service zone. Environment context includes the weather, network availability, etc. of the environment. Providing appropriate service using available contexts is a curtail task for context aware systems.

Different technologies are being used to encode information and retrieve (decode) latter when needed. Among these RFID, Bar Code, and QR codes are commonly used. RFID uses radio frequency electromagnetic fields to transfer data from a tag attached to an object. Bar code represents data about an object in which it is attached on in optical-machine readable form. Information storage capacity of barcodes is a maximum of 20 characters. Quick Response (QR) codes are 2-dimensional barcodes used to encode a small bit of information that can be scanned by mobile devices to decode them. They have the capability of encoding more than 7000 alpha numeric characters.

Chapter 3: Related Work

his Chapter is organized in to four sections. The next two sections (Section 3.1. and Section 3.2) discuss on major studies related to our work. Section 3.1 describes about researches conducted on mobile tourism guide. Works about iQR code and context aware iQR code are presented in section 3.3. The final section gives a summary of related works by providing the gap analysis which is already addressed by this study.

3.1 Works on Mobile Tourism Guide

Research works done on mobile tourism guide can be classified in to two broad categories. The first one is tourist recommendation which provides recommendation about location to visit, nearby restaurant, etc. Recommendation methods can be collaborative, content based or hybrid. The second is an information system which provides a detailed description of each tourist attractions at the point of interest.

The authors in [26, 27, 28] provide tourist recommendation services by using different contextual information in addition to location.

Context-Aware Intelligent Recommendation System for Tourism [26] develops a context-aware tourist recommendation system by using different tourist contexts. It mainly focuses on contexts to be used when making decisions about tourist attractions. Five contextual data: location, time, weather and social media sentiment were used. Hybrid (combination of collaborative, content based and demographic profiling methods) approach is used to make a context-aware intelligent recommendation. For intelligent decision making process, Artificial Neural Network (ANN) model was used. Mainly it handles necessary contexts like time, location, personalization, and it was device independent and scalable but a context modeling was the main issue that is not considered. Proper management of tourist context is the main problem.

Virtual Intelligent System for Informing Tourists (VISIT) [27] focuses on implementing context by building a level of intelligence into tourist based on context aware applications. The application discovers the user preferences implicitly and dynamically updates these preferences with each use of the applications. It used personal, time and weather contexts in addition to location to minimize information overload for tourist recommendation. It implements Bayesian

network model for probability determination. The use of social network allows tourists to share valuable information between them and to have an informed and rewarding tourist experience. However they only build small personal information (like family status and number of families). In addition personal context is not properly modeled.

The research work in [28] provides recommendation service by discovering visitor preferences from visitor's profile, analyzes each preference or proposed heritage to assure if it is liked by the visitor or not. In addition it monitors associating guide service with the analyzed heritages. However it only depends on other visitors rating.

For each heritage they have defined Confidence Level (CL) and this CL value was used to measure acceptance probability of heritages. This is calculated by computing confidence level of every heritage by using Equation 3.1 and recommend heritages that have the highest CL value.

Recommendation Acceptance Case

$$CL = CLp + NF + \alpha + CM$$

Recommendation Rejection Case [3.1]

$$CL = CLp - \alpha + NF$$

Where CL is Confidence Level, CLp is Previous Confidence Level, NF is Negative Factor, CM is CoMment and α is Confidence Level parameter.

Confidence Level (CL): is used to represent the extent to which a recommendation will be accepted. The higher CL value, the greater the possibility of discovered preference acceptance. A set of CL values is mapped one to one with a set of preferences. Its value is in the range 0 to 100. Any CL value higher than 100 or below 0 is converted to 100 or 0 respectively.

Confidence Level Parameter (α): is a constant used as adjustment parameter over the previous CL value. It is added and subtracted from CLp for case of recommendation acceptance and rejection respectively. A constant value 10 is used for increasing 10 percent of CL value about a given heritage.

Negative Factor (NF): holds the existing negative factor associated with the visitor not to accept a given preference approval request. It is used for appreciating or compromising the constant value (α)that is used to update CL.

Negative factors include bad weather condition, visitor will have short visiting time, etc.

They have assigned a minimum value of zero and maximum of value α/2.

CM: is used to represent user suggestion about a given heritage. It is a data source designed to be integer data type of three possible values to represent the satisfaction of a visitor. Possible values are: appreciating (positive), not interested (negative) and non-aligned (neutral).

Context-Aware Tourist Information System (CATIS) [29] is a web service based tourism information system. It is based on the scenario of a tourist is located in main road of Chicago metropolitan area. Since it is noon the system requests restaurant information around the current location of the tourist. The retrieved information considers food preference and economical class of the tourist. Finally, the resulting restaurant list is transformed into Web pages in a format appropriate for display on the client's wireless device. The elements of context in this work were location of a tourist, time of the day, speed and direction of travel, personal preferences, and device type. It describes how these elements are leveraged to adapt Web-based information that is delivered to mobile tourists. It has web services-based system architecture having wireless client device for browsing, application server for coordination, UDDI (Universal Description, Discovery and Integration) server to act as services registry, context manager that manages both dynamic and static contexts, and collection of web-service to provide different services. The architecture of CATIS is shown in Figure 3.1. The links between components in the architecture are numbered so that the sequence of information exchange will be maintained. The application server initially communicates with tourist's device to capture context data and hands to context manager. Then it integrates the output of context manager with the help of UDDI server and tourist information web server to serve the tourist via his/her device. However it doesn't present information about a tourist attraction and it lacks proper modeling of context information.

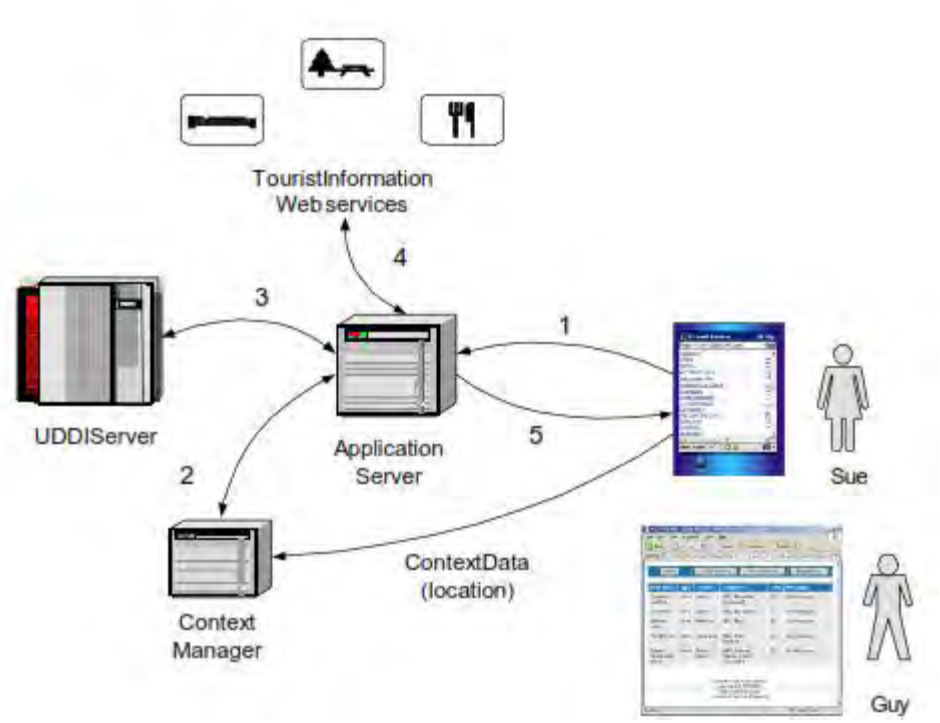

Figure 3.1: Architecture of CATIS

In GUIDE system first the authors gathered requirements from members of staff at Lancaster's Tourist Information Center (TIC) and identified requirements: Flexibility, Context Sensitive information, dynamic information and interactive services then developed a system that provide tourists with up-to-date and context-aware information about a city of Lancaster, United Kingdom (UK), to create tailored tours of the city, to access interactive services, to send and receive messages to/from their companions, to let other members of their group know their location and to leave virtual stick on notes at specific locations in the city so they can share their experiences with other tourists [30]. However it doesn't present information about a tourist attraction and it lacks proper modeling of context information.

3.2 Works on QR Code

In [31] researchers combine QR code with scenic spot's E-Ticketing system. For buying entrance ticket tourists open the official webpage and fill their personal profile like name, id number, cell phone number and number of tickets required. After that they perform online payment and the system generates a QR code to tourists and this generated code is decoded at the time of entrance. However it does not consider the context of users.

The research work in [32] uses QR code to deliver library instructions and help at the point of need. It uses QR codes at the University of Huddersfield Library to split into categories like: links to electronic resources, instructional videos, useful websites for further information, directly containing contact details, and as a way of storing information for future reference. However it does not consider the context of users.

- **QR codes to link to electronic resources-** QR codes are used to link some electronic books, journals and videos to the appropriate physical locations in the library to be accessed by students.
- **QR codes to link to instructional videos-** create and signpost video then link to QR code.
- **QR codes to link to further information on the Internet-** QR codes will be linked to external pages and resources to deliver necessary instructions and information.
- **QR codes to contact the library for further help-**allow users to get appropriate help at the time of need by linking contact information of works in QR code.
- **QR codes to store information for future reference-** QR codes are generated on the library catalogue that link to a version of the live catalogue record for each item. When library users find a useful item in the catalogue, they can scan the QR code and take full, accurate details with them while they look for that item within the physical library.

Context aware assessment using QR code [33] is an extension of system Siette[6] aiming to implement a general mechanism to deliver tests based on mobile devices and matrix codes. The main idea is to attach different QR codes to different places. By coding a question identifier inside a QR code, access to that question can be controlled by the recognition of the corresponding QR code.

By using communication and computing capabilities of tablets and smart phones, it allows to deliver tests not only in a classroom but also in an open environment where students walk around a certain area and the selection of items to be delivered depends on the student's location and circumstances. However, if QR codes are used outdoors, they add the geographical location of the questions, because the user will need to know where they are located in order to get there to answer them. To accomplish these new functionalities the following features are added to Siette

- An easy-to-use authoring tool for the teacher to define a question, associate it to a location and generate the corresponding QR code.
- Access control to each question based on QR code.
- A modification of the question selection mechanism from the item pool, forcing a question to be posed when its QR code is scanned.
- An easy-to-use tool that allows students to find out where the questions are located.

Figure 3.2 shows the sequences of screenshots of web pages presented to a user. Reading from left to right (1) the user scans the QR code containing the reference to the test. (2) The login page is presented. (3) The user is redirected to the test initial page. (4) The test begins, and it shows the waiting page with a map that shows where the questions are located. (5) User scans the code of a question. (6) The question appears in the web browser. (7) The answer is sent to the system and a new waiting page is presented. The previous question has already been answered, so it is removed from the map. (8) The user scans the second QR code. (9) The question is posed and the user sent back its response. The test has been taken using an iPad with a commercial application called "Scan". Note that the order in which questions are posed depends on the track followed by the user [19].

[6] Self-assessment in a feasible, adaptive web-based testing system

Figure 3.2: Context Aware Assessment Using QR Code

The main drawback was it does not implicitly retrieve context information from the user or the device like location, login information, time, etc.

In [34] the researchers defined contextual QR code as: "it's the result of a fusion between a public part of information (QR Code) and a private part of information (the context) provided by the device that scanned the code". Figure 3.3 shows the public and private parts of a contextual QR Code. The private part can be one or more information among the subsequent: user's profile, current task, device used, location, time and environment of the interaction. The machine decodes the QR Code and merges it with private data obtained during the interaction. Then, the resulting XML (Extensible Markup Language) file is sent to a web service (created in the author's laboratory) that computes the code and returns personalized messages. This web service was developed using C#.Net and it is able to retrieve, according to particular tags, the right module to invoke (For example "Hello World" or "Meeting" Applications). It lacks intelligent

decision making, and reusing of contexts, device, environment and task contexts are not added (it is not full context). At last it doesn't disclose how context is obtained.

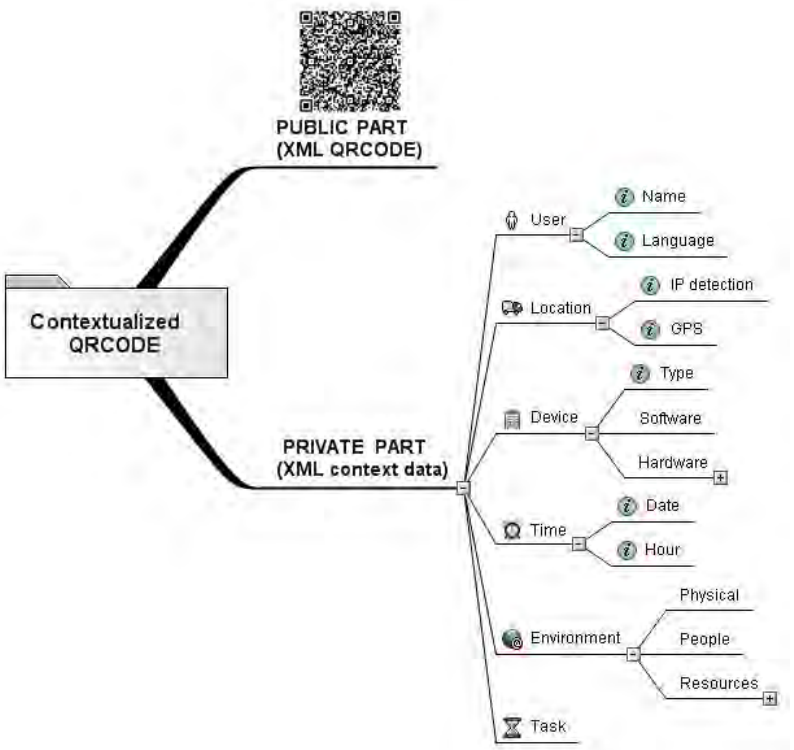

Figure 3.3: Public and Private Parts of a contextual QR Code

The work in [35] presents a new way for QR code deployment. It describes a new mashup which combines 2D barcodes and network proximity data by keeping the basic recognition as is, just to add some parameters on the final stage. In other words, customized QR code scanner will replace encoded value *http://some_domain.com/* with *http://some_domain.com?list_of_parameters*. And this list of parameters will describe user's context.

For adding geographical position of QR codes the encoded URL http*://some_domain.com* will be replaced by some like: *http://some_domain.com?lat=latitude&lng=longitude* where latitude and longitude are calculated values for *user's* location. For indoor deployment, the usage of GPS *based* positioning could be limited. Wi-Fi based collection *positioning* is limited by the need of collecting fingerprints a priori. By this reason network proximity info to the target URL will be passed. It means that the customized QR code scanner creates on the fly Wi-Fi fingerprint and

adds this information to the query string for encoded URL. The fingerprints are a collection of the following triplets:

- SSID name
- RSSI
- MAC-address

However it doesn't show how the context is modeled.

3.3 Summary

Research works in [26, 27] use different contexts for making an intelligent recommendation about tourist attractions. Location, Time, Environment and mainly social media sentiment are used for decision making process.

CATIS [29] provides tourist information by using web services and XML technology. It performs time and location based adaptation to inform about open restaurants, personal adaptation for recommending nearby restaurant and device adaptation to present the information in a suitable format of the device. GUIDE [30], a city guide system, is developed to solve issues faced by tourists while on visiting tourist centers.

The research work in [32] link QR codes to electronic resources, instructional videos, further Internet information, and library or store contact information to get library instruction and help at a point of need. QR code is implemented for assessment purpose in [33]. It allows delivering tests in an open environment. Contextual QR code [34] combines public QR code information and private data for delivering contextual information. Context aware QR code [35] shows a new model for presenting location information based on network, and/or WiFi proximity.

In spite of the aforementioned research works, a lot of work should be done on obtaining, processing and presenting iQR code information to users in a contextual manner depending on the necessary private context. So, this research work presents a new model to provide context aware iQR code information that supports reusability of contexts. In addition, it guides group of tourists by using proximity detection method.

Chapter 4: The Proposed System

4.1 Overview

The purpose of pervasive computing is to create an invisible computing environment that can support user activities with no or minimal destruction. Devices embedded in a pervasive environment require wireless communication capability to ensure the invisibility of the system. In addition, if the system can adapt itself depending on users' behavior and the environment's logical invisibility can be achieved easily. However, this capability is achieved by context awareness, which allows systems to exploit (acquire), represent and reason about a context available in a pervasive environment. Context information can be acquired from multiple sources such as sensors, users, and the environment. Proper representation of acquired contexts allows the reasoner to select and deliver appropriate service to users.

Layer based approach has a number of advantages in a pervasive computing environment. It has mobility, interoperability, greater compatibility and flexibility [36]. Based on this our proposed system is layer based. It has five different layers as shown in Figure 4.1, the first layer, context sensing, comprises context sensor and preprocessing components. The context sensor has multiple sensors used to acquire tourist information from different context sources. Those acquired context information are then preprocessed by the preprocessing unit. In the second layer, context modeling layer, context data management by designing an appropriate model step by step is made. The third layer, service management layer, is responsible for selecting and providing appropriate service depending on the reasoned context. It decodes, processes and maps iQR code data. The output management layer provides an appropriate interface to users. Finally the User Interface layer is used to link the service provided by the system with the user. It allows users to provide static context information and view scanned result of iQR code data. The details of major components in the proposed system are described in the next section.

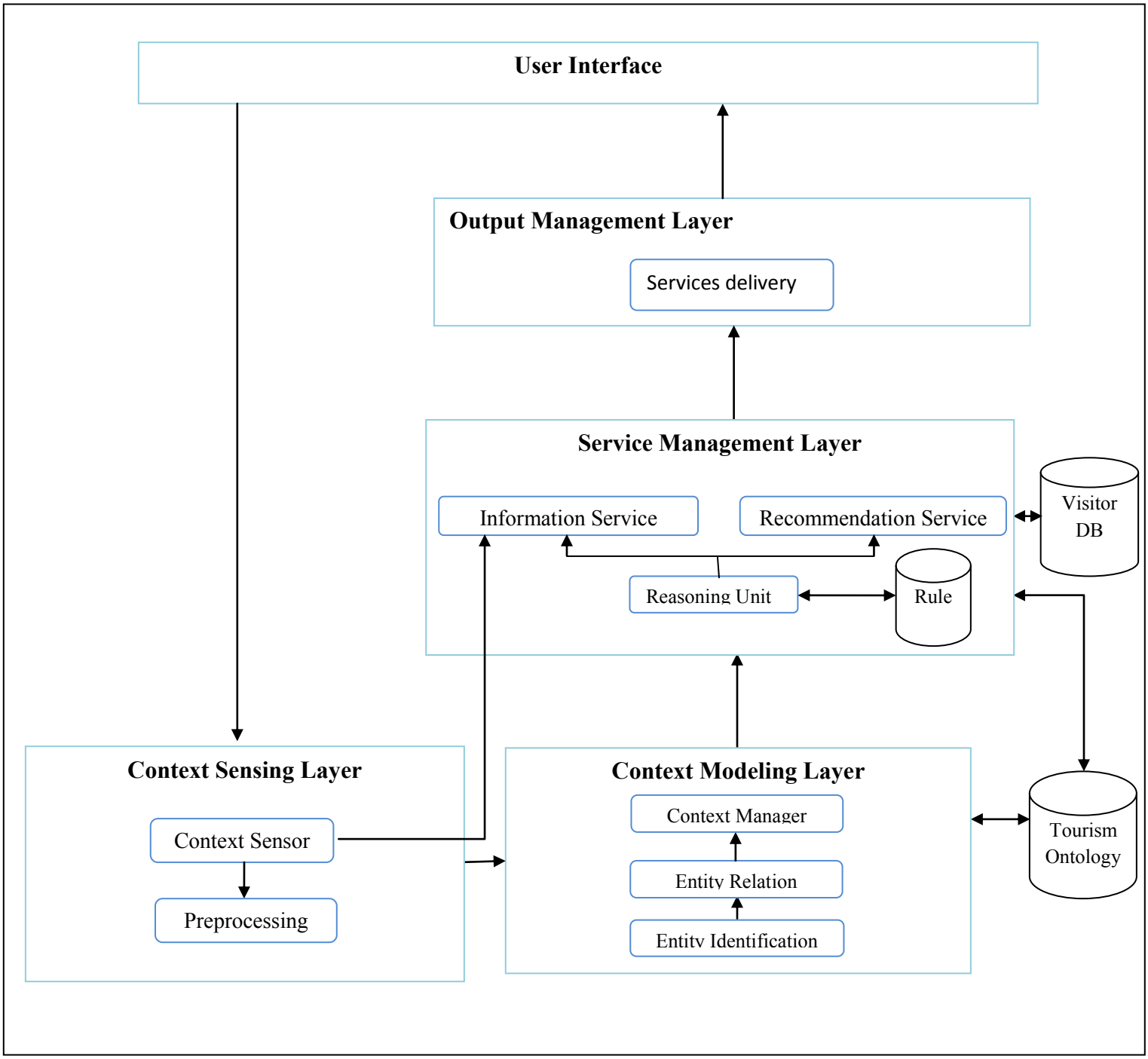

Figure 4.1: Architecture of Context Aware Tourist Information and Recommendation System using iQR code

4.2 Description of Major Components

4.2.1 Context Sensing Layer

Context sensing layer serves as an information collector. It detects context information from different context sources including GPS, iQR code, user profile, environment, etc. preprocess it by removing redundancy and unnecessary computing characteristics and finally presents to the context modeling layer for further action.

a. Context Sensing

In a context aware application useful contexts should be identified and used accordingly. Due to this reason, context sensing is recognized as an important feature for ubiquitous computing. By sensing context information such as location and identity of people and objects, context-enabled applications can present context information to users, or modify their behavior according to changes in the environment.

The proposed system collects dynamic and static contexts from tourists and the environment which are useful to the system. Sensors like GPS sensor and iQR codes are used for location identification. Static personal information (e.g., id, name, schedule, family status, job, personal preference, etc.) are also processed as context for adapting services accordingly. Virtual sources like weather services are used to provide weather information for a given location. In addition wireless connectivity such as Bluetooth and Wi-Fi are used as context sources. By correlating multiple sensor data it is also possible to provide specific information to tourists.

b. Preprocessing

Preprocessing is responsible for checking the correctness of raw data collected from different sensors. It checks for erroneous, null and other data related problems detected and correct them accordingly. It transforms the context data into a format that will be more easily and effectively processed by the context modeling Layer.

The preprocessing unit corrects incorrect values received from sensors. For instance, data received from GPS sensor are corrected if it returns incorrect location information. It also creates a high level representation of context data from low level sensor data. For example, sensed GPS

Position of tourists is mapped to actual position (such as at Gate 1 or Building 1). Incorrect values entered by tourists are also corrected by the preprocessing unit.

4.2.2 Context Modeling Layer

The context modeling layer creates an ontology representation of the context aware tourist information system. The context model captures the current situation that the tourist is acting in, including his/her preferences, interests, social dependencies, and physical and technical environment, and relationships with other entities. It considers four dimensions of contexts such as identity, location, time and environmental contexts of tourists.

Overall, by retrieving unstructured data from the context sensing layer the context modeling layer semantically enriches the data step by step.

a. Entity Identification

Context modeling requires context attributes like personal profile, location, time, etc. to describe the context of the tourism environment. Depending on this in entity identification all entities in a tourism environment are defined and the mapping of sensed data to attributes of entities is defined. Furthermore, this layer is used to specify relations between entities and their contexts, and it defines the model for nested entities.

b. Entity Relation

All entities that are part of the interaction relate to each other in certain ways. On the entity relation layer, dependencies are modeled, so as to express associations like x is with y or x scans y. The entity relationships can be dynamically added and deleted. For example, when a user is moving in a building he/she can have a dynamically instantiated relation to the room entity he/she currently is in.

c. Context Manager

This component is responsible for creating a context knowledgebase (ontology) by using identified entities and relationship of tourists with each other and/or with other entities. Ontology is used to represent knowledge, concepts, and relations about a domain and describes specific situations in a domain. The designed ontology supports the social relation between visitors in a

group tour environment. Compared with other context modeling approaches, ontological context models offer a clear advantage in terms of expressiveness and interoperability [37]. Ontology-based context modeling approaches are strong regarding the distributed composition, partial validation and formality. The formalism of choice in ontology-based models is either OWL [38] or RDF [39]. The syntax of both bases on XML with a focus on describing ontology. OWL was developed by the World Wide Web Consortium (W3C) Web Ontology Working Group used to model domains by defining classes, individuals, data type properties (characteristics of individuals) and object relation.

We have defined the ontology by using OWL. Using OWL for ontology model has the following advantages.

1. It allows necessary semantic interoperability between context-aware systems.
2. It also provides a high degree of inference by making and providing additional vocabulary along with a formal semantics to define classes, properties, relations and axioms.
3. It supports reasoning technique in a better way than any other modeling approaches.

4.2.3 Service Management Layer

The main task of this layer is to select and deliver appropriate service to users depending on reasoned about information. It provides information and recommendation services. Information service is provided by decoding encoded data from iQR code image and map with high level tourist context information to deliver contextual information to users. Recommendation service is based on a hybrid approach. it combines others rating with visitor's rating and preference to recommend POIs. The detail of the above activities is presented in the next sections.

4.2.3.1 Reasoning Unit

Reasoning unit uses predefined inference rules and context data to produce a high level description of context. At this level, beliefs about the tourist behavior, preferences, plans, and other characteristics are acquired through inference and learning. This new information can be useful for context information like the recommendation system during the process of selecting accurate tourist activities or resources for the visitor depending on his/her context.

The reasoning unit provides a reasoning service on the interpreted context information. It deduces new facts based on context information held in repositories, rules provided and ontology stored.

4.2.3.2 Information Service

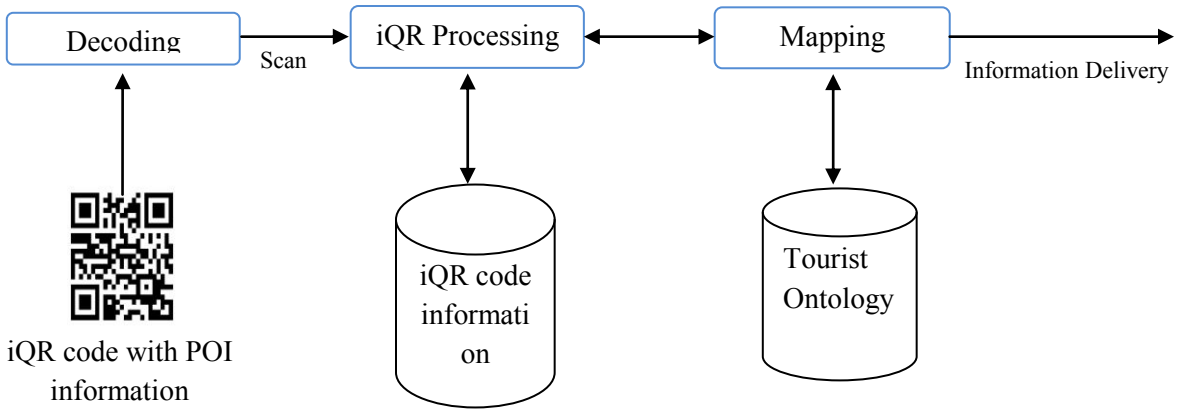

Figure 4.2: Service Manager Model

An information service is used to provide contextual information about POIs in a tourism environment. Information about POIs encoded on iQR codes is decoded and mapped with tourist context to deliver appropriate service. It has three components such as Decoding, iQR Processing and Mapping units. The decoding unit is used to recognize encoded information on iQR code. The iQR processing unit creates an interface to access data decoded from iQR code. Finally, the mapping unit generates the final information that will be displayed to users by using high level context generated by the reasoning unit. Figure 4.2 shows the information service components and their interactions.

a. Decoding

iQR code is a two dimensional barcode in which any text, URL, multimedia, etc. information can be encoded on it and can be printed on white papers, banners, product covers, T-shirts, etc. as an image. In order to recognize information encoded on iQR codes mobile phones should be equipped with camera and a special iQR code scanner application installed on it.

The decoding phase is a reverse process of the encoding procedure. It recognizes information encoded on iQR code and presents to the iQR processing unit. In order to decode iQR code image successfully tourists should align the iQR code inside a frame provided by the decoder, and the system will scan and read the iQR code data as shown in Figure 4.3.

Concepts that can be encoded on iQR code are represented by using OWL ontology. Creating a meaningful representation of every point of interest in ontology allows identify concepts from multiple sources and map them with other contexts to deliver appropriate information about a given POI. For instance, we can select only architectural information about a building if the visitor is an architect. Figure 4.4 shows sample representation of heritage information on iQR code.

Figure 4.3: iQR code Scanning Process

```
<OWL:class  rdf:about=”Empress Zewditu’s Chair”>
   <her:Event>
        <her:title>Taken to Italy</her:title>
        <her:date>1941 </her:date>
   </her:Event>
   <her:Event>
        <her:title>Returned to Ethiopia</her:title>
        <her:date>2000 </her:date>
   </her:Event>
</OWL:class>
```

Figure 4.4: Ontological Representation of Heritage on iQR code

b. iQR Processing

It is responsible for creating and managing ontology obtained from the decoding unit. It accepts query statements from other components and retrieves necessary information, then returns the result for further processing. It adds and removes necessary attribute and values about a POI.

c. Mapping

By using reasoned contexts it selects and presents appropriate service about POI. It selects appropriate attributes that can match the interest of users and presents it.

It uses context information retrieved from users to deliver appropriate service. Personal interests like name, identification, events, history, architecture, etc. about tourist attractions can be filled

when a visitor is first registered in to the system. Those contexts can be used when needed to query appropriate information from POI ontology.

When delivering information about tourist attractions the information should be presented in a way that can be easily understood by visitors. For example, if the encoded information on iQR code describes about an event by taking the event starting date and ending date, the system can present event length and how many years is passed after the event happened.

4.2.3.3 Recommendation Service

We have proposed a hybrid approach to provide a recommendation service to visitors. First the system recommends visitors by calculating the confidence level of POIs and determining the probability of acceptance for each POI. Secondly it analyzes visitors visit history and preference. The detail of a recommendation service is described below.

a. Confidence Level Analysis

This approach is used if the system doesn't build enough visit history about a visitor. We have adopted the research work in [28] for providing preferences to visitors when a tourist first enters to tourist attractions to visit certain POI. It is based on rating of others who have the same profile with the current visitor. It builds confidence level for each tourist attractions and recommends preferences that have the highest rating.

b. Context Dependent Rating (CDR)

The second approach uses personal rating of visited POIs to calculate CDR value and select preferences which are similar with highest rated POIs. It follows three steps. First CDR value of POIs visited by a tourist is calculated by considering Personal Rating, Visitors Location and interest as a parameter as shown in Equation 4.1. Second, visited preferences are ordered in a descending order by using CDR value. Finally, for each of the highest rated POI similar unvisited preferences are selected and delivered to visitors.

$$CDR = R + PM - \mathrm{L}M \qquad [4.1]$$

Where CDR is Context Dependent Rating, R is Visitor Rating, L_M is Location Measurement and P_M is Preference Measurement.

Visitor Rating (R): is personal rating of visited POI. A visitor after visiting the recommended preference he/she can rate POIs out of five i.e., 1 for Strongly Unsatisfied, 2 for Unsatisfied, 3 for Neutral, 4 for Satisfied and 5 for Strongly Satisfied. This value will be calculated with additional contexts to get CDR for all POI.

Location Measurement (L_M): is used to determine how far the distance between the visitor and the identified preference is. To identify location measurement we have calculated the distance between the visitor and the identified preference from the ontology graph.

Preference Measurement (P_M): is a literal value that will check heritage type and personal preference to give advantage for preferred POIs. If visitor's preference is the same as heritage type Pm value will be 5 otherwise Pm will be 3. Algorithm 4.1 shows CDR calculation and ordering process.

//Context Dependent Rating
Input:
Visitor Location (V_L)
Visitor Preference (V_P)
Visited Heritage List (V_H)
Visitor Ratings (V_R)
Varible:
Heritage Type(H_T)
Heritage Location(H_L)
Preference Measurement(P_M)
Process:
For each V_{Hi} in V_H
H_L=getLocaction(V_{Hi})
H_T=getHeritageType(V_{Hi})
If V_P equals to H_T then
P_M=5
Else
P_M=3
End if
P=Parentof(H_L,V_L)
Distance D=Distance(P,H_L) + Distance(P,V_L)
V_R=VisitorRating(V_{Hi})
CDR=V_R+P_M-D
End for
Return(SortDesending (CDR))
Output: List of Heritages ordered in descending using CDR value

Algorithm 4.1: Context Dependent Rating Calculator Module

c. Similarity Identification (Measurement)

After computing CDR value of visited POIs the next step is selecting the first N higher rated heritages and select most similar unvisited items by considering the nearest and heritage type similarity. Algorithm 4.2 shows identification, selection and presentation of preferences to visitors.

```
//similarity identification
Input:
        Visited Heritages(V_H) from algorithm 4.1
        Unvisited Heritages (U_H)
Variables:
        Distance (D)
        Heritage Type (H_T)
        Recommendation List (R_L)
        Similarity List (S_L)
Process:
    For each V_Hi in V_H
        For each U_Hj in U_H
            P=parentOf(V_Hi,U_HJ)
            D=Distance(P,V_Hi)+Distance(P,U_Hj)
            If HeritagType(V_Hi)= HeritagType(V_Hi) then
                H_T=5
            Else
                H_T=3
            End if
            S=H_T-D
            If S>=3
                    SL.add(U_Hj)
                    UH.remove(U_Hj)
             End if
        End for
        SortDessending(S_L)
        RL.add(S_L)
    End for
    Return R_L
Output: Recommendation List
```

Algorithm 4.2: Similarity Identification Module

4.2.4 Output Management Layer

The last layer, output management layer, is where the reactions to context changes are implemented. It provides appropriate interface to users that enable them to view necessary information about a tourist attraction and recommendation services provided by the system. It controls devices that receive the information, and in case of a group visit it dynamically adds and removes a device that delivers services to the intended one.

In case of group visit when group of visitors first register to the system their devices will be paired with each by using Bluetooth connection. When one device scans an iQR code that describes a given POI, the device dynamically identifies all connected devices and disseminates the information to group members who are connected to it.

4.2.5 User Interface

The user interface allows tourists to explicitly interact with the system. It creates appropriate interface to submit static context information about a visitor to the system. In addition, it allows tourists to scan iQR code of POIs and display results back to tourist.

4.3 Context Representation in a Tourism Domain

Customizing information provided by a context aware system in mobile environment is quite crucial since the user is on the move and he/she usually requires very specific information at a given time and place. In the tourism domain, visitors need personalized information about points of interest and nearby activities. This way, it is crucial for services to be able to acquire data about the visitor's context in order to adapt the functionality of the system to the gathered data. However this personalization process is not a trivial task, because data from different context sources have to be acquired and processed by context-aware systems in order to provide the user with the needed information. This processing involves the population of a context data model and its management in order to adapt the system's behavior to the situation of the visitor.

4.3.1 Ontology Representation

Context can be considered as a specific kind of knowledge [40]. Therefore, ontology based approaches represent knowledge, concepts and relationships about a domain and describe specific situations in a domain [41, 42].

The underlying goal of ontology development is to provide common terminologies and rich semantics to enable knowledge sharing and reuse between different systems [38, 43]. Such general vocabulary is an important feature to share information among different pervasive computing systems. Figure 4.5 shows the Context Representation model in heritage environment. Ontology class representation is shown in Figure 4.6 and Annex 4.

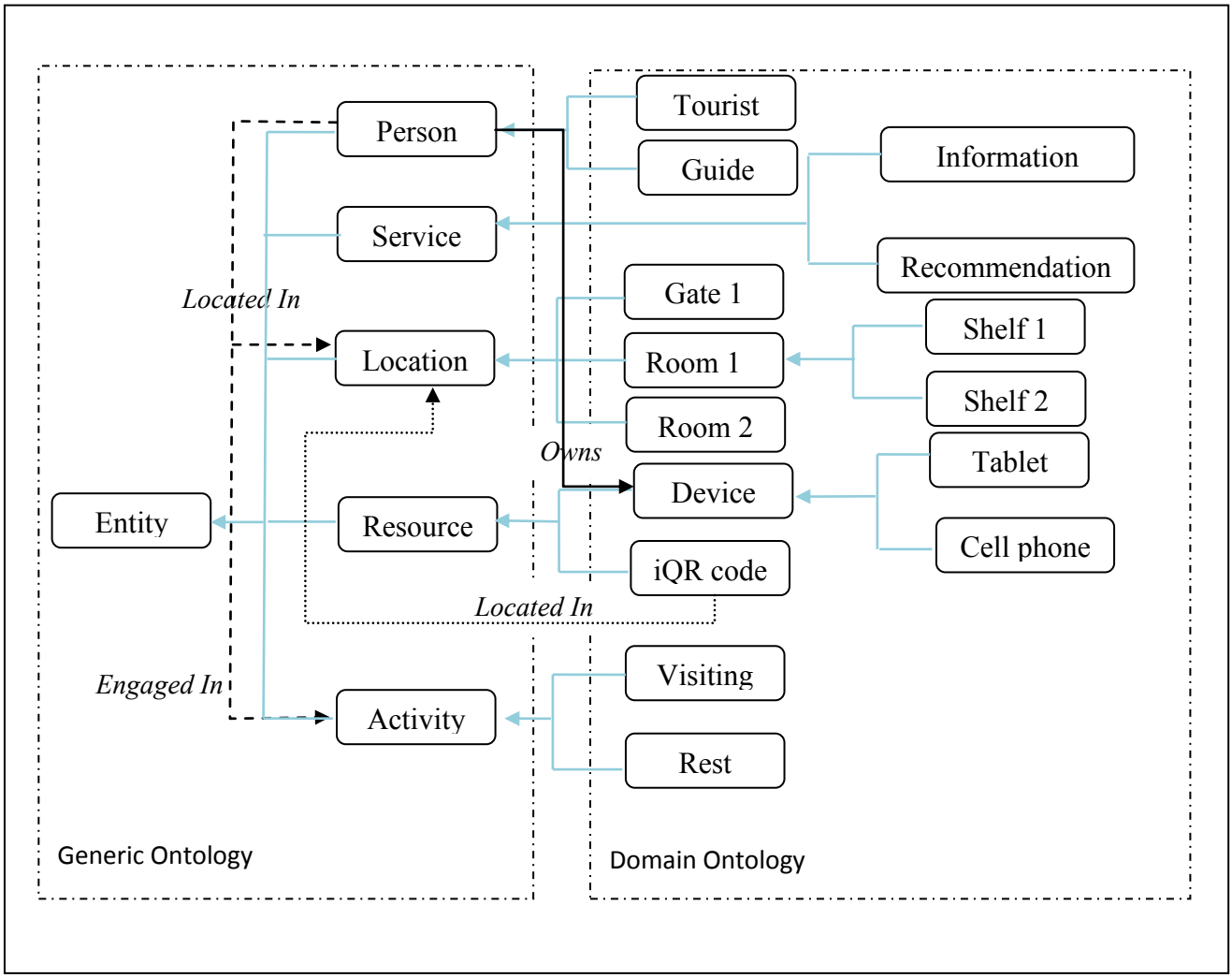

Figure 4.5: Ontology Based Model for Heritage Context Representation

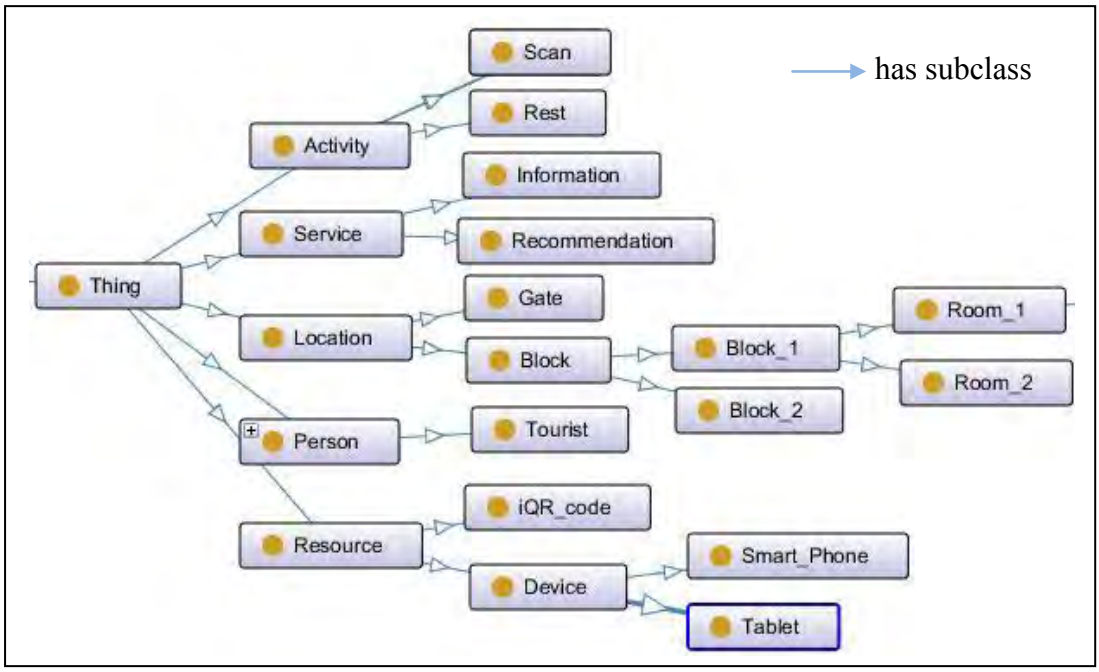

Figure 4.6: Ontology Class Representation for Context Representation Model

4.3.2 Ontology Description

Table 4.1 shows a short introduction about the list of classes used in our tourist information and recommendation system.

Table 4.1: CAMIS Ontology Concepts

No	Concept	Description	Inherits
1	Group	Is an abstract class that holds a collection of tourists that come to visit the tourist center in a group. colleagues, classmate, family, etc	Entity
2	Activity	Represents the task of tourists. like visiting or in recreation.	Entity
3	Person	Is a tourist who joins to visit a historical area. interacts with different classes and the properties of the class ”Person”	Entity
4	Resource	Available resources in a pervasive environment.	Entity
5	Location	abstract representation of the physical place where the person and/or iQR code is located	Entity
6	Service	It holds the information and recommendation services provided by ours system	Entity
7	Device	Any mobile handset equipped in the pervasive environment	Resource
8	iQR code	2D barcode a historical information encoded on it	Resource
10	Block	Is a separate floor in a single building	Location
11	Gate	Is an entrance for visitors to be used to detect visitors to connect or disconnect with CA Environment	Location
12	Smartphone	Is a type of device hold by tourists to scan iQRCode	Device

List of available objects and data properties of the proposed system are listed in Tables 4.2 and 4.3 respectively

Table 4.2: CAMIS Ontolology Object Property

No	Concept	Domain	Range
1	accomplishedBy	Person	Activity
2	accomplishedIn	Location	Activity
3	connectedWith	Device	Device
4	engagedIn	Person	Activity
5	Contains	Shelf	iQR code
6	holdBy	Device	Person
7	locatedIn		Location
8	ownsDevice	Person	Device
10	scaniQRcode	Person	iQR code
11	scannedBy	iQR code	Person

Table 4.3: CAMIS Ontology Data Property

No	Concept	Domain	Range
1	FullName	Tourist	String
2	Id	Tourist	Integer
3	Job	Tourist	String
4	Country	Tourist	String
5	FamilyStatus	Tourist	String
6	Preference	Tourist	String
7	startTime	Activity	Date/Time
8	endTime	Activity	Date/Time
10	locanctionName	Location	String
11	Xcoordinate	Location	Double
12	Ycoordinate	Location	Double
13	roomNumber	Location	Integer
14	shelfNumber	Loction	Integer

Chapter 5: Implementation and Discussion

This Chapter presents implementation detail of context aware tourist information and recommendation system implemented on National Museum of Ethiopia (NME) aiming to present context aware information about heritages available in the Museum. It is classified into five sections. The first Section is an overview of the implementation by presenting two scenarios to show the usability of the proposed system. In the second Section, list of tools and technologies used to develop the prototype are defined and illustrated. In the third Section implementation detail of the system is described with rule definition and description. The last Section provides sample demonstration of the proposed system.

5.1 Overview

Nowadays iQR codes can be efficiently used in tourist centers by incorporating them on every activity of tourists when visiting certain tourist destinations. By encoding multimedia content in addition to textual data visualized information can be presented or they can be used to collect feedbacks from tourists by creating feedback form and encoding it on iQR codes. In general, they are efficient for providing organized and quality information about tourist attractions.

We have selected NME to develop a prototype for the proposed system because it houses a collection of artifacts, art works and archaeological remains found throughout Ethiopia and it is easier to encode information about heritages on iQR codes. We have collected necessary information about heritages by visiting the site and interviewing staff members.

As described above National Museum of Ethiopia houses artifacts, arts, archeological remains of early human beings and faunal remains. It is a three story building. At the basement, the museum houses archeological findings of early species including the most known human skeleton, Lucy. The first floor holds artifacts, dresses, and sculptures collected from all corners of Ethiopia. The second floor shows the traditional and modern art works of well known artist in Ethiopia.

The following two scenarios show how to provide heritage information to an individual or group of visitors in a context aware museum environment.

Scenario 1:

1. Bob is an architect at XYZ Construction Company.
2. During his free time he wants to visit heritages at National Museum of Ethiopia.
3. At 2 PM he arrives at the museum.
4. When the system detects Bob at Gate 1 it displays a welcome screen and display personal profile submission form.
5. Bob fills his personal profile and submits to the system.
6. The system display list of heritage by looking his personal profile and heritage rating history. He starts visiting heritages.
7. When he scans the first heritage iQR code, the heritage's information is compiled and presented to him on a context aware manner. Then the system adds the heritage to visit list by accepting his personal rating.
8. Bob wants to see the list of available heritages near to him.
9. The system displays all nearest available heritages that are not visited by him.
10. After Bob finishes visiting by the help of CAMIS, he disconnects with the system and leaves the museum.

Scenario 2:

1. NTO tour and travel agency organized a group tour service having 20 members in a group.
2. Alice is a member and leader of the group.
3. When the tourists arrive to National Museum of Ethiopia, Alice submits group information to CAMIS system and a group profile is created.
4. At one of the heritages when Alice scans the iQR code information about the heritage, it will be delivered to all nearest members in the group.
5. The group visits all heritages by the help of CAMIS.
6. The group finishes the visit, disconnects with the system and leaves the museum.

5.2 Tools and Technologies Used

Tools and technologies used to develop the prototype are listed below:

- Android SDk[7]: is used to write the implementation of the work. The Android SDK provides the API libraries and developer tools necessary to build, test, and debug apps for Android. It comes with two parts: the base tools, and version-specific SDKs and related add-ons. The version used is 4.2
- Java programming language: When we are writing Android applications, we typically write them in Java source code. That Java source code is then turned into a format that Android actually runs (Dalvik bytecode in an Android package [APK] file). The Java™ Programming Language is a general-purpose, concurrent, strongly typed, class-based object-oriented language. It is designed to be platform independent with additional key principles such as usability, reliability and security. The version used is JDK 8.
- Eclipse IDE (Integrated Development Environment): Eclipse is a free open source IDE (Integrated Development Environment). It support for Java, HTML, CSS, JavaScript, C++, PHP, and more. Google has free Eclipse plugin to integrate with the Android SDK. The version Used is 4.2
- Jena[8] is an open source Java framework for building semantic web applications. It provides a programmatic environment for RDF, RDFS and OWL, SPARQL and includes a rule-based inference engine. We have used the Jena API for reasoning on user's activity by combining the ontology and rules of the proposed system.
- SQLite: is an open source Database Management System that comes with Android. It used to store and manage static context of tourists.
- Protégé: is used to construct the ontologies required in the prototype. It is a free, opensource platform that provides a suite of tools to construct domain models and knowledge-based applications with ontologies. It has ontology editing environment with support for the OWL 2 Web Ontology Language, and direct in-memory connections to description logic reasoners like Pellet.
- SPARQL- SPARQL is used to query the information held in the ontology modeled for the prototype. It is a data-oriented query language and a protocol for accessing RDF.

[7] http://developer.android.com/index.html
[8] http://jena.sourceforge.net/ - last date accessed Aug -17/2009

5.3 Implementation Detail

The proposed system is implemented as client-server architecture. For the client side we have developed a CAMIS Android based application using android SDK that can run on Android devices and the server side application is developed using Java programming language and will run on windows 7.

5.3.1 Client Side Implementation

The client side is implemented in Android development environment. Its target SDK is android API level 21 and the minimum SDK is API level 8. We have tested the application on Samsung Galaxy S4 Smartphone. Components that run on the client side are summarized and presented in Table 5.1.

Table 5.1: CAMIS Client Side Implemented Modules

Major Components	Description
Context Sensor	It is implemented as an Android activity class. It accepts static and dynamic context necessary to the system. • For outdoor user location detection GPS sensor is used • For explicitly accepting static contexts a separate activity is created.
iQR code Scanner	Is implemented as a separate Android activity. It uses Zxing (Zebra crossing) library to decode information encoded on iQR codes.
Context DB	Implements SQLite database for storing personal profile collected from tourists.
Query Execution	This module uses SPARQL query for executing necessary query statements.
Recommendation	It is a separate class that implements recommendation algorithms designed in Chapter 4.
User Interface	Is implemented as a separate Android activity. It is used to display the final result about POIs to the scanner device and/or nearest devices.

5.3.2 Server Side Implementation

The server side programming is implemented by using Java Programming Language. Table 5.2 illustrates server side components implemented in the prototype.

Table 5.2: CAMIS Server Side Implemented Modules

Major components	Description
Confidence level Manager	It is implemented as a separate Java class. It calculates CL value of every heritage by accepting users rating and it returns highest rated heritages.
Group Profile Manager	Location of each member in a group visit is managed by this module and used to return nearest members to a device that scans iQR code.

Ontology Rules

Figure 5.1 shows user defined rules for heritage information scenario. Sample group ontology representation is show in Figure 5.2.

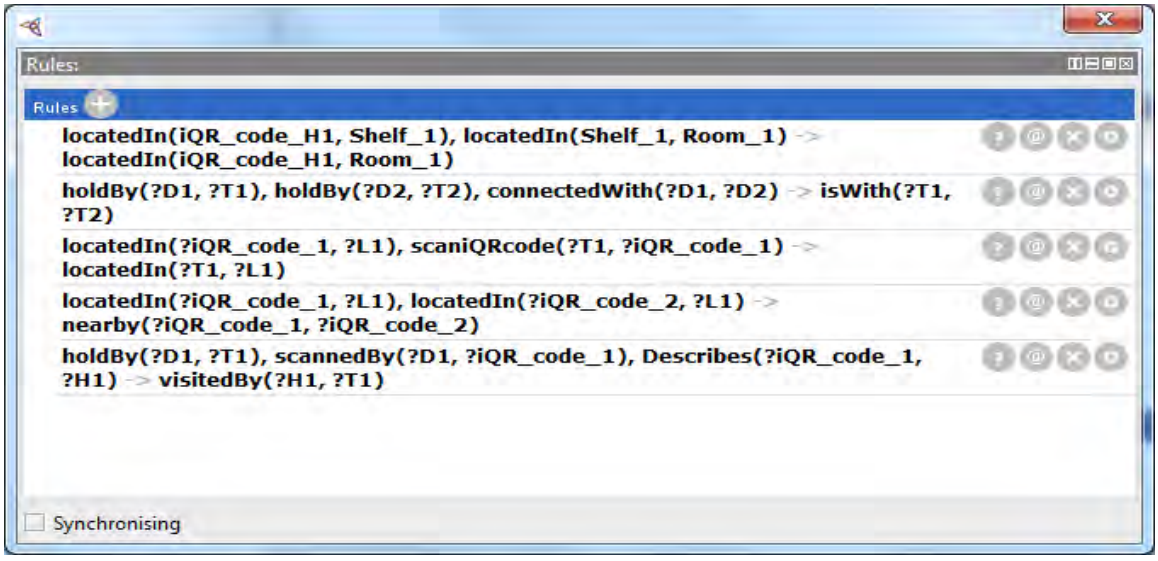

Figure 5.1: Sample Ontology Rules Implemented in CAMIS

```
<rdf:RDF
      xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
      xmlns:rdfs="http://www.w3.org/2000/01/rdf-schema#"
      xmlns:foaf="http://xmlns.com/foaf/0.1/"
      xmlns:admin="http://webns.net/mvcb/"
      xmlns:res="http://EMIS.org/Group#">
  <foaf:Group rdf:id="G1">
       <foaf:name>Group1</foaf:name>
       <foaf:member>
              <foaf:Person rdf:id="G1M1">
                     <foaf:name>Hassen</foaf:name>
                     <foaf:phone>0918818181</foaf:phone>
                     <res:LocatedIn>Bldg1</res:LocatedIn>
              </foaf:Person>
       </foaf:member>
       <foaf:member>
              <foaf:Person rdf:id="G1M2">
                     <foaf:name>Abdu</foaf:name>
                     <foaf:phone>09131313</foaf:phone>
                     <res:LocatedIn>Bldg2</res:LocatedIn>
              </foaf:Person>
       </foaf:member>
              <foaf:member>
              <foaf:Person rdf:id="G1M3">
                     <foaf:name>Maki</foaf:name>
                     <foaf:phone>09111111</foaf:phone>
                     <res:LocatedIn>Bldg2</res:LocatedIn>
              </foaf:Person>
       </foaf:member>
   </foaf:Group>
  <foaf:Group rdf:id="G2">
       <foaf:name>Group1</foaf:name>
       <foaf:member>
              <foaf:Person rdf:id="G2M1">
                     <foaf:name>Tedy</foaf:name>
                     <foaf:phone>09111111</foaf:phone>
                     <res:LocatedIn>Bldg1</res:LocatedIn>
              </foaf:Person>
       </foaf:member>
       <foaf:member>
              <foaf:Person rdf:id="G2M2">
                     <foaf:name>Sami</foaf:name>
                     <foaf:phone>09141414</foaf:phone>
                     <res:LocatedIn>Bldg2</res:LocatedIn>
              </foaf:Person>
       </foaf:member>
   </foaf:Group>
 </rdf:RDF>
```

Figure 5.2: Sample Ontology to Represent Group of Visitors

5.4 Demonstration and Evaluation

In CAMIS system context are acquired either statically or dynamically. When tourist/s first enter/s to a CAMIS environment the system displays personal profile submission form to be filled by visitors. As we describe in Annex 1 and Figure 5.3 a personal profiles will be submitted to the system to be used for identifying personal preferences of tourists. Figure 5.3 b and Annex 2 shows the main activity of CAMIS system used to scan the iQR code and display the scanned result.

In addition to static contexts necessary information about a tourist are collected dynamically. For example, location of a tourist will be dynamically acquired using scanned iQR code and GPS system.

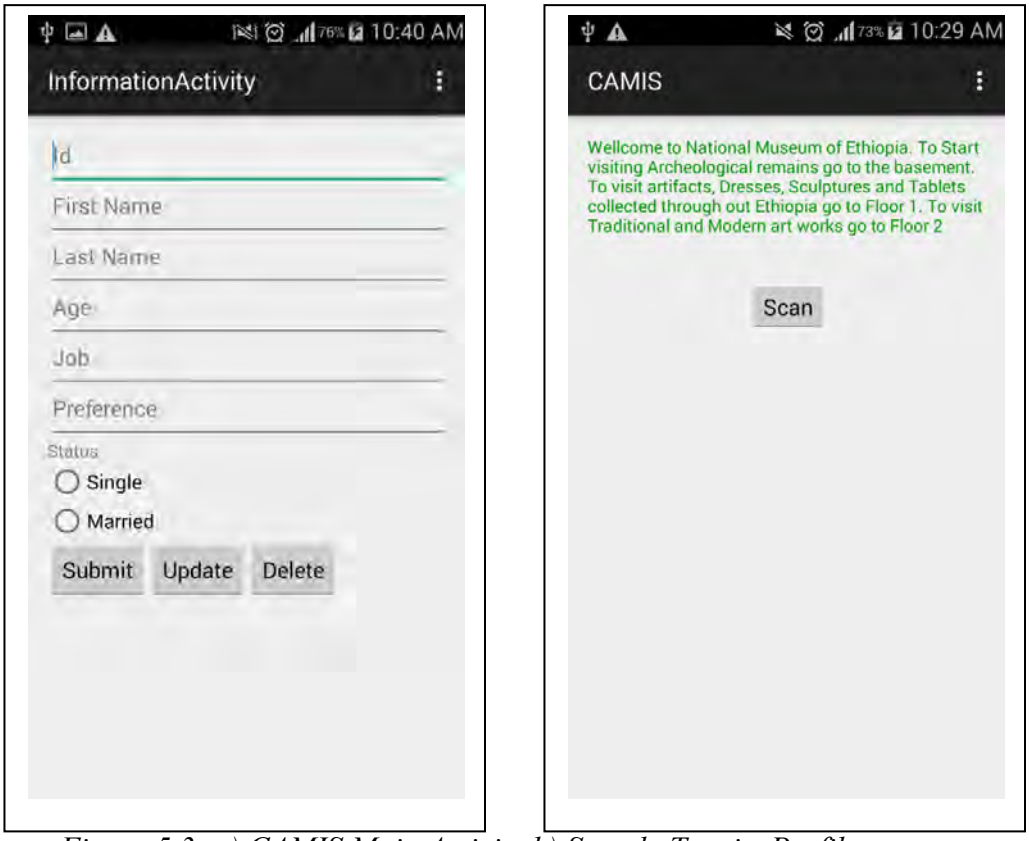

Figure 5.3: a) CAMIS Main Activity b) Sample Tourist Profile

Context aware description about heritage is displayed to tourists by analyzing information encoded on iQR code and displayed in a suitable manner.

When tourists are visiting certain heritages it is not necessary to scan the given heritage's iQR code by all members. By contextually analyzing the nearest member, if a single person (say the group leader) scans the code it can be displayed to every member in a group through their Smartphone or tablet.

For context aware group visit if anyone wants to see the nearest group members current location of a member will be detected and the nearest will be displayed as shown in Figure 5.3. Annex 3 shows how to retrieve OWL file and query the necessary statements.

Figure 5.4: List of Identified Nearest Group Members

Recommendation service is provided by the system by calculating interest of every visitor visit history and preference. The recommendation service is provided by considering current visitor location, his/her interest and nearby heritages. Figure 5.4 shows a recommendation service of CAMIS.

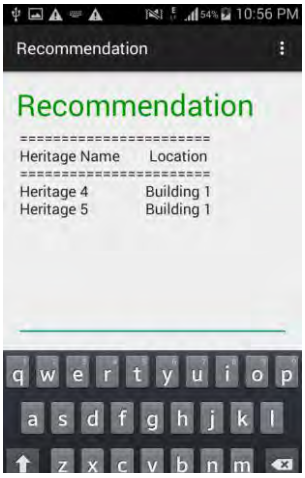

Figure 5.5: List of Available Heritage Preferences

Chapter 6: Conclusions and Future Works

6.1 Conclusions

Tourists visit certain historical area with little or no information about the tourist attraction they are going to visit. Organized information about a tourist attraction can be delivered to visitors by using technologies like RFID, QR codes and iQR codes.

In this study, we have proposed and developed a model for context aware tourist information and recommendation system using iQR codes. It provides information about Point of Interests by combining public information encoded on iQR code and private information (such as identity and personal interest) acquired from tourists and the environment. It also adapts service delivery in a group visit environment. In addition, recommendation services are provided by the system using visitor's history, visitors' interest, location and additional context information.

In order to show the usability of our system we have developed a prototype for National Museum of Ethiopia. In the prototype we have implemented the major components including acquiring necessary context information from visitors, managing acquired context and additional context information, decoding module and service delivery modules.

6.2 Contributions

To present information about Point of Interests encoded on iQR code in a context aware manner we have developed a new model that can map private and public context information to deliver appropriate service to visitors. The contributions of this research work are summarized as follows:

- Propose an architecture that maps tourist context and iQR code data to provide context based tourist information system.
- Designed a new way for group of tourists to visit tourist centers without any distraction.
- Designed a hybrid approach for recommending unvisited POI in a certain tourist attraction.

6.3 Future Works

The potential future works as a continuation of this work are listed below:

- To show applicability of the proposed system we have developed a prototype. However, it can be implemented in tourism environment to enhance the guiding system of tourist centers.
- The prototype of this research work is developed in heritage environment, it can be extended to other tourist centers like for city guide, and other historical areas.
- We have implemented the research work by encoding textual information in iQR codes so this can be further extended by encoding multimedia contents on iQR codes to provide visual information about a tourist attraction.
- Securing context information retrieved from Tourists and iQR codes can also be considered as a potential future work.

References

[1] Abowd and George D., "Software Engineering Issues for Ubiquitous Computing", in Proceedings of the 1999 International Conference on Software Engineering, pp.75–84, 22-22 May 1999.

[2] G. Chen, Guanling, and D. Kotz, "A Survey of Context-aware Mobile Computing Research", Technical Report TR2000-381, Department of Computer Science, Dartmouth College, Vol. 1, No. 2.1, 2000.

[3] Anagnostopoulos, Christos B., A. Tsounis, and S. Hadjiefthymiades, "Context Awareness in Mobile Computing Environment", Wireless Personal Communication, Vol. 42, pp. 445-464, 2007.

[4] W. Schwinger, C. Grün, B. Pröll, W. Retschitzegger, and A. Schauerhuber, "Context-Awareness in Mobile Tourism Guide- Comprehensive Survey", Rapport Technique, Johannes Kepler University, Linz, Austria, 2005.

[5] "Smartphone users around the world-statistics and facts[infographics]", Available at: http://www.go-gulf.com/blog/smartphone/ (Accessed on 10, February 2015).

[6] "ethio telecom CEO Media briefing at 2004 EFY Performance related press conference", Available at: https://www.ethiotelecom.et/press/news.php?id=74 (Accessed on 07, November 2013).

[7] "QR code How-To Guide", the Association of Nova Scotia Museums, 2010.

[8] "Understandig QR Codes", Available at: http://marksprague.wordpress.com/qr-codestechnology/understanding-qr-codes/ (Accessed on 10, February 2015).

[9] T. Lotlikar, R. Kankapurkar, A. Parekar and A. Mohite, "Comparative Study of Barcode, QR-code and RFID System", International Journal of Computer Technology and Applications, Vol. 4 (5), pp. 817-821, 2013.

[10] "QR Code for Tourism in Wales", Available at: http://www.youtube.com/watch-v=vJQ4-YAP6r0 (Accessed on 10, February 2015).

[11] "India Tourism Investing in QR Codes to Aid Tourists", Available at: http://skift.com/2013/10/07/india-tourism-investing-in-qr-codes-to-aid-tourists/(Accessed on 10, February 2015).

[12] "South Korea Boosts Tourism Qr Codes", Available at: http://beqrious.com/south-korea boosts-tourism-with-qr-codes/ (Accessed on 15, November 2013).

[13] "iQR Code", Available at: http://www.qrcode.com/en/codes/iqr.html (Accessed on 10, February 2015).

[14] G. Fischer, "Context-Aware Systems The 'Right' Information, at the 'Right' Time, in the 'Right' Place, in the 'Right' Way, to the 'Right' Person", AVI '12, Capri Island, Italy, 21-25 May 2012.

[15] L. Buriano, "Exploiting Social Context Information in Context-aware Mobile Tourism Guides", Telecom Italia - Innovation & Engineering Services, Reiss Romoli 274, 10148 Torino, Italy, 2006.

[16] M. Weiser, "The Computer for the Twenty-First Century", Scientific American, 265(3), pp. 94-100, September 1991.

[17] "Context", Available at: http://www.dictionary.com (Accessed 10, February, 2015).

[18] Schilit, Bill, N. Adams, and R. Want, "Context-aware Computing Applications", in Proceedings of the Workshop on Mobile Computing Systems and Applications, IEEE Computer Society, pp. 85–90, December 1994.

[19] Dey and Anind K., "Understanding and using context", Personal and Ubiquitous Computing Journal, Vol. 5, No. 1, pp. 4–7, 2001.

[20] Picard and R.W., "Affective Computing", MIT Press, 2000.

[21] K. Barrett and R. Power, "State of the Art: Context Management", M-Zones Research Program, pp.69-87, 2003.

[22] M. Satyanarayanan, "Challenges in Implementing a Context-Aware System", IEEE Pervasive Computing, Vol. 1, no. 3, pp. 2-2, 2006.

[23] Want, Roy and A. Hopper, Veronica Falcao, and Jonathan Gibbons, "The Active Badge Location System", ACM Transactions on Information Systems (TOIS), Vol. 10, No. 1, pp. 91-102, 1992.

[24] Dejene Ejigu, "Context Modeling and Collaborative Context-Aware Services for Pervasive Computing", National Institute of Applied Sciences (INSA de Lyon), 2007.

[25] Wang, X. Hang, D. Zhang, T. Gu, and H. Pung, "Ontology Based Context Modeling and Reasoning using OWL", In Pervasive Computing and Communications Workshops, 2004. Proceedings of the Second IEEE Annual Conference on, pp. 18-22, 2004.

[26] K. Meehan, T. Lunney, K. Curran, and A. McCaughey, "Context-Aware Intelligent Recommendation System for Tourism", Pervasive Computing and Communications Workshops (PERCOM Workshops), 2013 IEEE International Conference, pp 328 – 331, 2013.

[27] K. Meehan, T. Lunney, K. Curran, and A. McCaughey, "VISIT: Virtual Intelligent System for Informing Tourists", PGNET, Liverpool, 2012.

[28] Betelhem Teka, "Modeling Pervasive Context-aware Museum Guide Service", Unpublished MSC Thesis, Addis Ababa University, Addis Ababa, Ethiopia, December 2009.

[29] A. Pashtan, R. Blattler, A. Heusser, and P. Scheuermann, "CATIS: A Context-aware Tourist Information System", June 2003.

[30] K. Cheverst, N. Davies, K. Mitchell, A. Friday and C. Efstratiou, "Developing a Context-aware Electronic Tourist Guide: Some Issues and Experiences", In Proceedings of the SIGCHI Conference on Human Factors in Computing Systems, pp. 17-24, 2000.

[31] M. Zhang, D. Yao and Q. Zhou, "The Application and Design of QR Code in Scenic Spot's eTicketing System-A Case Study of Shenzhen Happy Valley", International Journal of Science and Technology, Vol 2, No. 12, December 2012.

[32] A. Walsh, "QR Codes – Using Mobile Phones to Deliver Library Instruction and Help at the Point of Need", Journal of Information Literacy, Vol. 4, No. 1, pp 55-64, June 2010.

[33] R. Conejo, J. Perez-de-la-Cruz, B. Barros and J. Galvez, " Context-aware Assessment Using QR Codes", Proceedings Research in Engineering Education Symposium, Kuala Lumpur, 2013.

[34] J. Rouillard, "Contextual QR Codes", In Computing in the Global Information Technology, 2008. ICCGI'08. The Third International Multi-Conference on IEEE, pp. 50-55, 2008.

[35] D. Namiot, M. Sneps-Sneppe, and O. Skokov, "Context-aware QR codes", World Applied Sciences Journal, Vol. 25, No. 4, pp 554-560, 2013.

[36] "List-the-major-advantage-of-layered-based-approach", Available at: http://wiki.answers.com/Q/List-themajor-advantages-with-layered-approach-toprotocols (Accessed on 10, February 2015).

[37] T. Strang and C. Linnhoff-Popien, "A Context Modeling Survey", In: Workshop on Advanced Context Modelling, Reasoning and Management, UBICOMP 2004 – The Sixth International Conference on Ubiquitous Computing, Nottingham, England, 2004.

[38] "OWL Web Ontology Language Guide", Available at: http://www.w3.org/TR/owl-guide/ (Accessed 10, February, 2015).

[39] "Resource Description Framework (RDF): Concepts and Abstract Syntax", Available at: http://www.w3.org/TR/rdf-concepts/ (Accessed 10, February, 2015).

[40] C. Bettini, O. Brdiczka, K. Henricksen, J. Indulska, D. Nicklas, A. Ranganathan, and D. Riboni, "A survey of Context Modeling and Reasoning Techniques", Pervasive and Mobile Computing, Vol. 6, No. 2, pp. 161–180, 2010.

[41] X. Ying and X. Fu-yuan, "Research on Context Modeling Based on Ontology", In Computational Intelligence for Modelling, Control and Automation, 2006 and International Conference on Intelligent Agents, Web Technologies and Internet Commerce, pp. 188-188, December 2006.

[42] M. Uschold and M. Gruninger, "Ontologies: Principles, Methods and Applications", the Knowledge Engineering Review, Vol. 11, No. 02, pp. 93–136, 1996.

[43] J. Ye, L. Coyle, S. Dobson, and P. Nixon, "Ontology-based Models in Pervasive Computing Systems", The Knowledge Engineering Review, Vol. 22, pp. 315–347, December 2007.

[44] J. Krumm, "Ubiquitous Computing Fundamentals", Microsoft Corporation, Redmond, Washington, USA, 2010.

[45] "QR code," Available at: en.wikipedia.org/wiki/QR_Code (Accessed 24, March, 2015).

[46] "What is QR Code," Available at: www.qrcode.com/en/about/ (Accessed 24, March, 2015).

[47] "QR codes," Available at: https://www.foxdesignsstudio.com/uploads/pdf/three_QR_-Code.pdf (Accessed 24, March, 2015).

[48] "QR Code Model 1 and Model 2," available at: https://www.qrcode.com/en/codes/-model12.html (Accessed 24, March, 2015).

[49] "Micro QR Code," Available at: https://www.qrcode.com/en/codes/microqr.html (Accessed 24, March, 2015).

[50] "SQRC", Available at: https://www.qrcode.com/en/codes/sqrc.html (Accessed 24, March, 2015).

[51] "LoqoQ", Available at: https://www.qrcode.com/en/codes/logoq.html (Accessed 24, March, 2015).

Annexes

Annex 1. Sample Code Segments

```
<!-- Layout definition for accepting user profile from Tourist -->
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:orientation="vertical"
    android:paddingBottom="@dimen/activity_vertical_margin"
    android:paddingLeft="@dimen/activity_horizontal_margin"
    android:paddingRight="@dimen/activity_horizontal_margin"
    android:paddingTop="@dimen/activity_vertical_margin"
    tools:context="com.aau.camis.InformationActivity" >
    <EditText
          android:id="@+id/etxtId"
          android:layout_width="match_parent"
            android:layout_height="wrap_content"
            android:hint="Id" />
    <EditText
          android:id="@+id/etxtFname"
          android:layout_width="match_parent"
            android:layout_height="wrap_content"
            android:hint="@string/fname" />
      <EditText
          android:id="@+id/etxtMname"
          android:layout_width="match_parent"
            android:layout_height="wrap_content"
            android:hint="@string/lname" />
      <EditText
          android:id="@+id/etxtAge"
          android:layout_width="match_parent"
          android:layout_height="wrap_content"
          android:hint="@string/age"
          android:inputType="number"
          android:maxLength="3" />
      <EditText
          android:id="@+id/etxtJob"
          android:layout_width="match_parent"
            android:layout_height="wrap_content"
            android:hint="@string/job" />
        <TextView
          android:id="@+id/textView1"
          android:layout_width="match_parent"
            android:layout_height="wrap_content"
            android:hint="@string/status"   />
       <RadioGroup
                android:layout_width="match_parent"
                android:layout_height="wrap_content">
                <RadioButton
                    android:id="@+id/rdoSingle"
                    android:layout_width="match_parent"
                    android:layout_height="match_parent"
                    android:text="@string/single" />
```

```
                    <RadioButton
                        android:id="@+id/rdoMarried"
                        android:layout_width="match_parent"
                        android:layout_height="match_parent"
                        android:text="@string/married" />
                </RadioGroup>
            <Button
                android:id="@+id/btnSubmit"
                android:layout_width="wrap_content"
                android:layout_height="wrap_content"
                android:text="@string/submit" />
             <Button
                android:id="@+id/btnUpdate"
                android:layout_width="wrap_content"
                android:layout_height="wrap_content"
                android:text="Update" />
         <Button
                android:id="@+id/btnDelete"
                android:layout_width="wrap_content"
                android:layout_height="wrap_content"
                android:text="Delete" />
</LinearLayout>
```

```
package com.aau.camis;
import android.support.v7.app.ActionBarActivity;
import android.content.Intent;
import android.os.Bundle;
import android.view.Menu;
import android.view.MenuItem;
import android.view.View;
import android.widget.Button;
import android.widget.EditText;
import android.widget.RadioButton;
public class InformationActivity extends ActionBarActivity {
       Button btnAdd,btnUpdate, btnDelete;
       EditText txtFName, txtMName, txtAge, txtJob,txtId;
       RadioButton rdoSingle, rdoMarried;
       String Fname,Lname,job,status,age,id;
       @Override
       protected void onCreate(Bundle savedInstanceState) {
              super.onCreate(savedInstanceState);
              setContentView(R.layout.activity_information);
              btnAdd =(Button) findViewById(R.id.btnSubmit);
              btnUpdate=(Button) findViewById(R.id.btnUpdate);
              btnDelete=(Button) findViewById(R.id.btnDelete);
              txtFName =(EditText) findViewById(R.id.etxtFname);
         txtMName =(EditText) findViewById(R.id.etxtMname);
         txtAge =(EditText) findViewById(R.id.etxtAge);
         txtJob =(EditText) findViewById(R.id.etxtJob);
         txtId=(EditText) findViewById(R.id.etxtId);
         rdoSingle=(RadioButton) findViewById(R.id.rdoSingle);
         rdoMarried=(RadioButton) findViewById(R.id.rdoMarried);
```

```
 btnAdd.setOnClickListener(new View.OnClickListener() {
       @Override
       public void onClick(View arg0) {
       // Button Submit Click Listener
              assignValue();
              SQLiteDB sqliteobject=new SQLiteDB(InformationActivity.this);
              sqliteobject.addData(id,Fname, Lname, age, job,
status,InformationActivity.this);
                           clear();
              } });
  btnUpdate.setOnClickListener(new View.OnClickListener() {
       @Override
       public void onClick(View arg0) {
       // Button Update Click Listener
              assignValue();
              SQLiteDB sqliteobject=new SQLiteDB(InformationActivity.this);
              sqliteobject.updateRecord(id,Fname, Lname, age, job,
                     status,InformationActivity.this);
              clear();
              } });
 btnDelete.setOnClickListener(new View.OnClickListener() {
                     @Override
                     public void onClick(View arg0) {
                           // Button Delete Click Listener
                           id=txtId.getText().toString();
                           SQLiteDB sqliteobject=new
SQLiteDB(InformationActivity.this);
                           sqliteobject.deleteData(id, InformationActivity.this);
                           clear();
                     } });
       }
public void assignValue(){
       Fname=txtFName.getText().toString(); Lname=txtMName.getText().toString();
       job=txtJob.getText().toString(); age=txtAge.getText().toString();
       id=txtId.getText().toString();
       if (rdoSingle.isChecked()){
              status="Single";
       }
       else{
              status="Married";
       }      }
public void clear(){
    txtFName.setText(""); txtMName.setText(""); txtAge.setText("");
txtJob.setText("");
    txtId.setText(""); rdoSingle.setChecked(false); rdoMarried.setChecked(false);
}
@Override
public boolean onCreateOptionsMenu(Menu menu) {
              getMenuInflater().inflate(R.menu.information, menu);
              return true;
       }
       @Override
       public boolean onOptionsItemSelected(MenuItem item) {
              int id = item.getItemId();
              if (id == R.id.action_settings) {
                     Intent i=new Intent(InformationActivity.this,MainActivity.class);
                     startActivity(i);
              }
```

```
package com.aau.camis;
import android.content.Context;
import android.database.Cursor;
import android.database.sqlite.SQLiteDatabase;
import android.database.sqlite.SQLiteException;
import android.widget.Toast;
//SQLite Database
public class SQLiteDB {
	SQLiteDatabase db=null;
	Cursor c;
	Toast T;
	int counter;
	public SQLiteDB(Context cont){
		db=cont.openOrCreateDatabase("MYDB",Context.MODE_PRIVATE,null);
		db.execSQL("CREATE TABLE IF NOT EXISTS TouristInfo (Id VARCHAR, fName
VARCHAR,
			 lName VARCHAR, Age VARCHAR, Job VARCHAR, Status VARCHAR);");
	}
	public void addData(String id,String  Fname, String Lname,String  age, String
job,String
				status, Context cont){
		try {
			c=db.rawQuery("select * from TouristInfo",    null);
			counter=c.getCount();
			if(counter>=1){
				T= Toast.makeText(cont, "You already created a Personal
profile No
				   need to create new record, You can update it" ,
				   Toast.LENGTH_LONG);
				T.show();
			}
			else{
			  db.execSQL("INSERT INTO TouristInfo VALUES ('" + id + "','" +
Fname + "',
				'" + Lname + "', '" + age + "', '" + job + "', '" + status +
				"');");
			  Toast.makeText(cont, "Your Profile Information Added
Successfully!" ,
				   Toast.LENGTH_SHORT).show();
			}
		}
			catch (SQLiteException se ) {
				T=Toast.makeText(cont, " Could not create or Open the
database " +
					cont.getClass().getSimpleName() ,Toast.LENGTH_LONG);
				T.show();
		} finally{
			db.close();
		}
	}
	public void deleteData(String id, Context cont){
		c=db.rawQuery("SELECT * FROM TouristInfo WHERE Id='"+id+"'", null);
	   try{
			if(c.moveToFirst())
		   {
		   // Deleting record if found
			db.execSQL("DELETE FROM TouristInfo WHERE Id='"+ id +"'");
```

```
public void deleteData(String id, Context cont){
         c=db.rawQuery("SELECT * FROM TouristInfo WHERE Id='"+id+"'", null);
          try{
                    if(c.moveToFirst())
                  {
                  // Deleting record if found
                    db.execSQL("DELETE FROM TouristInfo WHERE Id='"+ id +"'");
                    T= Toast.makeText(cont, "Your Profile Removed from Database
successfully!" ,
                         Toast.LENGTH_SHORT);
                          T.show();
                  }
                    else{
                          Toast.makeText(cont, "No record found with ID= " + id ,
                             Toast.LENGTH_SHORT).show();
                    }
          }
              catch (SQLiteException se ) {
                    T=Toast.makeText(cont, " Could not remove profile from database "
+ \
                         cont.getClass().getSimpleName() ,Toast.LENGTH_LONG);
               T.show();
              } finally{
                          db.close();
              }
       }
       public void updateRecord(String id,String  Fname, String Lname,String  age,
String
                    job,String status, Context cont){
              c=db.rawQuery("SELECT * FROM TouristInfo WHERE Id='"+id+"'", null);
              try{
                    if(c.moveToFirst())
                  {
                  // Update record if found
                  db.execSQL("UPDATE TouristInfo SET fName='" + Fname +"', lName='" +
Lname +
                      "', Age='"+ age +"', Job='"+ job +"', Status= '"+ status +"' WHERE
                      ID='"+ id +"'");
                    T= Toast.makeText(cont, "Your Profile Updated successfully!" ,
                          Toast.LENGTH_SHORT);
                          T.show();
                  }
                    else{
                          Toast.makeText(cont, "No record found with ID= " + id ,
                            Toast.LENGTH_SHORT).show();
                    }
          }
              catch (SQLiteException se ) {
                    T=Toast.makeText(cont, " Could not update profile from database "
+
                           cont.getClass().getSimpleName() ,Toast.LENGTH_LONG);
                    T.show();
              } finally{
                          db.close();
              }
```

Annex 2: QR code Scanner Module Implementation Code

```
<!-- Scanner Activity User Interface -->
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:paddingBottom="@dimen/activity_vertical_margin"
    android:paddingLeft="@dimen/activity_horizontal_margin"
    android:paddingRight="@dimen/activity_horizontal_margin"
    android:paddingTop="@dimen/activity_vertical_margin"
    tools:context="com.aau.camis.Scanner"
    android:orientation="vertical"
    android:background="#000000">
    <TextView
        android:id="@+id/txt1"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="@string/hello_world"
        android:textColor="#FFFFFF" />
    <FrameLayout
          android:layout_width="match_parent"
          android:layout_height="match_parent"
          android:layout_gravity="center_horizontal">
            <include layout="@layout/capture"/>
      </FrameLayout>
</LinearLayout>
```

```
package com.aau.camis;
import android.content.Intent;
import android.graphics.Bitmap;
import android.os.Bundle;
import android.widget.TextView;
import com.google.zxing.Result;
import com.google.zxing.client.android.CaptureActivity;
public class Scanner extends CaptureActivity
{
      TextView txt1;
    /** Called when the activity is first created. */
    @Override
    public void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_scanner);
        txt1=(TextView) findViewById(R.id.txt1);
    }
    @Override
    public void handleDecode(Result rawResult, Bitmap barcode)
    {
      DateTimeCalculator dtc=new DateTimeCalculator();
      String Res=dtc.main(rawResult.getText());
      Intent i=new Intent(Scanner.this,ScanResult.class);
      i.putExtra("value", Res);
      startActivity(i);
    }
}
```

```
package com.aau.camis;
import java.text.DateFormat;
import java.text.ParseException;
import java.text.SimpleDateFormat;
import java.util.Date;
import java.util.Scanner;
import java.util.concurrent.TimeUnit;
public class DateTimeCalculator {
	public String main(String date){
		String res="";
		Scanner sc2 = null;
		boolean b;
		String alpha=date.replaceAll("[^a-zA-Z0-9/\\s]", "").replaceAll("\\s+", "
");
		sc2 = new Scanner(alpha);
		while (sc2.hasNextLine()) {
			Scanner s2 = new Scanner(sc2.nextLine());
		 while (b = s2.hasNext()) {
			String s = s2.next();
			boolean x=parseString(s);
			res= res + " "+ s;
			if(x){
			 Date currentDate=new Date();
			 SimpleDateFormat sdf=new SimpleDateFormat("dd/MM/yyyy");
			 Date d1,d3;
			 String d2;
			 try {
				d1=sdf.parse(s);
				d2=sdf.format(currentDate);
				d3=sdf.parse(d2);
				long diff=d3.getTime()-d1.getTime();

				long diffDay=TimeUnit.MILLISECONDS.toDays(diff);
				res= res + "(" + diffDay/365 +" Years age)";

			 } catch (ParseException e) {
				e.printStackTrace();
				}
			}
		  }
	    }
			return res;
	}
	public static boolean parseString(String s){
		DateFormat df = new SimpleDateFormat("MM/dd/yyyy");
		boolean ret;
		try
		{
		   df.parse(s);
		   ret= true;
		}
		catch(ParseException e)
		{
		  //not a date
			ret= false;
		}
		return ret;
```

```
package com.aau.camis;
import android.content.Intent;
import android.os.Bundle;
import android.support.v7.app.ActionBarActivity;
import android.text.method.ScrollingMovementMethod;
import android.view.Menu;
import android.view.MenuItem;
import android.widget.TextView;
public class ScanResult extends ActionBarActivity {
private TextView txt1;
private Intent i;
	@Override
	protected void onCreate(Bundle savedInstanceState) {
		super.onCreate(savedInstanceState);
		setContentView(R.layout.activity_scan_result);
		txt1=(TextView) findViewById(R.id.txt1);
		i=getIntent();
		String Result=i.getExtras().getString("value");
		txt1.setText(Result);
		txt1.setMovementMethod(new ScrollingMovementMethod());
	}
	@Override
	public boolean onCreateOptionsMenu(Menu menu) {
		// Inflate the menu; this adds items to the action bar if it is present.
		getMenuInflater().inflate(R.menu.scan_result, menu);
		return true;
	}
	@Override
	public boolean onOptionsItemSelected(MenuItem item) {
		int id = item.getItemId();
		if (id == R.id.action_settings) {
			Intent i=new Intent(ScanResult.this,MainActivity.class);
			startActivity(i);
			//return true;
		}
		return super.onOptionsItemSelected(item);
	}
}
```

Annex 3: Code Segment for Reading OWL File and Query Result Using SPARQL

```
String gid,mid,loc;
InputStream i;
int Counter=0;
gid=etxtShow.getText().toString();
mid=etxtmid.getText().toString();
loc=etxtloc.getText().toString();
Model model2=ModelFactory.createDefaultModel();
try {
      i=getAssets().open("Tour.owl");
      model2.read(i,null);
} catch (IOException e) {
      // TODO Auto-generated catch block
      e.printStackTrace();
}
  String q1="prefix foaf1: <http://xmlns.com/foaf/0.1/>" +
      "prefix res: <http://EMIS.org/Group#> " +
      "prefix rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>" +
            "SELECT ?name ?phone ?mid " +
      "WHERE {" +
      "?y foaf1:member ?x; rdf:id \"" +gid +"\" ."+
      "?x res:LocatedIn \""+loc +"\" ." +
      "?x rdf:id ?mid ." +
      "?x foaf1:name ?name ." +
      "?x foaf1:phone ?phone ." +
      "FILTER(?mid!=\"" + mid +"\") }";
  String q2="prefix foaf1: <http://xmlns.com/foaf/0.1/>" +
      "prefix res: <http://EMIS.org/Group#> " +
      "prefix rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>" +
            "SELECT (count(*) as ?count )"+
      "WHERE {" +
      "?y foaf1:member ?x; rdf:id \"" + gid +"\" ."+
      "?x res:LocatedIn \"" + loc + "\" ." +
      "?x rdf:id ?mid ." +
      "FILTER(?mid!=\"" + mid +"\") }";
 Query query=QueryFactory.create(q1);
 QueryExecution qexec=QueryExecutionFactory.create(query, model2);
 ResultSet  results = qexec.execSelect();
 ByteArrayOutputStream go = new ByteArrayOutputStream ();
 ResultSetFormatter.out((OutputStream)go,results,query);
 String result1 = go.toString();
 etxtloc.setText(result1);
```

Annex 4: Sample Ontology from Protégé

```
<?xml version="1.0"?>
<!DOCTYPE rdf:RDF [
    <!ENTITY owl "http://www.w3.org/2002/07/owl#" >
    <!ENTITY xsd "http://www.w3.org/2001/XMLSchema#" >
    <!ENTITY rdfs "http://www.w3.org/2000/01/rdf-schema#" >
    <!ENTITY rdf "http://www.w3.org/1999/02/22-rdf-syntax-ns#" >
    <!ENTITY untitled-ontology-3
"http://www.semanticweb.org/hp/ontologies/2014/11/untitled-ontology-3#" >
]>
<rdf:RDF xmlns="http://www.w3.org/2002/07/owl#"
     xml:base="http://www.w3.org/2002/07/owl"
     xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
     xmlns:owl="http://www.w3.org/2002/07/owl#"
     xmlns:untitled-ontology-
3="http://www.semanticweb.org/hp/ontologies/2014/11/untitled-ontology-3#"
     xmlns:xsd="http://www.w3.org/2001/XMLSchema#"
     xmlns:rdfs="http://www.w3.org/2000/01/rdf-schema#">
    <Ontology rdf:about="http://www.semanticweb.org/hp/ontologies/2014/11/untitled-
ontology-3"/>
    <!--
////////////////////////////////////////////////////////////////////////////////
//
    //
    // Object Properties
    //

////////////////////////////////////////////////////////////////////////////////
//
     -->
    <!-- http://www.semanticweb.org/hp/ontologies/2014/11/untitled-ontology-
3#accomplishedBy -->
    <ObjectProperty rdf:about="&untitled-ontology-3;accomplishedBy"/>
    <!-- http://www.semanticweb.org/hp/ontologies/2014/11/untitled-ontology-
3#accomplishedIn -->
    <ObjectProperty rdf:about="&untitled-ontology-3;accomplishedIn"/>
    <!-- http://www.semanticweb.org/hp/ontologies/2014/11/untitled-ontology-
3#connectedWith -->
    <ObjectProperty rdf:about="&untitled-ontology-3;connectedWith"/>
    <!-- http://www.semanticweb.org/hp/ontologies/2014/11/untitled-ontology-
3#contains -->
    <ObjectProperty rdf:about="&untitled-ontology-3;contains">
        <rdfs:range rdf:resource="&untitled-ontology-3;iQR_code"/>
    </ObjectProperty>
    <!-- http://www.semanticweb.org/hp/ontologies/2014/11/untitled-ontology-
3#engagedIn -->
    <ObjectProperty rdf:about="&untitled-ontology-3;engagedIn">
        <rdfs:range rdf:resource="&untitled-ontology-3;Activity"/>
        <rdfs:domain rdf:resource="&untitled-ontology-3;Person"/>
    </ObjectProperty>
```

```
    <!-- http://www.semanticweb.org/hp/ontologies/2014/11/untitled-ontology-3#holdBy
-->

    <ObjectProperty rdf:about="&untitled-ontology-3;holdBy">
        <rdfs:domain rdf:resource="&untitled-ontology-3;Device"/>
        <rdfs:range rdf:resource="&untitled-ontology-3;Person"/>
    </ObjectProperty>
    <!-- http://www.semanticweb.org/hp/ontologies/2014/11/untitled-ontology-
3#locatedIn -->
    <ObjectProperty rdf:about="&untitled-ontology-3;locatedIn">
        <rdf:type rdf:resource="&owl;TransitiveProperty"/>
        <rdfs:range>
            <Restriction>
                <onProperty rdf:resource="&untitled-ontology-3;xCoordinate"/>
                <someValuesFrom rdf:resource="&xsd;double"/>
            </Restriction>
        </rdfs:range>
        <rdfs:range>
            <Restriction>
                <onProperty rdf:resource="&untitled-ontology-3;yCoordinate"/>
                <someValuesFrom rdf:resource="&xsd;double"/>
            </Restriction>
        </rdfs:range>
    </ObjectProperty>
    <!--

///////////////////////////////////////////////////////////////////////////////////
//
    //
    // Data properties
    //

///////////////////////////////////////////////////////////////////////////////////
//
     -->
<!-- http://www.semanticweb.org/hp/ontologies/2014/11/untitled-ontology-3#Country -->
    <DatatypeProperty rdf:about="&untitled-ontology-3;Country"/>
    <!-- http://www.semanticweb.org/hp/ontologies/2014/11/untitled-ontology-
3#FamilyStatus -->
    <DatatypeProperty rdf:about="&untitled-ontology-3;FamilyStatus">
        <rdfs:subPropertyOf rdf:resource="&owl;topDataProperty"/>
    </DatatypeProperty>
    <!-- http://www.semanticweb.org/hp/ontologies/2014/11/untitled-ontology-
3#FullName -->
    <DatatypeProperty rdf:about="&untitled-ontology-3;FullName">
        <rdfs:range rdf:resource="&xsd;string"/>
    </DatatypeProperty>
    <!-- http://www.semanticweb.org/hp/ontologies/2014/11/untitled-ontology-3#Id -->
    <DatatypeProperty rdf:about="&untitled-ontology-3;Id"/>
    <!-- http://www.semanticweb.org/hp/ontologies/2014/11/untitled-ontology-3#Job -->
    <DatatypeProperty rdf:about="&untitled-ontology-3;Job"/>
    <!-- http://www.semanticweb.org/hp/ontologies/2014/11/untitled-ontology-
3#LocationName -->
    <DatatypeProperty rdf:about="&untitled-ontology-3;LocationName"/>
```

```
   <!--
////////////////////////////////////////////////////////////////////////////////
//
    //
    // Classes
    //

////////////////////////////////////////////////////////////////////////////////
//
     -->
    <!-- http://www.semanticweb.org/hp/ontologies/2014/11/untitled-ontology-
3#Activity -->
    <Class rdf:about="&untitled-ontology-3;Activity"/>
    <!-- http://www.semanticweb.org/hp/ontologies/2014/11/untitled-ontology-3#Block -
->
    <Class rdf:about="&untitled-ontology-3;Block">
        <rdfs:subClassOf rdf:resource="&untitled-ontology-3;Location"/>
    </Class>
    <!-- http://www.semanticweb.org/hp/ontologies/2014/11/untitled-ontology-3#Block_1
-->
    <Class rdf:about="&untitled-ontology-3;Block_1">
        <rdfs:subClassOf rdf:resource="&untitled-ontology-3;Block"/>
    </Class>
    <!-- http://www.semanticweb.org/hp/ontologies/2014/11/untitled-ontology-3#Block_2
-->
    <Class rdf:about="&untitled-ontology-3;Block_2">
        <rdfs:subClassOf rdf:resource="&untitled-ontology-3;Block"/>
    </Class>
    <!-- http://www.semanticweb.org/hp/ontologies/2014/11/untitled-ontology-
3#Classmate -->
    <Class rdf:about="&untitled-ontology-3;Classmate">
        <rdfs:subClassOf rdf:resource="&untitled-ontology-3;Group"/>
    </Class>
    <!-- http://www.semanticweb.org/hp/ontologies/2014/11/untitled-ontology-
3#Collegue -->
    <Class rdf:about="&untitled-ontology-3;Collegue">
        <rdfs:subClassOf rdf:resource="&untitled-ontology-3;Group"/>
    </Class>
    <!-- http://www.semanticweb.org/hp/ontologies/2014/11/untitled-ontology-3#Device
-->
    <Class rdf:about="&untitled-ontology-3;Device">
        <rdfs:subClassOf rdf:resource="&untitled-ontology-3;Resource"/>
    </Class>
 <!-- http://www.semanticweb.org/hp/ontologies/2014/11/untitled-ontology-3#Family -->
    <Class rdf:about="&untitled-ontology-3;Family">
        <rdfs:subClassOf rdf:resource="&untitled-ontology-3;Group"/>
    </Class>
   <!-- http://www.semanticweb.org/hp/ontologies/2014/11/untitled-ontology-3#Gate -->
    <Class rdf:about="&untitled-ontology-3;Gate">
        <rdfs:subClassOf rdf:resource="&untitled-ontology-3;Location"/>
    </Class>
```

```
<!-- http://www.semanticweb.org/hp/ontologies/2014/11/untitled-ontology-3#Group -->
    <Class rdf:about="&untitled-ontology-3;Group"/>
    <!-- http://www.semanticweb.org/hp/ontologies/2014/11/untitled-ontology-
3#Guidance -->
    <Class rdf:about="&untitled-ontology-3;Guidance">
        <rdfs:subClassOf rdf:resource="&untitled-ontology-3;Person"/>
    </Class>
    <!-- http://www.semanticweb.org/hp/ontologies/2014/11/untitled-ontology-
3#Information -->
    <Class rdf:about="&untitled-ontology-3;Information">
        <rdfs:subClassOf rdf:resource="&untitled-ontology-3;Service"/>
    </Class>
    <!-- http://www.semanticweb.org/hp/ontologies/2014/11/untitled-ontology-
3#Location -->
    <Class rdf:about="&untitled-ontology-3;Location"/>
 <!-- http://www.semanticweb.org/hp/ontologies/2014/11/untitled-ontology-3#Person -->
<Class rdf:about="&untitled-ontology-3;Person"/>
    <!-- http://www.semanticweb.org/hp/ontologies/2014/11/untitled-ontology-
3#Recommendation -->
    <Class rdf:about="&untitled-ontology-3;Recommendation">
        <rdfs:subClassOf rdf:resource="&untitled-ontology-3;Service"/>
    </Class>
    <!-- http://www.semanticweb.org/hp/ontologies/2014/11/untitled-ontology-
3#Resource -->
    <Class rdf:about="&untitled-ontology-3;Resource"/>
   <!-- http://www.semanticweb.org/hp/ontologies/2014/11/untitled-ontology-3#Rest -->
    <Class rdf:about="&untitled-ontology-3;Rest">
        <rdfs:subClassOf rdf:resource="&untitled-ontology-3;Activity"/>
    </Class>
<!-- http://www.semanticweb.org/hp/ontologies/2014/11/untitled-ontology-3#Room_1 -->
    <Class rdf:about="&untitled-ontology-3;Room_1">
        <rdfs:subClassOf rdf:resource="&untitled-ontology-3;Block_1"/>
    </Class>
 <!-- http://www.semanticweb.org/hp/ontologies/2014/11/untitled-ontology-3#Room_2 -->
    <Class rdf:about="&untitled-ontology-3;Room_2">
        <rdfs:subClassOf rdf:resource="&untitled-ontology-3;Block_1"/>
    </Class>
   <!-- http://www.semanticweb.org/hp/ontologies/2014/11/untitled-ontology-3#Scan -->
    <Class rdf:about="&untitled-ontology-3;Scan">
        <rdfs:subClassOf rdf:resource="&untitled-ontology-3;Activity"/>
    </Class>
<!-- http://www.semanticweb.org/hp/ontologies/2014/11/untitled-ontology-3#Service -->
    <Class rdf:about="&untitled-ontology-3;Service"/>
 <!-- http://www.semanticweb.org/hp/ontologies/2014/11/untitled-ontology-3#Shelf -->
    <Class rdf:about="&untitled-ontology-3;Shelf">
        <rdfs:subClassOf rdf:resource="&untitled-ontology-3;Room_1"/>
    </Class>
<!-- http://www.semanticweb.org/hp/ontologies/2014/11/untitled-ontology-3#Shelf_1 -->
    <Class rdf:about="&untitled-ontology-3;Shelf_1">
        <rdfs:subClassOf rdf:resource="&untitled-ontology-3;Shelf"/>
    </Class>
```

```
    <!--

///////////////////////////////////////////////////////////////////////////////////
//
    //
    // Individuals
    //

///////////////////////////////////////////////////////////////////////////////////
//
     -->
 <!-- http://www.semanticweb.org/hp/ontologies/2014/11/untitled-ontology-3#Allice -->
    <NamedIndividual rdf:about="&untitled-ontology-3;Allice">
        <rdf:type rdf:resource="&untitled-ontology-3;Tourist"/>
        <untitled-ontology-3:Id rdf:datatype="&xsd;integer">001</untitled-ontology-
3:Id>
        <untitled-ontology-3:FullName>Allice</untitled-ontology-3:FullName>
        <untitled-ontology-3:Country>Ethiopia</untitled-ontology-3:Country>
        <untitled-ontology-3:FamilyStatus>Single</untitled-ontology-3:FamilyStatus>
        <untitled-ontology-3:Job>Engineer</untitled-ontology-3:Job>
        <untitled-ontology-3:scaniQRcode rdf:resource="&untitled-ontology-
3;iQR_code_H1"/>
        <untitled-ontology-3:scaniQRcode rdf:resource="&untitled-ontology-
3;iQR_code_H2"/>
    </NamedIndividual>
    <!-- http://www.semanticweb.org/hp/ontologies/2014/11/untitled-ontology-3#Bob -->
    <NamedIndividual rdf:about="&untitled-ontology-3;Bob">
        <rdf:type rdf:resource="&untitled-ontology-3;Tourist"/>
        <untitled-ontology-3:Id rdf:datatype="&xsd;integer">2</untitled-ontology-
3:Id>
        <untitled-ontology-3:Country>Italy</untitled-ontology-3:Country>
        <untitled-ontology-3:FullName>Married</untitled-ontology-3:FullName>
        <untitled-ontology-3:Job>Teacher</untitled-ontology-3:Job>
        <untitled-ontology-3:FullName>Bob</untitled-ontology-3:FullName>
        <untitled-ontology-3:scaniQRcode rdf:resource="&untitled-ontology-
3;iQR_code_H2"/>
    </NamedIndividual>
    <!-- http://www.semanticweb.org/hp/ontologies/2014/11/untitled-ontology-
3#Galaxy_Note_3 -->
    <NamedIndividual rdf:about="&untitled-ontology-3;Galaxy_Note_3">
        <rdf:type rdf:resource="&untitled-ontology-3;Tablet"/>
        <untitled-ontology-3:hasName>Tab S</untitled-ontology-3:hasName>
        <untitled-ontology-3:hasAddress>C0:01:45:4B</untitled-ontology-3:hasAddress>
        <untitled-ontology-3:hashModelNo>S4601</untitled-ontology-3:hashModelNo>
        <untitled-ontology-3:holdBy rdf:resource="&untitled-ontology-3;Bob"/>
        <untitled-ontology-3:connectedWith rdf:resource="&untitled-ontology-
3;Galaxy_S4"/>
    </NamedIndividual>
    <!-- http://www.semanticweb.org/hp/ontologies/2014/11/untitled-ontology-
3#Galaxy_S4 -->
    <NamedIndividual rdf:about="&untitled-ontology-3;Galaxy_S4">
        <rdf:type rdf:resource="&untitled-ontology-3;Smart_Phone"/>
        <untitled-ontology-3:hashModelNo>I9195</untitled-ontology-3:hashModelNo>
        <untitled-ontology-3:hasAddress>B1:08:0A:04</untitled-ontology-3:hasAddress>
        <untitled-ontology-3:hasName>Galaxy S4</untitled-ontology-3:hasName>
```

```
        <untitled-ontology-3:holdBy rdf:resource="&untitled-ontology-3;Allice"/>
        <untitled-ontology-3:connectedWith rdf:resource="&untitled-ontology-
3;Nexus_6"/>
    </NamedIndividual>
 <!-- http://www.semanticweb.org/hp/ontologies/2014/11/untitled-ontology-3#Gate_1 -->
    <NamedIndividual rdf:about="&untitled-ontology-3;Gate_1">
        <rdf:type rdf:resource="&untitled-ontology-3;Gate"/>
    </NamedIndividual>
 <!-- http://www.semanticweb.org/hp/ontologies/2014/11/untitled-ontology-3#Gate_2 -->
    <NamedIndividual rdf:about="&untitled-ontology-3;Gate_2"/>
<!-- http://www.semanticweb.org/hp/ontologies/2014/11/untitled-ontology-3#Nexus_6 -->
   <NamedIndividual rdf:about="&untitled-ontology-3;Nexus_6">
        <rdf:type rdf:resource="&untitled-ontology-3;Smart_Phone"/>
        <untitled-ontology-3:hasName>nexus 6</untitled-ontology-3:hasName>
        <untitled-ontology-3:hashModelNo>N3104</untitled-ontology-3:hashModelNo>
        <untitled-ontology-3:hasAddress>B2:00:00:A0</untitled-ontology-3:hasAddress>
        <untitled-ontology-3:holdBy rdf:resource="&untitled-ontology-3;Bob"/>
        <untitled-ontology-3:connectedWith rdf:resource="&untitled-ontology-
3;Galaxy_S4"/>
    </NamedIndividual>
 <!-- http://www.semanticweb.org/hp/ontologies/2014/11/untitled-ontology-3#Note_4 -->
    <NamedIndividual rdf:about="&untitled-ontology-3;Note_4">
        <rdf:type rdf:resource="&untitled-ontology-3;Smart_Phone"/>
    </NamedIndividual>
 <!-- http://www.semanticweb.org/hp/ontologies/2014/11/untitled-ontology-3#Room_1 -->
    <NamedIndividual rdf:about="&untitled-ontology-3;Room_1">
        <rdf:type rdf:resource="&untitled-ontology-3;Room_1"/>
    </NamedIndividual>
 <!-- http://www.semanticweb.org/hp/ontologies/2014/11/untitled-ontology-3#Room_2 -->
    <NamedIndividual rdf:about="&untitled-ontology-3;Room_2">
        <rdf:type rdf:resource="&untitled-ontology-3;Room_1"/>
    </NamedIndividual>
 <!-- http://www.semanticweb.org/hp/ontologies/2014/11/untitled-ontology-3#Scan_1 -->
    <NamedIndividual rdf:about="&untitled-ontology-3;Scan_1">
        <rdf:type rdf:resource="&untitled-ontology-3;Scan"/>
        <untitled-ontology-3:startTime rdf:datatype="&xsd;integer">1000</untitled-
ontology-3:startTime>
        <untitled-ontology-3:endTime rdf:datatype="&xsd;integer">1001</untitled-
ontology-3:endTime>
        <untitled-ontology-3:accomplishedBy rdf:resource="&untitled-ontology-
3;Allice"/>
        <untitled-ontology-3:accomplishedIn rdf:resource="&untitled-ontology-
3;iQR_code_H1"/>
    </NamedIndividual>
<!-- http://www.semanticweb.org/hp/ontologies/2014/11/untitled-ontology-3#Shelf_1 -->
    <NamedIndividual rdf:about="&untitled-ontology-3;Shelf_1">
        <rdf:type rdf:resource="&untitled-ontology-3;Shelf_1"/>
        <untitled-ontology-3:shelfNumber rdf:datatype="&xsd;integer">1</untitled-
ontology-3:shelfNumber>
        <untitled-ontology-3:locatedIn rdf:resource="&untitled-ontology-3;Room_1"/>
    </NamedIndividual>
```

```
<!-- http://www.semanticweb.org/hp/ontologies/2014/11/untitled-ontology-3#Shelf_2 -->
   <NamedIndividual rdf:about="&untitled-ontology-3;Shelf_2">
        <rdf:type rdf:resource="&untitled-ontology-3;Shelf_1"/>
        <untitled-ontology-3:shelfNumber rdf:datatype="&xsd;integer">2</untitled-
ontology-3:shelfNumber>
        <untitled-ontology-3:locatedIn rdf:resource="&untitled-ontology-3;Room_1"/>
    </NamedIndividual>
 <!-- http://www.semanticweb.org/hp/ontologies/2014/11/untitled-ontology-3#Shelf_21 -
->
   <NamedIndividual rdf:about="&untitled-ontology-3;Shelf_21">
        <rdf:type rdf:resource="&untitled-ontology-3;Shelf_21"/>
        <untitled-ontology-3:shelfNumber rdf:datatype="&xsd;integer">21</untitled-
ontology-3:shelfNumber>
        <untitled-ontology-3:locatedIn rdf:resource="&untitled-ontology-3;Room_2"/>
    </NamedIndividual>
    <!-- http://www.semanticweb.org/hp/ontologies/2014/11/untitled-ontology-
3#Sony_Xperia_Z2 -->
    <NamedIndividual rdf:about="&untitled-ontology-3;Sony_Xperia_Z2">
        <rdf:type rdf:resource="&untitled-ontology-3;Smart_Phone"/>
    </NamedIndividual>
    <!-- http://www.semanticweb.org/hp/ontologies/2014/11/untitled-ontology-
3#iQR_code_H1 -->
    <NamedIndividual rdf:about="&untitled-ontology-3;iQR_code_H1">
        <rdf:type rdf:resource="&untitled-ontology-3;iQR_code"/>
        <untitled-ontology-3:heritageType>Fossil</untitled-ontology-3:heritageType>
        <untitled-ontology-3:locatedIn rdf:resource="&untitled-ontology-3;Shelf_1"/>
    </NamedIndividual>
    <!-- http://www.semanticweb.org/hp/ontologies/2014/11/untitled-ontology-
3#iQR_code_H2 -->
    <NamedIndividual rdf:about="&untitled-ontology-3;iQR_code_H2">
        <rdf:type rdf:resource="&untitled-ontology-3;iQR_code"/>
        <untitled-ontology-3:heritageType>Artifact</untitled-ontology-3:heritageType>
        <untitled-ontology-3:locatedIn rdf:resource="&untitled-ontology-3;Shelf_1"/>
    </NamedIndividual>
    <!-- http://www.semanticweb.org/hp/ontologies/2014/11/untitled-ontology-
3#iQR_code_H3 -->
    <NamedIndividual rdf:about="&untitled-ontology-3;iQR_code_H3">
        <rdf:type rdf:resource="&untitled-ontology-3;iQR_code"/>
        <untitled-ontology-3:heritageType>Painting</untitled-ontology-3:heritageType>
        <untitled-ontology-3:locatedIn rdf:resource="&untitled-ontology-3;Shelf_2"/>
    </NamedIndividual>
    <!-- http://www.semanticweb.org/hp/ontologies/2014/11/untitled-ontology-
3#iQR_code_H4 -->
    <NamedIndividual rdf:about="&untitled-ontology-3;iQR_code_H4">
        <rdf:type rdf:resource="&untitled-ontology-3;iQR_code"/>
        <untitled-ontology-3:heritageType>Fossil</untitled-ontology-3:heritageType>
        <untitled-ontology-3:locatedIn rdf:resource="&untitled-ontology-3;Shelf_1"/>
    </NamedIndividual>
    <!-- http://www.semanticweb.org/hp/ontologies/2014/11/untitled-ontology-
3#iQR_code_H5 -->
    <NamedIndividual rdf:about="&untitled-ontology-3;iQR_code_H5">
        <rdf:type rdf:resource="&untitled-ontology-3;iQR_code"/>
        <untitled-ontology-3:heritageType>Fossil</untitled-ontology-3:heritageType>
        <untitled-ontology-3:locatedIn rdf:resource="&untitled-ontology-3;Shelf_21"/>
    </NamedIndividual>
```

```
 <!-- http://www.semanticweb.org/hp/ontologies/2014/11/untitled-ontology-3#scan_2 -->
    <NamedIndividual rdf:about="&untitled-ontology-3;scan_2">
        <rdf:type rdf:resource="&untitled-ontology-3;Scan"/>
        <untitled-ontology-3:startTime rdf:datatype="&xsd;integer">0425</untitled-
ontology-3:startTime>
        <untitled-ontology-3:endTime rdf:datatype="&xsd;integer">0426</untitled-
ontology-3:endTime>
        <untitled-ontology-3:accomplishedBy rdf:resource="&untitled-ontology-3;Bob"/>
        <untitled-ontology-3:accomplishedIn rdf:resource="&untitled-ontology-
3;iQR_code_H1"/>
    </NamedIndividual>
 <!-- http://www.semanticweb.org/hp/ontologies/2014/11/untitled-ontology-3#scan_3 -->
    <NamedIndividual rdf:about="&untitled-ontology-3;scan_3">
        <rdf:type rdf:resource="&untitled-ontology-3;Scan"/>
        <untitled-ontology-3:startTime rdf:datatype="&xsd;integer">0851</untitled-
ontology-3:startTime>
        <untitled-ontology-3:endTime rdf:datatype="&xsd;integer">0852</untitled-
ontology-3:endTime>
        <untitled-ontology-3:accomplishedBy rdf:resource="&untitled-ontology-
3;Allice"/>
        <untitled-ontology-3:accomplishedIn rdf:resource="&untitled-ontology-
3;iQR_code_H2"/>
    </NamedIndividual>
</rdf:RDF>
<!-- Generated by the OWL API (version 3.5.0) http://owlapi.sourceforge.net -->
```

Declaration

I, the undersigned, declare that this research is my original work and has not been presented for degree in any other university, and that all sources of materials used for the research have been acknowledged.

Declared by:

Name: **Hassen Ibrahim**

Signature: _____

Date: _____

Confirmed by advisor:

Name: **Dr. Dejene Ejigu**

Signature: _____

Date: _____

Place and date of submission: Addis Ababa University, February 2015.