THE DESIGN OF SECURE MOBILE DATABASES: AN EVALUATION OF ALTERNATIVE SECURE ACCESS MODELS

by
Kate Johnson

A Master's paper submitted to the faculty
of the School of Information and Library Science
of the University of North Carolina at Chapel Hill
in partial fulfillment of the requirements
for the degree of Master of Science in
Information Science.

Chapel Hill, North Carolina

August, 2002

Approved by:

_____

Advisor

Kate Johnson. The Design of Secure Mobile Databases: An Evaluation of Alternative Secure Access Models. A Master's paper for the M.S. in I.S. degree. August, 2002. 110 pages. Advisor: Stephanie W. Haas.

This research considers how mobile databases can be designed to be both secure and usable. A mobile database is one that is accessed and manipulated via mobile information devices over a wireless medium. A prototype mobile database was designed and then tested against secure access control models to determine if and how these models performed in securing a mobile database.

The methodology in this research consisted of five steps. Initially, a preliminary analysis was done to delineate the environment the prototypical mobile database would be used in. Requirements definitions were established to gain a detailed understanding of the users and function of the database system. Conceptual database design was then employed to produce a database design model. In the physical database design step, the database was denormalized in order to reflect some unique computing requirements of the mobile environment. Finally, this mobile database design was tested against three secure access control models and observations made.

Headings:

Database management

Database design

Database security

Table of Contents

1. Introduction 6

2. Literature Review 10

3. Research Methodology 16

    3.1 Preliminary Analysis 17

    3.2 Requirements Definition 42

    3.3 Conceptual Database Design 47

    3.4 Physical Database Design 50

        3.4.1 Data Allocation and Caching 51

        3.4.2 Transaction Management 57

4. Testing Against Secure Access Models 71

    4.1 Discretionary Access Control 71

    4.2 Mandatory Access Control 73

    4.3 Role-Based Access Control 76

    4.4 Analysis 78

5. Conclusion 90

6. References 92

7. Appendices 96

List of Figures

1. Literature Review Schema 11

2. Mobile System Architecture 20

3. Client-Server Structure Schema 52

List of Tables

1. Tables within Electronic Medical Record Database 22

2. Potential Threats to Electronic Medical Record System 30

3. User Authorization Rules 40

4. System Use Cases 43

5. Functional Requirements 46

6. Summary of User Roles and Views Types 49

7. Summary of Lock Scenarios 66

8. Unilateral Commit Protocol Components 68

9. Comparison of Secure Access Models Against System Potential Threats 84

10. Comparison of Secure Access Models Against System Security Requirements 86

11. Comparison of Secure Access Models Against Select Denormalization Steps 88

1. Introduction

Divergent pressures are occurring within American healthcare, and the technology it uses. Growing numbers of healthcare practitioners are using mobile computing devices (i.e., Palm Pilots, Compaq iPAQs) to access, manipulate, and store patient data. At the same time, the Health Insurance Portability and Affordability Act (HIPAA) of 1996 requires hospitals, and other healthcare providers, to implement and maintain rigorous measures to secure patient data. There is growing insistence from patients as well that their data be securely handled and individual privacy respected.

1.1 Growing Use of Mobile Computing Devices for Wireless Data Access

In a 2001 survey conducted by the trade group, Health Information and Management Systems Society (HIMSS), 50% of the survey participants stated that use of "mobile information appliances" was the top emerging information technology for the next two years. The group surveyed was 928 senior level information technology executives within healthcare and vendor organizations. In a similar vein, the use of handheld personal digital assistants by clinicians was ranked as the third emerging trend.[1]

Originally, these mobile information appliances were used as electronic day-planners, with calendars, scheduling functions, and address books installed. However, the advent of wireless technology has allowed for real-time data processing and transmission. Increasingly, the data being manipulated is contained within electronic medical records (EMRs). EMRs, as defined by the Institute of Medicine, are "an

electronic repository of information about patients that presents an appropriate view (s) of patient information to healthcare providers."[2]

This data access and management is occurring over mobile computing networks. In brief, a mobile computing network consists of a hard-wired backbone network with base stations, or access points, that are equipped with wireless interfaces and can communicate with mobile units to support data access.[3] Mobile devices vary in their capabilities. At their simplest, they are capable of only downloading data, and as such, do not affect the original database. In contrast, other mobile devices have the capacity to upload, query, and process new data in ways that significantly affect the database. Mobile devices that interact with the database will be the type modeled and studied as part of this research. These emerging data management capabilities of these devices are giving rise to a facet of mobile computing – mobile database access and interaction.[4]

1.2 Growing Insistence on Health Data Security and Privacy

In August 2002 and February 2001, the patient privacy and data security rules within HIPAA were issued, respectively. Healthcare organizations have two years to fully comply with these standards. Stringent requirements were specified within each rule, for the strict control and dissemination of healthcare data. In particular, HIPAA's security standards have:

- requirements for physical, administrative and technical security access control mechanisms to data,

- requirements for audit trails, user access controls and alarms for data security breaches and directions for the levying of civil and criminal penalties if data is negligently and/or maliciously released.[5] [6]

The two disparate trends, the growing use of wireless data access via mobile devices and growing insistence on the security of health data, present a security problem. The reasons that the combination of these two trends presents a security headache for healthcare organizations lie in the minimal security mechanisms of mobile computing devices and wireless technology. As originally designed and built, mobile devices possessed little or no security measures. Their design focus was on making the device as usable as possible, with little thought to security. Moreover, wireless technology shares this dubious record for security. Wireless Encryption Protocol (WEP) is the format for encrypting wireless transmission and it is considered to have only rudimentary security. @Stake (a digital security consulting and research firm) conducted an in-depth analysis of handheld device security using the Palm Pilot as their focal point. Their conclusion was that mobile computing devices provide sparse security options and are ill suited for holding sensitive data. "In their current state, caution should be taken when employing portable devices for security purposes."[7] In a more frivolous demonstration of this point, a local healthcare information executive entitled a recent presentation "Handhelds and Security: An Oxymoron?"[8]

These potential security holes, via both the handheld device and wireless connectivity, call for a granular and multilevel security setup. As mobile databases are

increasingly used, security arrangements on the data level itself are needed. However, the phrase "easier said than done" applies here. Two sets of problems in trying to secure mobile data access arise. The first set comes from the physical limitations of the devices, and the constraints the limitations place upon database design and management. Low computer memory, limited bandwidth, and device mobility all affect how data is accessed, manipulated, and transmitted via the devices. Many of these data design issues, including data distribution and query processing, are similar to problems encountered with distributed database design, but the mobile environment causes some additional problems. The majority of the research on mobile devices has centered around questions on how to design mobile databases for optimal use. The second problem set is related to the newness of these devices. The Palm Pilot, as the first handheld device sold, entered the market only six years ago, in 1996.[9] Mobile database access via wireless connectivity is in its infancy. Research into effective security arrangements has been undertaken, but it, like the device, is new.

To summarize, mobile database use is increasing within healthcare, at a time when pressure for data security is also increasing. In addition, the innate insecurity of the devices requires a granular and multilevel approach to security. The mobile databases themselves require security models and mechanisms, yet research into secure access models for mobile databases is newly emerging.

What then is called for is a synthesis. At the intersection of these seemingly divergent trends is a common point - how mobile databases can be designed to be both secure and usable. The questions to be asked are how can secure mobile databases be

designed so that they are both secure and usable given their special constraints? What secure data access model(s) will work in the mobile computing environment?

In this research, the following was done to answer those questions. A prototype mobile database for the healthcare setting was designed after a full evaluation of its environment, users, and functional requirements. A conceptual database was designed from these analyses and requirements. A physical database design was then derived after significant denormalization for the mobile computing environment was performed. This database design was tested against several secure access models, and conclusions were made about where the access models worked with the mobile database design and where the design and access models were in conflict.

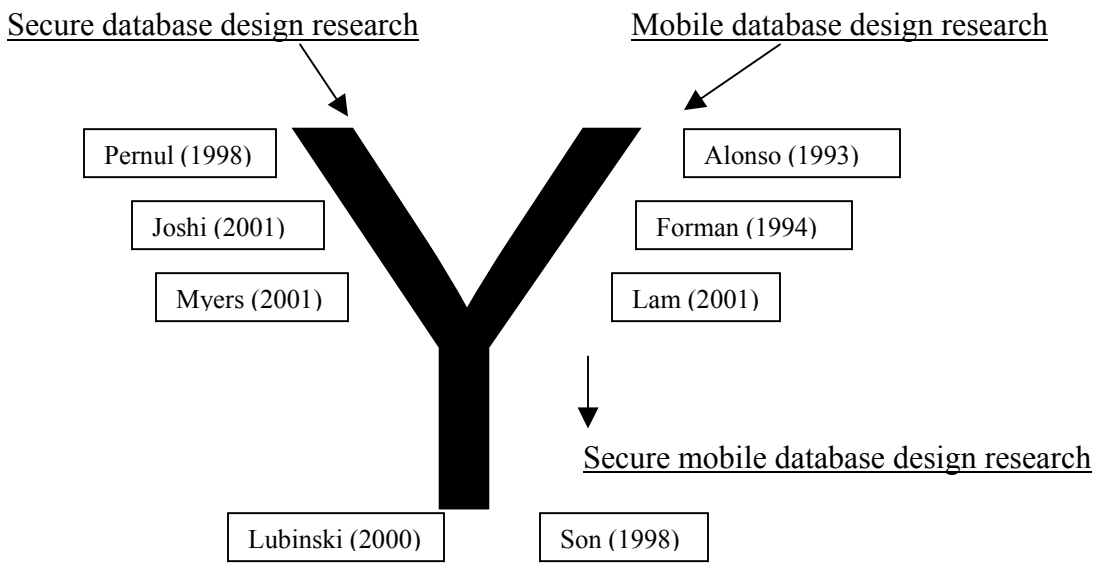
2. Literature Review

The introduction provided evidence the demand for both mobile database access and secure database access, within healthcare, are increasing together. At their intersection is the need for secure mobile databases. Yet, the research field for secure mobile database design is newly emerging, and as such, is disjointed. What is occurring is that research done in secure data access models, and separate research done in optimal mobile database design, are gradually coming together into emerging research on secure mobile database design.

Thinking of the letter 'Y' provides a useful visual (see Figure 1). On the upper left branch is the work done on secure database design. On the upper right branch is the work done on mobile database design. The lower central portion of the letter represents their

co-mingling. This research work is intended to be on the lower portion of the letter - testing what secure data access models are workable with mobile databases.

Figure 1: Literature Review Schema



This literature review represents a "knitting-together" of these, until recently, disparate disciplines.

We begin on the upper left branch with a review of database security models. Joshi (2001) provides an informative historical overview of several secure access control models for distributed data applications.[10] Before the growth in distributed and mobile databases, two database security models predominated: discretionary access control (DAC) and mandatory access control (MAC). Joshi notes the models' strengths for static database protection, but argues they lack the capability for supporting secure access in

emerging applications. He goes on to further discuss newly emerging access control schemes, such as role-based access control (RBAC) and the use of metadata intermediation, but notes these models are neither fully developed nor tested. "The DAC and MAC models lack capabilities needed to support security requirements of emerging enterprises and Web-based applications…Newer models have the potential to support emerging applications. However, these security models are yet to be fully developed and assessed."

Augmenting the traditional entity relationship (ER) database schema to explicitly address security concerns is the central focus in both the Pernul[11] (1998) and Myers[12] (2000) papers. Commonly, entity relationship diagrams are used to provide considerable detail about a database's entities, attributes, and relationships. Data access controls are rarely, if ever, acknowledged. Pernul (1998) takes the entities present in a database and subjects them to varying security classifications ranging from unsecure (U) to total security (TS). These hybridized entities (original plus security classification) are then re-modeled in the ER diagram. This application of security classifications at the entity level represents a limited mandatory access control (MAC) approach. Pernul's argument for adding security classification to the most basic level of database modeling (i.e., entity classification and modeling) is quite compelling, although limited. He does not, for example, apply security classification to data users as well.

Myers (2000) takes the approach that decentralized databases, with much of the data stored in near anonymous state, is a privacy enhancing technique. This approach reflects more the discretionary access control, or DAC, approach. In Myers' article,

information is modeled as various entities, but unlike traditional ER diagrams, each information entity is associated with an owner (termed a principal) and the label the owner assigns to the data. The label is the way principals assign a security level to the information. So, as in Pernul (1998), the traditional entity is altered into one explicitly involving a security level. In addition, the owner of the information entity can choose to release the information in an anonymous, or near anonymous, state by "declassifying" it. This declassified information entity would remain part of the information flow within the ER diagram.

We turn our attention to the right side of the Y – mobile database design. The literature presented here was intentionally selected for its focus on one aspect of mobile computing design. Forman[13] (1994) focuses on data distribution and caching, and Alonso[14] (1994) and Lam[15] (2001) focus upon transaction management. Each of these three facets is a key and unique issue in mobile database design.

As presented by Forman (1994) and Alonso (1994), three dominant issues characterize data access in mobile computing environments. They are network communication characteristics, mobility, and portability. Communication occurs over wireless networks, which are prone to disconnections, noise, and low bandwidth. Mobility causes data to change very quickly. A stationary database commonly has resource-intensive interactions with a few users whereas a mobile database will experience multiple users making fairly minimal database changes. As a result of this frequency, data within a mobile database can be quite volatile. Finally, portability places restrictions on the kind of computing devices that can be used in mobile environments.

Generally, these devices, as compared to their stationary counterparts, have significantly lower memory, processing power, and are powered by batteries.

After having described this mobile platform, Forman (1994) turns his attention to how these characteristics affect mobile databases. Forman begins by arguing that many of the data management issues seen with distributed databases can be applied to mobile databases with certain additional considerations. The issues he focuses upon are data distribution and data caching, and each issue relates to the allocation and load of data between the base database stations and mobile data units. An effective mobile database design attempts to specify an optimal load between stationary and mobile units. Caching, or temporarily storing data at the mobile station, is closely related to distribution, and also requires an optimal balance between mobile and stationary database units.

Alonso (1994) and Lam (2001) focus on transaction management in mobile database use. A transaction consists of a sequence of database operations executed as a discrete action. Each step in the sequence needs to be executed or none of them should be. If a transaction cannot complete, it is rolled-back and the database is unaffected. Typically, a database will lock those items involved in a transaction until the sequence is completed. However, as presented by Alonso (1994) and Lam (2001), mobile device issues introduce complexity to locking. If a transaction is occurring, and the device loses network connection due to power interruptions or the user moving out of wireless range, does the transaction automatically abort and roll back or do locks on data items remain until connectivity is restored? Alternatively, Lam (2001) proposes that initiation of a transaction by a mobile device cause the stationary database to replicate all involved data

items. If the device is unable to complete the transaction, the stationary database inserts the replicated data items in place of the original items and disregards any locks placed on these items.

We are now at the co-mingling of secure data access and mobile database design - secure mobile database design. Two applications of a secure access model (metadata and role-based access control) to mobile database design are presented. Lubinski[16] (2000), in her discussion of databases accessed via mobile computing devices, advocates the use of metadata as an intermediary between the mobile device and fixed database. Termed an "adaptation component", this metadata would serve to enforce security and access control to the sensitive data. Metadata is defined in four parts: the humans accessing the data and their roles, the location of the mobile computing device accessing the data, hardware and software characteristics of the computing device, and characteristics of the information being accessed.

Son[17] (1998) considers the role-based access control (RBAC) model for mobile databases. "In summary, the essence of Role-Based Access Control is that rights and permissions are assigned to roles rather than to individual users. Users acquire these rights and permissions by virtue of being assigned membership in appropriate roles." In his article, Son delineates the various strengths of the RBAC model. The first strength relates to organizational structure. As roles frequently represent organizational duties or titles, RBAC can support organization-specific security models. A second strength is RBAC's incorporation of the earlier security models of DAC and MAC. DAC and MAC

policies can be expressed by "embedding" user rights expressed under the earlier models into the role access rights specified by RBAC.

To summarize, a review of the literature reveals that the secure mobile database design is a developing research field. Moreover, its "parent" research fields (secure database design and mobile database design) continue to evolve as well. More in-depth understanding is occurring on how to craft more stringent and usable security methods while, at the same time, knowledge grows on how to design database that work more effectively in the mobile setting. Secure mobile database design is the joining of these disparate and growing fields.

This research will show this joining of disparate research also. Modeling of the prototypical database will show specific modification due to the special concerns of the mobile computing environment as noted in Alonso (1994), Forman (1994), and others. Once this mobile database is thoroughly modeled, access control models as discussed in Joshi (2001), Pernul (1998), Myers (2001) and others will be tested against this design.

3. Research Methodology

> "To keep analysis, and design, ... of distributed secure health information systems manageable, such systems as well as their underlying concepts have to be formalised and systematized using modelling techniques."[18]

System security does not occur in a vacuum. Before we understand how a mobile database can be secured, it must first be defined, designed, and then tested against selected secure access models to see what works and what does not. The approach taken

here consists of a combination of two techniques: the traditional relational database design model (RDDM) and Unified Modeling Language (UML). UML is a "modeling language, a notation used to express and document designs."[19] While the use of UML is frequently associated with object-oriented databases, this project will use the relational database model. The relational database model provides for thoroughness in database design (i.e., it begins with requirements definition to conceptual data modeling, and proceeds through to the actual physical design of the database). Moreover, the majority of database management systems continue to use the relational model. The relational database design model is not without criticisms, however, and it is to bolster that weak part UML is being used. In its requirements collection and analysis step, the RDDM is criticized as being too data-centric. User requirements, which do not use or generate data items directly, are difficult to include in a RDDM. UML, in contrast, allows for more robust modeling of user requirements through its construction of use cases.[20] A use case can be thought of as a scenario of how a user will interact with the system. An example use case in a healthcare application is when a nurse queries the database for the medication list of a specific patient. She alters the dosage on one medication and inserts the new data into the patient's record.

3.1 Preliminary Analysis

The main intent of preliminary analysis is to delineate the scope of the researched system. As noted in the introduction, much of the mobile data access and manipulation within healthcare involves use of an EMR, or electronic medical record. This section

will use an EMR as the sample healthcare application so as to adhere to this real-world fact. In this section, several facets of this system are studied, ranging from the environment in which the EMR will be used to the anticipated threats against it. To demonstrate the need for data security, the security weaknesses of networks and wireless channels will be presented. The security and performance requirements of the system will be studied. The section will conclude by describing some exclusions from this research.

3.1.1 Environmental Characteristics of the EMR and Database System

The main objective of this section is to identify the scope and features of the studied system. The delineated system has four facets: the wireless LAN and mobile network setup, the electronic medical record, the underlying EMR database structure, and the mobile device's capability and varying operational modes. Please note this system division into four facets has more to do with the need for logical exposition than with the EMR system itself. These facets are tightly interwoven; changes and limitations with one facet may have a ripple effect upon others. Good system design requires that all must be considered simultaneously.

3.1.1.1 Wireless Network Setup

Access to the EMR will occur as part of an inpatient hospital's wireless local area network (WLAN). Connectivity to this WLAN will be limited to the geographic region of the hospital itself and will range approximately a few square kilometers. The WLAN

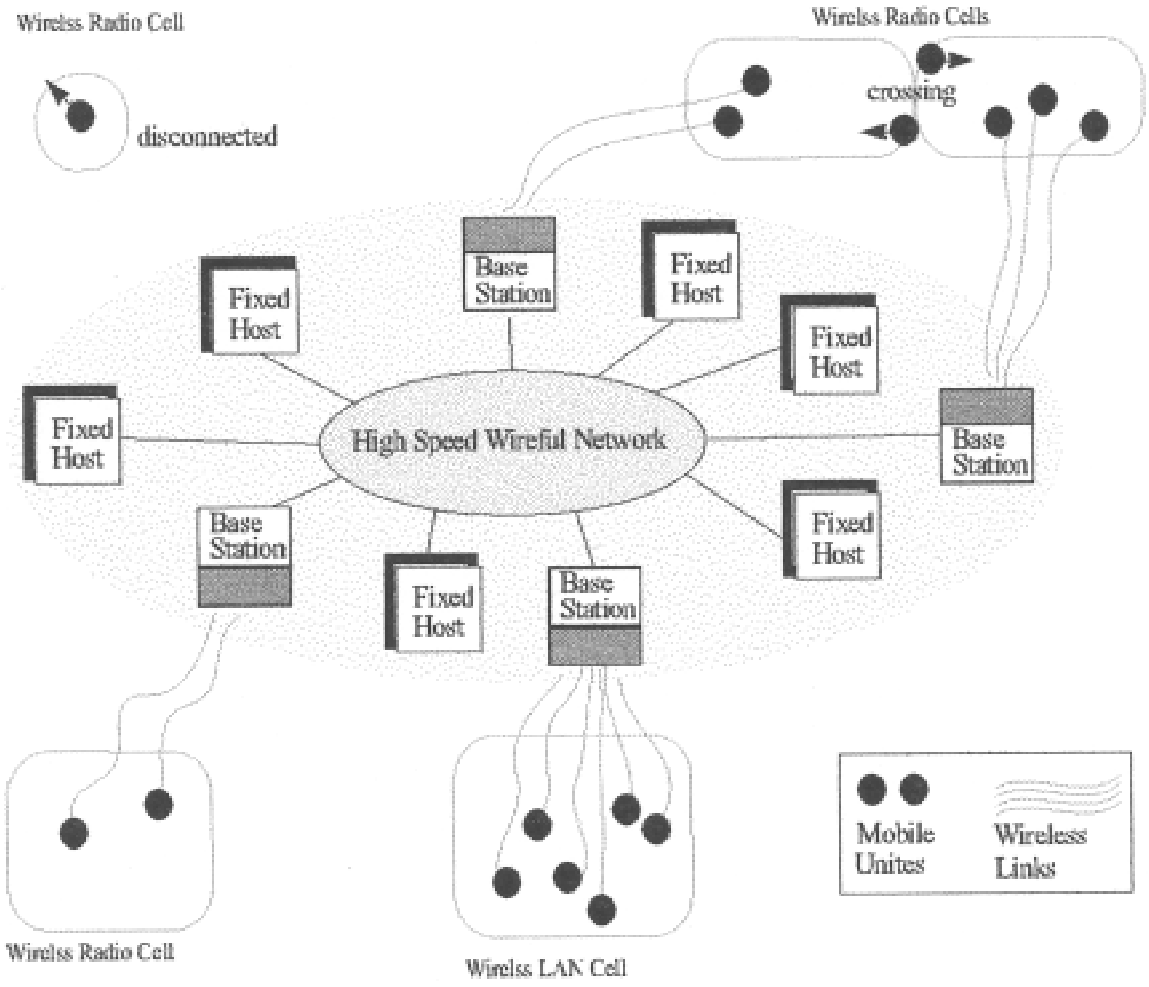
will access a nonpublic domain and predefined user accounts will be required to access this domain.

A WLAN is a data communication system implemented as an extension to the hospital's wired LAN. Using electromagnetic waves, the WLAN will transmit and receive data over the air, minimizing the need for wired connections. The architecture for a WLAN is shown in Figure 2.[21]

The WLAN architecture contains two distinct types of hosts, mobile and fixed, which are connected to a wired network. Some of the fixed hosts, called access points, base stations or mobile support stations, are augmented with a wireless interface to communicate with mobile hosts. In a WLAN configuration, an access point serves as a transmitter/receiver (transceiver) device. It connects to the wired network from a fixed location, and receives, buffers, and transmits data between the WLAN and the wired network infrastructure. A single access point can support a small group of users and can function within a range of less than one hundred to several hundred feet.

The geographical area covered by a base station is called a cell. Each mobile host can directly communicate with one base station, the one covering the geographical area in which the mobile host moves. Mobile device users access the WLAN through wireless LAN adapters, which are implemented as fully integrated devices within hand-held computers. WLAN adapters provide an interface between the network operating system and the airwaves.

Figure 2: Mobile System Architecture



The wireless medium will be 802.11, which refers to a family of specifications developed by the Institute of Electrical and Electronics Engineers (IEEE) for wireless LAN technology. 802.11 specifies an over-the-air interface between a wireless client and a base station or between two wireless clients. There are several specifications within 802.11:

- 802.11a: An extension to 802.11 that applies to wireless LANs and provides up to 54 Mbps in the 5GHz band.
- 802.11b (also known as Wi-Fi): An extension to 802.11 that applies to wireless LANS and provides up to 11 Mbps transmission in the 2.4 GHz band.

It will be assumed this LAN is using the Wi-Fi specification.

WLANs allow users to wirelessly communicate, be mobile while doing so, and use portable devices. Each of these properties, however, introduces problems.

- Disconnection: Frequent disconnections, between mobile devices and the network, can occur due to noise and interference.
- Bandwidth: Two factors affect bandwidth. The first is its limited capacity. As noted in the earlier discussion, Wi-Fi provides up to 11 Mbps - a "narrow pipe" that dictates a limited volume of information. The second factor is variability. Noise, interference, and the number of mobile users on an access point at one time can introduce variation in the amount of bandwidth available to transmit data.
- Security Risks: Wireless networks pose additional security risks to wired networks. These risks will be discussed in Section 3.1.3.

3.1.1.2 EMR Characteristics

The EMR used in this paper contains the demographic information of a patient along with information on his medications, allergies, family history and other salient

facts. An individual patient's data is contained within one row for each table. Table 1 lists the EMR's tables.

Table 1: Tables within Electronic Medical Record Database	
Patient Information (Tables)	**Primary Key**
1. Demographics (main)	PatientID
2. Lab Test	PatientID
3. Diagnostic Tests	PatientID
4. Diagnostic Test Lookup Table	Diagnostic International Classification of Disease Code v. 9 (ICD9) Code
5. Procedures	PatientID
6. Procedures Lookup Table	Procedural ICD9 Code
7. Treatment Plan	PatientID
8. Allergies	PatientID
9. Family Medical History	PatientID
10. Medications	PatientID
11. Medications Lookup Table	Medication Code
12. Physician Referrals	PatientID

This EMR is a simplified abstraction of a real EMR. A cursory review of the EMR entities shows many omissions from an actual EMR (e.g., no encounter history, progress notes, or patient problem list). The intent of this EMR example is to demonstrate the contents of a typical patient database and illustrate the challenges of designing a secure mobile database.

The EMR is based upon a relational database with patient data distributed over ten table entities. The patient demographic table is the central entity and other entities are linked via the key of a patient identification number. A full listing of the EMR's data elements and its complete entity-relationship model are available in Appendix B.

Delineating an EMR requires considering two facets – the usage of the system and the data contained within it. The usage of the system is considered first. The EMR

system supports both a main EMR, accessible by fixed hosts, and a portion accessible via the mobile platform. The mobile portion contains the most salient and time-sensitive aspects of the total EMR. Rather than serving as the hospital's main patient EMR, the EMR used via the mobile platform is more of an accessory. Because it is viewed primarily as a mobile convenience, there are some limitations upon its use. First, it is assumed that initial patient registration and data entry has already occurred (i.e., by clerks in an admissions office using desktops), and the mobile devices and wireless LAN are used for some data acquisition, viewing, manipulation, and updating. Second, an entire EMR cannot neither be created nor deleted via a mobile device, but insertion and deletion is allowed for select data elements within entities. Lastly, a mobile device can access only one EMR at a time and is subject to the concurrency control techniques used with the database.

Our focus now turns to the data contained within the system. Formatting and security concerns come into play here. Data that is stored in a database and transmitted over wireless channels is intentionally chosen to be character or numerical string fixed length field types. Data that is commonly found in a paper based medical record (i.e., X-ray images, free text fields) will not be contained in this mobile EMR. The reason for this restriction is that images are commonly very large files and would transmit slowly, or not at all, over a wireless network connection.

Though they are similar in field type, the data within the EMR vary considerably in their security requirements. It would be an unusual situation if a patient's name must be blocked from most authorized system users' access; it should be commonplace that

most system users do not access a patient's HIV status. Because of this variety, security should be granular and assigned to individual data elements. For this reason, views have been constructed throughout the EMR. These views are discussed in detail in Section 3.3.

Although a mobile EMR gives clinicians quick and portable access to needed information, problems are present. These problems are compounded because they require seemingly contradictory responses from the EMR system. Quick and consistent access and updates to the data are needed. Yet this volume of transactions could introduce "dirty data" into a system where data integrity is essential. The need for security and the limitations of the mobile setup exert divergent pressures as well. Good security requires overhead upon a system - access control and authorization must be followed, concurrency control maintained to ensure data integrity, etc. Yet, the narrow bandwidth, transaction interference because of disconnection and other features of mobility require the overhead be kept to a minimum.

3.1.1.3 Database Characteristics

In this research, the database being studied adheres to some principles of distributed database design but with key modifications for the mobile platform. Like one version of a distributed database, the system consists of a central database where data storage occurs. Multiple processing devices may access, retrieve, and manipulate the data. The database system is homogeneous across the database server and the units accessing it. Data insertion, and updating occur on a mobile device, but data is ultimately

transferred to the central server for synchronization into the main database and storage. (Data may be temporarily cached upon devices other than the server but not stored.)

Owing to the limitations of a mobile database, however, techniques different from those developed for distributed design are required as well. Compared with wired networks, mobile networks are usually much slower and more unreliable. Disconnection between a mobile device and the network can occur frequently. It is much more difficult to execute transactions in a mobile environment. The communication delay for the processes of a mobile transaction is unpredictable and can be lengthy. Another serious problem in a mobile database is the potential risk of network disconnection, which can significantly affect the management of a transaction. Not only is the processing of the disconnected transaction affected, other transactions may also be affected if they want to access the data items currently locked by the disconnected transaction.

For this research, three design decisions have been made regarding the database's setup. The first deals with the distribution of computation between the database server and mobile devices. The device possesses enough memory and computing power to perform some distributed computation locally. This approach has both advantages and disadvantages. The disadvantages for performing a portion of the computation on a mobile host are that it can tax the device's power consumption and complicate data replication and transaction management. The advantages are that it allows the autonomous operation of a mobile device during partial and total disconnections and limits the volume of data transferred over limited bandwidth. In this research, we take the approach that part of the computation will be executed in a mobile host.

The second assumption concerns data distribution. As noted above, data storage occurs ultimately on the server. However, to ensure availability of the data to mobile devices and reduce the volume of information going through the wireless channels, some distribution of the data must occur between the server, the mobile devices themselves, and other units.

The last assumption concerns transaction modeling. Within this research, transactions must achieve disparate goals. Maintaining data integrity is key so transactions have to be managed such that overwrites and ambiguities do not occur. At the same time, the limitations of the mobile setting (i.e., constrained bandwidth, intermittent disconnections) are accommodated.

The issues surrounding data allocation, and transaction modeling are extensively explored in the Physical Data Modeling section, and will not be discussed here.

3.1.1.4 Mobile Device Characteristics

The mobile devices studied for this research are battery-powered portable computers. The devices are frequently referred to as personal digital assistants (PDAs), with Palm Pilots being perhaps the best known. The devices contain wireless network interface (NIC) cards that allow the device to access the wireless LAN. The devices function within an area, known as a cell, which is restricted by the dimensions of the wireless LAN. The mobile computing platform functions as a client-server model, with one twist. Between the mobile clients and the database server are base stations, which contain transmitters and receivers for communicating with the devices. So the

communication channel technically spans three units. For purposes of this research, the communication channel will be modeled as a link between mobile client, access point (also known as a base station) and database server.

The portability of mobile devices places limitations on their capabilities. Small interface screens, limited battery power, and small memory capability characterize these devices. The limitations upon bandwidth have been noted already. In addition, unlike fixed hosts, which are either fully connected to or disconnected from the network, mobile hosts have additional operational modes. The amount of bandwidth determines if the device is fully or partially connected. Fully connected means the device is connected to the LAN with full bandwidth available. In disconnected mode, the device is completely divorced from the LAN. The device uses sleep mode to conserve battery power and energy. Within this mode, the device is moribund and action is not performed by it. The device returns to normal operation when action is performed upon it or it receives a message. A device operates in partially connected mode when bandwidth upon the LAN is limited and conversations between device and server must be terse and infrequent. If several mobile stations were using an access point at the same time, the already limited bandwidth (11 Mbps under the 802.11b specification) would be distributed over the multiple devices.

The varying operational modes pose transaction management risks. Does a device's disconnection during a transaction demand an abort of that transaction? Or, should the transaction and data locks be extended until the device returns to the fully connected mode?

3.1.2 Perceived Threats to the EMR System

Evaluating the perceived system threats is an exercise in risk analysis and management. The following steps in risk analysis were used: establishing the context, risk identification, risk analysis, ranking the risks, treating the risks, and monitoring and reviewing the risks. Establishing the context has been done in the preceding section. In this section, risk identifying, analyzing, and ranking will occur. The fifth and sixth steps, treating the risks, and monitoring and reviewing, are considered later in this paper.[22]

The potential threats to this system are of multiple types and from multiple sources. Thinking of a three-axis matrix provides a useful construct. (See Table 2). The first axis concerns the actors, or individuals posing a threat, and they are divided into those internal and external to the hospital organization. The second axis considers whether the harm is accidental or malicious. Influenced by these two axes is the third axis, or type of threat. Five types of threats were identified. They are loss of confidentiality, loss of privacy, loss of data availability, loss of data integrity, and loss of system integrity. The types will be discussed in turn.

- Loss of confidentiality: The main event when this loss occurs is unauthorized access and disclosure. Two different scenarios are possible here. The first is an unauthorized individual accessing the system itself. The second is when authorized users access information for which they have no need-to-know.
- Loss of privacy: This loss occurs when individual patient data is divulged inappropriately. An example of this would be when sensitive data, such as a patient's diagnosis, is readily accessible to all system users.

- Loss of system responsiveness: The EMR system's intent is to provide data to legitimate users. Two scenarios are possible here. The first is a complete blockage – no data is available to legitimate users. The second scenario deals with timeliness. When data transmission rates are so slow users are unable to do their work, the system has become nearly as useless as when data is entirely blocked.
- Loss of data integrity: Data is added, modified and/or deleted inappropriately. The sum effect of these actions is that data elements become corrupted. An example would be an external actor who maliciously inserts erroneous information into a patient's record.
- Loss of system integrity: The system must have constraints upon alterations in either user or object status. For example, no user other than the database manager should be able to grant privileges to other users. Similarly, a user with high security privileges should not be able to access a data object and then alter it so it is accessible to users with lower privileges. (i.e., no "write-downs")

Table 2: Potential Threats to EMR System		
	Internal Actor	**External Actor**
Accidental Intent	1. Confidentiality loss: System does not curtail user access 2. Privacy loss: System does not prevent patient data divulged 3. System responsiveness loss: Accidentally blocks other users access to data 4. Data integrity loss: Data accidentally corrupted 5. System integrity loss: System does not curtail privileges from being altered	1. CL: System does not prevent user access 2. PL: System does not prevent patient data divulged 3. SRL: System does not prevent or detect and end user access. 4. DIL: Data accidentally corrupted 5. SIL: System does not prevent privileges from being granted.
Malicious Intent	1. CL: Legitimate user accesses areas blocked to him 2. PL: Patient data divulged 3. SRL: Maliciously blocks other legitimate users from accessing data 4. DIL: Data maliciously corrupted 5. SIL: User privileges granted/revoked, data privileges altered	1. CL: Unauthorized individual accesses system 2. PL: Patient data divulged 3. SRL: Maliciously blocks legitimate users from accessing data 4. DIL: Data maliciously corrupted 5. SIL: User privileges granted/revoked, data privileges altered

The next step in this threat analysis is to rank the threats. The greater the harm a threat poses, the higher the risk it should be assigned. We begin with our actors, internal and external, and the internal actor poses the greatest risk because she already possesses some trust from the system. Indeed, in a 2000 report, Price Waterhouse Cooper estimates that internal actors commit 85% of cyber attacks.[23] Secondly, we look at intent. Though

accidental blundering can cause harm, maliciousness is more dangerous because the actor intends to cause harm, can select those system areas in which the greatest harm can be done, and can work to conceal his activities. An area where great harm can be done is in data integrity. Additions or deletions of data are likely to cause notice, but subtle alterations to data can pass unnoticed and result in wrong medications, undetected allergies and other hazards to patients. Thus, the threats posed by the malicious internal actor poses the greatest risk.

We turn to the other three areas. Even though her actions are unintended, the internal actor with accidental intent poses the second greatest threat. As noted above, the internal actor, regardless of intent, already enjoys some trust and latitude from the system. The malicious and accidental external actors pose the third and fourth greatest risks, respectively.

3.1.3. Weaknesses of Current Security Arrangements

There may be readers who are saying at this point "Focusing on database security is beside the point. There are security measures for networks, and wireless channels. Use those and your system will be secure." While it is true those security measures exist, it is also true that weaknesses have been identified in each. In this section, the weaknesses will be briefly discussed and the concomitant need for database security demonstrated.

3.1.3.1 Network Security Weaknesses

Networks are, by their definition, a collection of linked computers. This collection of computers, connected either via wired, wireless, or both connection types, communicate to one another. The hospital WLAN considered in this research will very likely communicate with other networks and computers outside it as well.

It is because of this very interdependence and communication that network weaknesses exist. Let us think of our prototypical network and consider where there might be security holes. We begin with the computers themselves. The operating systems may have bugs that inadvertently divulge data and make the system as a whole insecure. Applications running onto the operating system may open their own insecurities. And the humans using the computers are potentially the most dangerous element and may accidentally or maliciously weaken security as well. Communication between the computers may be sniffed or intercepted and sensitive data, such as usernames and passwords, revealed. The links the network uses to communicate with the outside world may be sniffed, attacked, or entered surreptitiously.

The above paragraph illustrated the numerous weak links that may exist in the chain of network security. From this point, we turn to the types of network attacks that are possible because of these weaknesses. There are four primary types of attacks: interruption, interception, modification, and fabrication.[24] The best-known example of an interruption attack is a denial-of-service (DOS) attack. During a DOS attack, the network is flooded with so much bogus traffic it either cannot respond to the legitimate traffic or it is overwhelmed and shuts down. The net effect either way is the network becomes

unusable. These attacks can be difficult to combat because attackers frequently conceal their source, or have multiple sources. Because the attack source is ambiguous and varied, the victim does not know whose incoming traffic to block.

In an interception attack, a tool called a sniffer is frequently used. A sniffer monitors network traffic and will obtain valuable data such as passwords as the data goes through the sniffer's surveillance. A defense against this type of attack is encryption of the channels through which the data passes. To work, however, the encryption must be from "end-to-end" (i.e., from the source of the sensitive data to its final destination).

Modification attacks intend to change data or programs contained within the network. Email viruses, which modify or delete data from a user's computer, are weapons in a modification attack. Defensive tactics against this attack would be email filtering and anti-virus programs but the number and variety of viruses can overwhelm these defenses. Vigilant system administrators who keep their anti-virus definitions up-to-date, and prudent users who do not open unknown email attachments, are also essential in defending against modification attacks.

Lastly, we come to fabrication attacks. In this type of attack, malformed data is crafted and send to a device on the network to be processed. The intent is the data causes the device to behave unexpectedly and open a security hole. An example of this type of attack is a buffer overflow. With a buffer overflow, the attacker attempts to write their instructions (contained within the malformed data) to some key part of the device's instructions. Once the device encounters that part of its instructions, it will execute the bogus code written there.

This is not an exhaustive problem list for network security. As noted in the section's opening paragraph, weaknesses also exist for applications, operating systems, and other network components.

3.1.3.2 Wireless Channel Security Weaknesses

The 802.11 standards contain an encryption option, which is intended to provide confidentiality. From the beginning of this standard, WEP (wireless encryption protocol) was not designed to be very robust. WEP was defined in the 802.11 standard as "protecting authorized users of a WLAN from casual eavesdropping."[25] Over the last year, however, WEP has been shown to possess many weaknesses. Additionally, tools to exploit these weaknesses are now freely available over the Internet.

For encryption, WEP uses RC4, a symmetric algorithm known as a stream cipher. A symmetric algorithm is one that relies on a single shared key (as opposed to a public key) that is used at one end to convert plaintext into ciphertext, and at the other end to convert the ciphertext back to plaintext. The sender and the receiver share the same key, and it must be kept secret. In addition, stream ciphers encrypt data as it is received, as opposed to block ciphers that collect data in a buffer and then encrypt it a block at a time. Stream ciphers operate by expanding the shared key into an infinite pseudo-random key stream, which is logically combined (or XORed) with the plaintext to produce ciphertext. Being a symmetric cipher, the user uses the shared key at the receiving end to regenerate the identical key stream, which is then XORed with the ciphertext to reproduce the plaintext.[26]

In practice, an infinitely long key stream is never produced; it is only as long as the data stream being encrypted. Once a key has been used to generate a key stream, the same key can never be reused again because it will generate the same key stream. If an attacker can obtain two different ciphertexts encrypted with the same key stream, the encryption process can be broken and the contents of the shared key determined.

There are several problems with this security setup. The IEEE standard does not specify how the secret key is established, or contain any provision for key management. Often, a single key is shared between all mobile devices and access points, and used repeatedly. The designers of WEP tried to get around this by appending a unique initialization vector (IV), a 24-bit number, to the common shared 40-bit key. The effect is that instead of having only one 40-bit shared key available for use, there are now many different 64-bit shared keys. The receiver only needs to know the secret shared 40-bit portion, which is common to all of them. The unique 24-bit IV vector determines which of the keys was used to encrypt a particular packet. The key stream is generated with this unique 64-bit key and the key and the key stream are supposed to change for every packet.

The problem is the IV is transmitted unencrypted with each packet. There are only a finite number of IVs available for use, and there is no mechanism in the standard for changing the shared key when the available unique IVs are used up. In addition, many vendor cards reset the IV to 0 each time a card is initialized, and increments the IV by 1 with each packet.[27]

3.1.4. System Performance and Security Requirements

In this section, we address what performance and security requirements are expected from the system. We identify the specific requirements, their meaning, and their specific application in a mobile database environment.

Our performance requirements fall into five broad categories.

- Reliability of operations: This is defined as system's ability to continue operations under normal conditions and despite some critical failures.
    - Specific mobile database concerns:
        - Connection across the wireless network can be weak at times and devices will be disconnected from the network.
        - Mobile devices, unlike their fixed counterparts, will disconnect from the network in order to conserve power. This system should make different assumptions about connectivity than traditional system in which disconnection is considered device failure.
- Data delivery, timeliness and response time: Users should be able to access requested records in a timely manner. Latency in the system needs to be kept to a minimum.
    - Specific mobile database concerns:
        - Because of limited and variable bandwidth, the communication across the wireless medium should be judiciously used to alleviate any congestion that impedes delivery.

        - Varying bandwidth and wireless channel noise will pose obstacles to timely data delivery. The system, primarily transaction management, must take necessary steps to ameliorate these obstacles.

- Data availability: This is defined as the system's readiness to respond to and deliver on data requests.
    - Specific mobile data concerns:
        - Varying bandwidth and wireless channel noise will pose obstacles to data availability. Via lock management and caching, the mobile database setup will attempt to ensure maximum data availability to users.
        - Because of their tendency to disconnect and leave the network, devices with locks upon data can block or impede other users' access to data. The mobile database setup will need transaction management that accommodates and manages for this effect on data availability.
- Data Integrity Maintenance and Protection: This is defined as preventing the corruption of data.
    - Specific mobile data concerns:
        - Risks to transactions because of the intermittent connectivity of devices have the potential to corrupt data within a mobile database.

        - Portable computing devices increase the risk of data loss through loss, physical damage, theft, and unauthorized access. Minimizing the data kept on the device can help reduce these risks.

- Miscellaneous: This category is a combination of two mobile data concerns.
    - Specific mobile data concerns:
        - Support for Varying Operational Modes: The system must recognize, and accommodate, the varying operational modes (connected, disconnected, partially connected) of the mobile device. These varying mobile device modes influence how a transaction must be executed and committed.
        - User Updates on System Setup: Users will be kept informed of other locks and changes in lock status. Cascading updates will occur to avoid the dirty read problem.

We turn our attention to the security requirements of the system. The conditions for this database setup present a major obstacle to security. The system contains highly sensitive information that must be accessed in a timely manner by multiple users with varying security levels. In addition, the mobile platform's constraints of limited bandwidth, small memory, etc. work against the overhead incurred with security practices. Therefore, good database security must be designed to serve several masters and fight multiple hazards. Database security must be concerned with defining and

controlling access and information flows into the database. In addition, information flows within the database must be controlled.

We now turn our attention to the security requirements of the system. Specific requirements are:

- Access control: Preventing unauthorized individuals from accessing the database or making malicious changes to the data.
- User authentication: User accounts will be identified and assigned to a specified security classification level upon login.
- User authorization: Users will be subject to specific database access (i.e., views), manipulation, and updating rules. The requirements will enable users to access selected portions of the database without gaining access to other database portions. For example, the attending physician and nurses would have full access to the EMR of a patient for whom they are caring. They will be able to read (R) and write (W) the record. A ward clerk, conversely, would only need access to a limited data set from the EMR. Table 3 contains a summary of this information.

Table 3: User Authorization Rules

EMR Granularity / Users	Entire EMR	Tables	Data Elements
Attending Physician	R, W	R, W	R, W
Nurse	R, W	R, W	R, W
Consulting Physician	R, W	R, W	R, W
Other Clinician*	R, W	R, W	R, W
Ward Clerk	No	R, W	R, W**
Laboratory Staff	No	R, W***	R, W
Pharmacy Staff	No	R, W****	R, W

* Other clinician equals physical therapists, social workers, technicians etc.
**Data elements containing medical values not viewable by this user.
***Laboratory staff may access the laboratory value and patient demographic tables only.
****Pharmacy staff may access the medications and patient demographics tables only.

- Privileges granting or revocation: No user, other than the database administrator (DBA), may grant and/or revoke database privileges to another user. Similarly, no user other than the DBA may alter the data view granted to another user.
- Data consistency and integrity: The system's concurrency locks will be structured to prevent unauthorized alternations to the data. Users would be prevented from making accidental or malicious changes to the data.
- Data availability to legitimate users: The system will be accessible and available for authorized users. Transactions, and concurrency control locks, will be processed in such a way that users have, at minimum, read access the majority of time. Users need a timely response to their data requests and updates.

- Non-repudiation and accountability: All changes to the data will be explicitly linked to a specific user. Because of this linkage, no user will be able to repudiate any changes he made to the data.

3.1.5. Exclusions

It is important to note what is, and is not, addressed in this research. Because of the constraints of space and time, this research is not an exhaustive review of the design of secure mobile databases. The following areas are not addressed within this paper:

- Data transmission via 802.11 wireless channels alone is reviewed. Data transmission via infrared or cellular networks is not evaluated. Devices equipped with the Bluetooth protocol, for example, are not addressed in this research.
- Wireless users commonly pass through multiple cells during use of their mobile device. The process during which a mobile host enters a new cell is called a hand-off, and to accommodate smooth hand-off, cells usually overlap. This hand-off process complicates database management significantly. This research will consider data transfer occurring only within cells, and not evaluate the changes introduced by hand-offs.
- Data encryption, and the security introduced by those techniques, will not be discussed.

- All denormalization steps, which could be taken for a mobile database, will not be performed. Significant denormalization can occur for optimal query processing, for example.
- All secure data access models have not been explored. Lubinski (2000), for example, proposes the use of metadata as a means of securing access.

3.2. Requirements Definition

The requirements definition section has two goals: "to establish the scope of the system to be built, and establish a detailed understanding of the desired capabilities of the system."[28] To establish this detailed understanding, we build upon information from the preliminary analysis stage. We then delineate the users of the database system, their needs, and the functions of the system in detail by drawing up a list of prospective database users, and the functional requirements for this prototypical database of patient medical records.

The way the requirements definition will be specified is via use cases. Within each use case are descriptions that describe the use case scenario and the flow of information through it. The cases contain, at a minimum, these elements:

- use case name,
- users involved in the use case (i.e., actors),
- a basic description of the use case, and
- the flow of events occurring within the use case.

More extensive use cases will contain additional items. These items may include:

- limitations (constraining factors upon the execution of the use case),
- preconditions (conditions that must be true for use case to execute),
- postconditions (state of the system after the use case executes), and
- alternative event flows.

Table 4 lists the thirteen use cases written for this application. The uses cases themselves are in Appendix A.

Table 4: System Use Cases
Provide Medical Care Overview
Access Medical Record Overview
Update Medical Record Overview
Close Medical Record Overview
Security Verification and View Construction
Access Patient Demographics Table
Update Patient Demographics Table
Access Patient Medications Table
Update Patient Medications Table
Access Patient Laboratory Table
Update Patient Laboratory Table
Access Patient Treatment Table
Update Patient Treatment Table

Examination of the use cases for information on system users revealed four themes. The first theme was that only a select group of users needed access to the mobile system. This limitation in the number of users occurred for two reasons. Because the mobile EMR represents only a portion of the total patient medical record and the majority

of the data is entered via desktop computers, only a small number of individuals need mobile accounts. Secondly, patient privacy is enhanced when all users do not have full access to medical data. Instead, users should only access data for which they have a need-to-know.

The second theme dealt with user access. This small group of mobile users fell into three distinct roles, direct clinical care provider, ancillary care provider, or support staff. Direct clinical care providers consisted of an attending physician and a nurse and these users require full access to medical data. Ancillary care providers consisted of a consulting physician, laboratory and pharmacy staffs, and ancillary staff such as physical therapists or social workers. This second group of users requires limited access to medical data pertinent to their specialties. For example, laboratory and pharmacy staff needs access to demographic information and the laboratory and medication tables, respectively. A ward clerk makes up the support staff and he needs access to patient demographic and administrative data. Data containing medical observations, such as laboratory test values, would be unavailable to this user.

The third theme concerned the contents of the database itself. This EMR, accessed via the mobile devices, does not need to be comprehensive and contain all patient data. Rather, it is a subset of the patient's data that is salient, timely, and likely to be changed. The EMR's data can also be formatted as text and not images. Text files are commonly much smaller than image files and require less bandwidth to transmit.

The final theme dealt with conflicting user demands for data availability and integrity in the mobile database environment. Maintaining data integrity within a

database means that changes must be handled carefully so users do not read inaccurate data, write over each other's edits, and incur other transaction mishaps. Yet, at the same time, the data locks and other mechanisms used to ensure integrity can result in users not being able to access data in a timely manner. The uses cases indicated these conflicting demands had to be addressed in the system design and ameliorated as much as possible.

We turn our attention now to the functional requirements, or capabilities, of the system. Examination of the use cases and performance and security requirements revealed a need for three capability categories. The first of these was for commonly found database operations and transactions (i.e., store data, select record, update record, close record). The second category was the operational capabilities of the system. The performance requirements specified in Section 3.1.4, such as reliability of operations and data delivery and timeliness, require system support. The last category deals with security. Detailed security requirements, which include non-repudiation, user authentication, and user authorization, need to be supported. These security requirements are also detailed in Section 3.1.4.

Table 5 below is a summary of the functional requirements data. Beside each parameter is the functional requirements pertaining to it. Each requirement possesses a brief description of it. Please note this table is a consolidation of the data. Full elucidation is found in the use cases in Appendix A.

Table 5: Functional Requirements	
Parameter	**Functional Requirements**
General operations	1. Provide for medical data storage and querying. 2. Perform common transactions of select, update, close record.
Reliability of operations	1. Accommodate mobile device disconnection.
Data delivery and timeliness	1. Support display of patient record once queried for with a patient id. 2. Support searching of patient record using key words or table names. 3. Support the concurrent display of multiple types of data.
Data availability	1. Support display of all or selected data within a patient record. 2. Display available results and indicated incomplete procedures as pending. 3. Provide patient-oriented (e.g. versus encounter-oriented) organization of and access to patient records. 4. Support controlled external access from mobile devices.
Data integrity and maintenance	1. Provide data management features that inspect inserted data for accuracy. 2. Support the collection and storage of patient data and orders.
Access control	1. Support access to EMR system by username and password. 2. Support access to patient records by patient id. 3. Limit login attempts to three. 4. Provide mobile device verification with IP address.
User authentication	1. Require valid and current username and password.
User authorization	1. Provide customized views, upon user login, to limit data access.
Non-repudiation and accountability	1. Provide security checks to control user access to patient information based on username and password. 2. Maintain security audit trail of all unsuccessful system logons including user ID, date and time.
Change in privilege status	1. Provide ability to prohibit unauthorized downloading of data to other devices.

3.3 Conceptual Database Design

The conceptual design phase represents a "summing-up" of the preliminary analysis requirements definition. Within this phase, we examine the information generated in the previous two sections and produce a conceptual design model. This model contains the database tables, their contents, and relationships between the tables.[29] In addition, because this is a database with multiple users and varying access levels, views will be constructed and then integrated.

We begin with the data gleaned from the preliminary analysis. Used within a hospital setting, this EMR contains a salient subset of the entire patient's record. The medical record is patient-oriented rather than encounter or department-oriented and the patient id serves as the primary unifying key throughout the database. Table 1 presents a listing of the tables within the EMR. The full entity-relationship model is contained in Appendix B.

From the contents we turn to the users and their roles within the system. For this research, a view integration approach[30] was taken to the modeling. In this approach, the various system users were grouped according to security levels and the user's need to see all, or a portion, of a patient's medical record. Views were then constructed, for each group, dependent upon their access rights. Once these various views have been constructed, they are then integrated together to create a composite model for the entire database.

Five types of views were identified, and they correspond to the roles of direct clinical care provider, ancillary care provider, and administrative staff. Table 6 contains

a summary of the view information. View type one is a complete data access view. The attending physician and nurse have full access to all patient data contained within the EMR. These users are participating in all aspects of the patient's care, and because of this participation, need to view and update all available data. View type two is for the consulting physician and other clinician (e.g., physical therapist or social worker). This group of clinicians participates in certain aspects of the patient's care pertaining to the clinician's specialty. For example, an orthopedist consulting on a case would need access to data pertaining to the patient's skeletal and muscular status, and a consulting respiratory therapist would need access to a patient's pulmonary status. View types three and four are reserved for the laboratory and pharmacy staff groups. The laboratory staff's access would be limited to the patient demographic and laboratory tables. Pharmacy staff would be similarly limited to the patient demographic table and the medications table. In these views, these two staff groups may upload, query, and update data directly pertaining to their work but not access data for which they have no need-to-know. Finally, the ward clerk requires a view, view type five, that allows him to do his administrative work. For this work, he would need access to data with which he can schedule patient appointments, check laboratory orders, verify insurance information etc. Data containing medical observations and values is not required to do this type of work and, in his view, that data is barred from his access.

Table 6: Summary of User Roles and Views Types

	Whole or Limited EMR View	View Type	View Reason
Direct Care Provider			
Attending Physician	Whole View	Type One	Requires complete data access
Nurse	Whole View	Type One	Requires complete data access
Ancillary Care Provider			
Consulting Physician	Limited View	Type Two	Requires data pertaining to specialty
Other Clinician	Limited View	Type Two	Requires data pertaining to specialty
Laboratory Staff	Limited View	Type Three	Requires data pertaining to laboratory
Pharmacy Staff	Limited View	Type Four	Requires data pertaining to pharmacy
Administrative Support			
Ward Clerk	Limited View	Type Five	Requires administrative data only

Here is an example of how a member of the laboratory staff would interact with the mobile database:

- A lab staff person has been requested by a nurse to perform a lab test and post the result to a patient's mobile EMR.
- The individual conducts the test and obtains the result.
- She obtains a mobile device and types in her username and password.
- Upon successful login to the system, she types in the patient's identification number.

- After the system verifies the id number, the lab staff person views the patient demographic table. She sees the option to select the laboratory table on the device screen as well. No other options are present.
- She verifies it is the correct patient and selects the lab table.
- She enters the laboratory test name, date, test value, and normal value range for that test into the device.
- She submits the data and ends her session.

The use cases, Access Laboratory Table and Update Laboratory Table, within Appendix A, contain more detailed information on these processes. Appendices C and D contain schemas representing the ER model views for the clerk, pharmacy staff, and laboratory staff users.

3.4 Physical Database Design

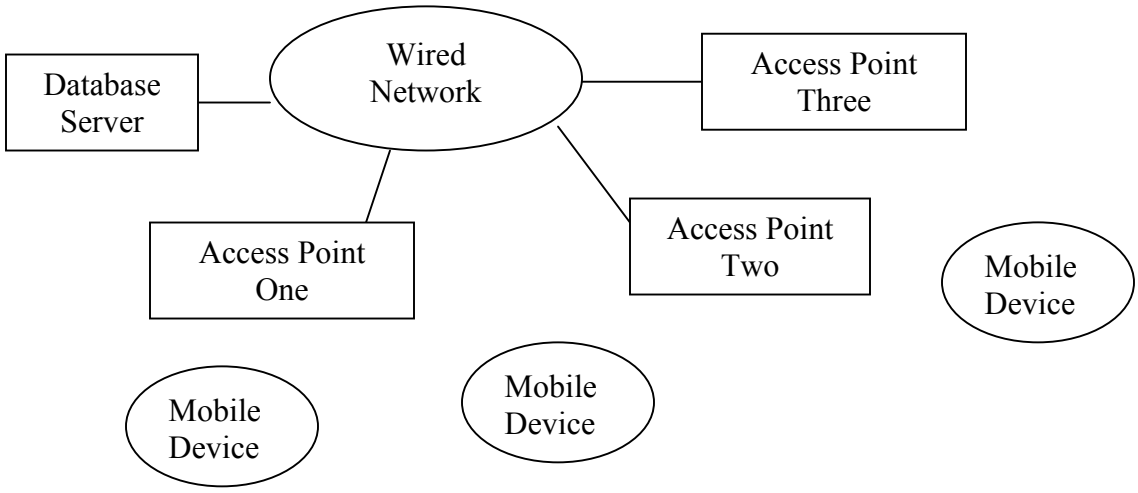
Traditionally, denormalization of the logical data model is done during physical data modeling in order to make the database more usable, perform queries faster etc. In this research, the denormalization is also done to explicitly represent the special considerations of mobile database systems. Specifically, the data model will be denormalized to reflect the issues of data allocation and caching, and transaction management.

3.4.1. Data Allocation and Caching

We begin this section by explaining the necessity for data allocation and caching in the mobile environment. Wireless networks labor under limited bandwidth and intermittent connectivity. Moreover, mobile devices will be disconnected or weakly connected to the network at times. Because of these factors, the allocation and caching of data away from the central server, becomes important to enhance data availability and data retrieval performance. "The most interesting and important difference [between mobile and traditional databases] is that mobile devices when disconnected are often operational. Disconnections from the network can be voluntary or involuntary. Consequently, information stored within the mobile device becomes crucial to maintaining productivity during a period of disconnection."[31] Therefore, with data allocation and caching, user productivity, system performance, and data availability are improved.

Determining the optimal data structure under a mobile computing environment first requires establishing the information system's structure and components. By doing so, we know our "players" (i.e., the components that can cooperate in data management and storage). Figure 3 provides a very general schema for our modified client-server structure.

Figure 3: Client-Server Structure Schema



However, this general schema is deceptive. It suggests that little change is needed from the traditional client-server structure for our setup to work well in a mobile environment. The peculiar characteristics of the mobile setting, however, require us to blur some of the distinctions found in the traditional structure. Client-server systems assume that the location of client and server hosts does not change and the connection among them is also fixed. As a result, the functionality and data management between client and server is also fixed. In a mobile environment, however, the distinction between clients and servers may have to be temporarily blurred, resulting in an extended client-server model. The memory and power limitations of mobile devices may require certain operations normally performed on clients to be performed on access points or the server. Conversely, the need to cope with uncertain connectivity requires clients to sometimes perform the functions of a server. The choice of how much management to give either client or server occurs along a continuum.

At one end of the continuum is the thin client architecture model. In this model, most functionality remains with the stationary server. In the thin client architecture, mobile devices do minimal data management. No data is allocated to them, and replication and caching are minimal. The other extreme is the full client architecture. In this model, many server functions are emulated on the client devices and, therefore, the uncertainty of connectivity and communication is minimized. Some portion of the database is allocated to the clients, data replication occurs across the wired and wireless network, and caching is extensive upon the mobile devices.[32]

For this research, a middle ground has been chosen with three components interacting. These components are the database server and access points on the wired network, and the mobile device on the wireless network. The database server retains much of the data management found in traditional client-server models. For example, no replication occurs with ultimate data storage remaining on the database server. With data allocation and caching, significant alterations have been made for the mobile setting. The access points are used as intermediaries between the server and devices, and significant data allocation and caching occurs on those points.

To allow data availability while precluding multiple copies through replication, the dedicated database server ultimately manages and stores all shared data. This central database can have fragments temporarily copied out and allocated to access points as dictated by user request and access privileges. (The logic for this arrangement is presented below.) Mobile device users, within the access point's cell, may then enact transactions upon the data at the access point. The net effect is that data is being cached

upon the access points.

The allocation of data then leads us to three additional issues: caching validation, granularity, and replacement.[33] Cache validation refers to the timeliness of the cached data. The cached file is considered valid as long as its data matches that of the server. Once updates occur to the server's data, the cached data is considered invalid and must be refreshed or flushed from the device. Three primary mechanisms exist for ensuring cache validation.

- Server messages to clients: The server sends invalidation messages to the clients or access points. An invalidation message regarding a data item that just changed is directed to clients (access points) that are caching that particular item. To do this, the server has to determine which access point has cached the data involved. The access point, in turn, relays that message to the mobile devices using the cached data. Since disconnected clients cannot be reached, each device upon reconnection has to contact the server to obtain a new version of the cache.
  - Advantages: The message is directed to specific access points and devices. Network traffic is minimized.
  - Disadvantages: The server has to be stateful and know which data is cached and where. A server is said to be stateful when it maintains a memory of the status (i.e., state) of the processes running upon it.
- Client queries: The access points periodically query the server to verify

the validity of their caches. They then relay this information to the pertinent mobile devices.

  - Advantages: The server can be stateless and not remember cache status. The information is exchanged directly between the server and the necessary access points and devices. Network traffic is minimized.
  - Disadvantages: This option has the potential to generate much traffic on both the wired and wireless networks.

- Server broadcast: The server periodically broadcasts a report in which only the database items, which have been updated, are broadcasted. But, since access points may have caches of different ages, these reports have to be within a well-defined window or marked with an update timestamp.
  - Advantages: The server is stateless since it does not know about the state of the client's caches.[34]
  - Disadvantages: This option can generate much network traffic. Potentially sensitive information can be widely broadcast across the network and increase the risk of interception.

For this research, the first option of server messaging was selected. The server is more robust than the access points and devices in terms of memory and connectivity. It can remain stateful and recall cache status. In addition, under this approach, cache updating is directed specifically to access points and devices. Network traffic is

minimized and sensitive information not widely distributed.

The granularity of data to be cached is our second issue. Data fragmentation may occur along different granular "fault lines": the entire patient record, horizontal fragmentation (rows), or vertical fragmentation (columns). The different levels of granularity each pose advantages and disadvantages. Fragmenting the entire patient record keeps the entire record intact for ease of user use and navigation. Concurrency control issues are also minimized because fragmenting the entire record prevents multiple reads and writes upon it. Allowing the entire record to be allocated, however, precludes other users from write access to that record.

Smaller fragments, as in horizontal or vertical methods, have the advantage of being smaller "data packages". Given the access point's limited memory, these smaller packages will be less taxing. Their granularity poses a disadvantage in that reincorporating them into the database will require more rigorous concurrency control.

Fragmenting the entire patient record will occur in this research, and was done for the following reasons. Keeping the entire record intact will facilitate user access to and navigation within the record. Data integrity of the record will be enhanced because multiple reads and writes cannot occur upon it. The downside that other users will be prevented from obtaining write locks is ameliorated by the fact that read locks will be still be available for data viewing.

We conclude the allocation design by considering cache replacement. When a cache is full, a cache replacement mechanism must occur to make room for incoming database items. The outgoing database items are either written back to the database if

transactions upon them have concluded or flushed from the cache. The new data item is then put in its place. Which entry is flushed is can be determined by one of two mechanisms - first in first out (FIFO) or least recently used (LRU).

The decision to cache data upon the access points was made in the following way. At certain times, parts of the database will become "hot spots" (i.e., several users want to access the same data simultaneously). It was assumed these hot spots would be concentrated around a few medical records and the users would be similarly geographically concentrated, as in a patient's room. The "hot" records could be allocated from the central database and temporarily cached to the nearest access point.

Why is the caching occurring on the access points, and not on individual devices? Two reasons are given for this decision. First, the access point, unlike the mobile devices, is neither likely to disconnect from the network nor experience bandwidth bottlenecks between it and the server. The point is a fixed node upon the wired network. In addition, multiple users, within the access point's cell, can access the data. The primary weakness of this model is that mobile devices will not have data if fully disconnected from the network. Disconnection from the network precludes contacting the access points as well as the server. However, it was decided the costs of having needed data sequestered on a device overcame the benefits of individual device caching.

3.4.2. Transaction Management

Transaction processing and concurrency control are multifaceted problems. Facets include desired transaction characteristics, management of data during transaction

processing, and commitment of data into the database at transaction end. To make this problem more manageable in this research, it was divided into the following ways. First, the unique transaction processing requirements of the mobile environment are presented. Second, the inadequacy of traditional transaction management methods for mobile environments is defined. Third, alternative methods of mobile transaction presented are considered. Fourth, the transaction method selected for the transaction execution, data commitment, and recovery phases will be presented.

Five transaction-processing requirements for the mobile environment were identified. A mobile transaction requires:

- Support for long-lived transactions,
- Support for fault tolerance because of frequent disconnections by mobile devices,
- Minimized communication (or chattiness) between server and device because of bandwidth limitations,
- Support for interruptions because of bandwidth limitations and weak network connections,
- The ability to divide computation between the mobile device and server. There are two reasons for this. The first is because mobile devices have limited memory and computing power. The second is the disconnection mobile devices experience. A device must be able to continue work even when divorced from the network.[35]

The reader may be asking at this point, "Why are traditional transaction models unusable?" Traditional transaction management is assumed to be ACID (atomic, consistent, isolated, and durable). A transaction is atomic if all or none of its operations are executed, consistent when its execution maintains database consistency, isolated when it does not generate or observe partial results for other transactions, and durable if its results are permanently committed to the database. Atomicity presents a problem because disconnections intermittently interrupt mobile transactions. Consistency is problematic because mobile transactions before execution may have to refresh data that is out-of-date due to local caching. Isolation is difficult because ensuring data availability while trying to limit the traffic between server and mobile device can result in multiple users looking at the same data. Because mobile transactions can be error-prone and long-lived, ensuring durable transactions has its difficulties as well. Moreover, the traditional two-phase protocols, used for locking and data commitment, can result in a high volume of communication over a narrow wireless channel. Pituora (1994) summarizes the traditional ACID inadequacy succinctly:

> "Mobile transactions are long-running, error-prone and heterogeneous. As a consequence, modeling mobile transactions as ACID transactions is very restrictive. ACID transactions have limited expressive power and offer no way of modeling computations with a complex control structure. Furthermore, ACID transactions do not support partial commitment or abortion of a transaction, or partial recovery. Finally, there is no way of "suspending" a transaction to survive a disconnection."[36]

If traditional models do not work, our next step then must be the selection of an alternative transaction model that does work well in mobile environments. Three

alternatives were considered and are presented below. Two of these models, the pessimistic and optimistic, are also used in traditional systems, but, with modification, function in mobile settings. This is not an exhaustive list of alternative transaction models; Seydim (1999) provides an extensive model overview.[37]

- Pessimistic Model: This method, as its name indicates, takes a cautious approach to transaction management. Before a database operation can be executed, checking is done to ensure data has not been interfered with or corrupted. Binary locking and timestamping[38] are two alternative means the model uses to ensure concurrency. Using the timestamping alternative, here is an example of how this method works:
  - In the read phase, a transaction can read data from the database. No updates are possible on the data however. A second transaction may also read the same data.
  - The second transaction requests a write lock upon the data item.
  - The database grants the write lock.
  - If the first transaction then requests a write lock, two actions are possible. Under the timestamping Wait-Die version, the first transaction must wait until the second transaction's write lock is lifted. Under the Wound-Wait version, the first transaction's request for a write lock aborts the second transaction's lock. The first transaction proceeds with its write lock and the second transaction starts again at a later time.

- Advantages:

  1. Lock or timestamping application provides defense against transaction interference.

  2. Because the device has a hold upon certain data items, a disconnection does not automatically abort the transaction. A device could obtain a write lock, disconnect and then reconnect quickly, and continue the transaction.

  3. Conflicts between different modifications to the same data items are eliminated. A user must hold the exclusive write lock before modification.

- Disadvantages:

  1. Data availability is less than in other concurrency models. For example, a write lock upon a data item prevents other users from viewing it.

  2. Communication between server and device is greater than with other models. The requesting and granting of locks or timestamps generates traffic.

- Optimistic Model: Unlike the pessimistic model, the optimistic model does not require checking while the transaction is executing. Also called the validation or certification model, the optimistic model occurs in four phases:

- In the read phase, a transaction can read data from the database. No updates are possible on the data however. A second transaction may read the same data simultaneously.
- In the local update phase, a user wishing to update data first obtains a local copy of the data items to be used in the transaction. All updates are first performed upon the local copy of the data before being written to the database. Note the second transaction is unaffected by this step.
- At the end of the transaction's execution in step 2, the validation phase checks to see whether any of the transaction's updates violate serializability (i.e., interfere with other transactions). The second transaction may have, or not, updated its own local copies.
- With the write phase, if the transactions do not interfere and serializability is not violated, the transaction updates are written to the database. If serializability is violated, the transactions are aborted and the updates discarded.

- Advantages:

  1. Transactions occur with a minimum of overhead because no checks are done until the validation phase.

  2. This model is better suited for short transactions than long ones because shorter transactions will tend to generate fewer conflicts. Most transactions in this research are assumed to be short.

- Disadvantages:

  1. Conflicts between different modifications to the same data items are possible. A user does not require an exclusive write lock before modification.

  2. The optimistic model assumes there is little interference among transactions, and most users are not requiring access to the same data items. With little interference between them, most transactions can be successfully validated. However, in a situation where much transaction interference is occurring, many transactions will be aborted. This can result in decreased user productivity and increased frustration.

  3. If the mobile device disconnects prior to the validation phase, the transaction is rolled-back. Recall that under this model the device does not have any hold upon the data items.

- Clustering Model: Unlike the previous two models, whose differences center on transaction verification, the clustering model introduces two additional techniques. In this technique, mobile transactions are broken into finer categories or subtransactions. The transactions are classified into strong read/write and weak read/write transactions. In the second technique, the data within the system is divided into clusters.

  A weak transaction may commit even if it observes inconsistent data

values, provided that the degree of inconsistency is within acceptable limits the user or system has predefined. Dividing a mobile transaction into sub-transactions such that data consistency and atomicity requirements are applied to each subtransaction, instead of the entire transaction, is the second feature of this model.

- Advantages:

  1. This method relaxes the consistency requirements for transactions and breaks down a mobile transaction into smaller sub-transactions. This sub-division makes transactions shorter and helps resolve the device disconnection and bandwidth limitation problems.

- Disadvantages:

  1. Deciding whether a transaction is weak or strong adds complexity for the system or user.
  2. For a medical database, as in this research, the needs for data integrity are such that few or no transactions may be weak.

Now that we have reviewed three transaction management options, we must decide how transaction management will be performed in this research. Making this decision involved weighing trade-offs. Under the pessimistic model, protection of data integrity is paramount. On the other hand, the optimistic and clustering methods were more congruent with the limitations of the mobile environment. Moreover, data

availability was greater under these two as well. After reviewing the alternatives, the pessimistic model with the timestamp wait-die method of concurrency control was chosen. The needs for data integrity in this research are such that the overhead generated by the pessimistic model is warranted. Steps were taken, however, to ameliorate this overhead as much as possible.

An EMR can be locked in one of four modes: read, write, intent to read, and intent to write. With the read lock, the user may read information contained within the EMR, but information cannot be inserted, changed or deleted. With read locks other transactions may read the locked data but not update it. This lock is non-exclusive. With the write lock, the user may insert, change, or delete information contained within the EMR. Entire EMRs or entities may not be deleted, however. With write locks, no other transaction may access the locked data until the transaction commits or is invalidated.

The two intent locks are intended to lower the amount of lock information transmitted between the database server and the mobile device while allowing the server to remain aware of user's lock intentions. Here is a common scenario for a lock request: A user requests a write lock upon an EMR. The database observes a write lock already exists upon the EMR. Rather than denying the lock requestor, the database instead automatically converts the write lock request to an intent to write request. Once the original write lock is removed, the IW lock reconverts to a write lock and the user is granted write access to the EMR. Table 7 is a summary of the multiple scenarios that can occur when a lock is requested.

Table 7: Summary of Lock Scenarios

Lock Held		
Lock Request	**R**	**W**
IR	Yes	No
IW	Yes	No
R	Yes	No
W	Yes	No

Let us now combine the timestamping method with locking to see how the system would perform in its entirety. Recall we are trying to fulfill disparate requirements of maintained data integrity, data availability, and mobile limitation accommodation.

Scenario One:
1. Transaction One (T1) starts at timestamp one (TS1)
2. T1 requests and receives a read (R) lock
3. T2 starts at TS2
4. TS2 requests a R lock. An intent-to-read (IR) lock is granted
5. T1's R lock is released or invalidated if TS1 grows too large
6. T2's IR lock becomes a R lock.

Scenario Two:
1. T1 starts at TS1
2. T1 requests and receives a W lock
3. T2 starts at TS2 and requests a W lock
4. T2 receives an IW lock
5. T1's W lock is released or invalidated if TS1 grows too large
6. T2's IW lock becomes a W lock.

Scenario Three:
1. T1 starts at TS1
2. T1 requests and receives a R lock
3. T2 starts at TS2 and requests and receives a W lock
4. T1's R lock is maintained with user notification of W lock

5. T2's W lock is released or invalidated if TS2 grows too large
6. T1 receives a cascading update.

Three additional system features require clarification at this point. Contending with failed transactions should require some pre-emptive action. The device should be able to declare to the network - "I am disconnecting." With this declaration, the server can nullify the locks held by the device. The server does not need to spend unnecessary time and processing power wondering if and why the disconnection has occurred.

Cascading updates are intended to resolve the dirty read problem. A dirty read occurs when a user unknowingly views data that is out-of-date and inaccurate. A cascading update does increase the communication volume between device and server but minimizes data integrity issues.

Lastly, mobile devices may disconnect during transaction and potentially leave the system in limbo. With the information gleaned from the timestamp, transactions will be aborted and locks invalidated for devices whose timestamps grow excessively long. This feature should be editable by the system administrator; during busy database periods, an allowable timestamp period will be shorter and stricter. During less intensive use of the database, more generous criteria will be used to judge timestamp periods.

3.4.2.1 Commit and Recovery Protocol

As with transaction locking, the traditional protocols for commit and recovery do not function well in mobile environments. The traditional protocol, two-phase commit or 2PC, suffers from the following weaknesses when used in a mobile setting:

- Large communication overhead: 2PC requires two rounds of messages between device and server. This volume of communication can tax an already constrained wireless channel.
- Lengthy device connection required: 2PC requires prolonged device connection for the commit to be successful.
- Misinterpreted device disconnection: A device disconnecting during a transaction can lead to an aborted transaction under traditional settings.

To address these weaknesses, this research will use a modified unilateral commit and protocol (UCM), developed by Bobineau et al. (2000), as its commit and recovery protocol. Five components cooperate within UCM and are presented in Table 8.

Table 8: Unilateral Commit Protocol Components

Component	Purpose	Location
1. Database server	Final data repository	Wired network
2. Log Agent	Logs each transaction before execution	Wired network
3. Coordinator	Directs termination protocol	Wired network
4. Mobile device users	Request transactions	Mobile
5. Device Agents	Represents users in termination and recovery protocols	Access points (aka mobile support stations)

A sample transaction will proceed in this manner:

- User requests a write lock upon some data items to begin a transaction.
- This action is logged by the Log Agent and relayed to the server.

- The lock is granted and the device can access and update the data. This action is logged by Log Agent.
- The user completes his work and requests permission to commit his updates. The Log Agent logs this and the request is relayed to the server.
- The server receives the commit request and observes the transaction's atomicity, consistency, and isolation properties are guaranteed. The durability property cannot be guaranteed yet because the entire transaction has not yet been written to logs. The server agrees to the commit request.
- The Coordinator receives this decision and writes it, and the previous actions, to the server's memory. The Coordinator then sends the server's decision to the Device Agent.
- The Device Agent instructs the device to send the updates and sends an acknowledgement when it receives the data.
- The Device Agent conveys the data to the Coordinator. It finishes writing the log and then conveys the data onto the server.
- The transaction ends.

How a recovery would proceed is dependent on where in the transaction a user disconnected. If the user disconnects during steps one through three, the Coordinator broadcasts an abort, the locks are released, and the transaction is rolled back. If the device disconnects after step seven, the transaction is unaffected and commits. For steps four through seven, a device may disconnect for a time and the Log Agent, Coordinator,

and Device Agent will maintain the transaction information. If the mobile device reconnects within an acceptable period of time, the transaction picks back up and continues. If the mobile device does not connect within this period, the locks and released and the transaction rolled back.

This transaction scenario yields several salient points on how this UCM protocol works well in the mobile setting.

- Full log files are written and maintained on the wired network, rather than the mobile devices.[39] Mobile devices can be subject to catastrophic failure in the event of the user dropping the device or theft of the device and logs can be lost or corrupted.
- The full log files allow the transaction to be kept in hiatus briefly and then resumed if the mobile device disconnects and reconnects.
- The Device Agent's assistive role allows the mobile device to disconnect earlier than if it was working alone. Note the device could disconnect after step seven above and not affect the commit.
- Communication across the wireless channel occurs only over the short distance between access point and device and is limited in amount. Much of the communication goes across the more abundant bandwidth of the wired network.
- Maintaining data integrity after recovery from aborted or failed transactions is not dependent upon the mobile device. Information maintained by the Log Agent will allow for recovery to occur.

4. Testing Against Secure Access Models

In this section, the mobile data model will be tested against three secure data access models: discretionary access control (DAC), mandatory access control (MAC), and role-based access control (RBAC).

Initially, the three models will be presented and described. Strengths and weaknesses of each model will be analyzed. Then they will be tested against the mobile database model constructed earlier in this research. Coverage and gaps between the data and security models will then be discussed.

4.1 Discretionary Access Control (DAC)

The discretionary access control approach hinges upon the granting and revoking of privileges. These privileges are identified with a user or account, and can be of two types. The first type, account level, allows the user system privileges such as Create/Delete Table, or Alter Table (ability to add/delete table columns). The second type, or table level, is more granular. It allows the user to access, or not, specific data within the database. Generally, users access is based upon tables, but access can be more finely limited to columns or rows within a table. In addition, users may have specific SQL privileges such as Select, Update, or Delete within the tables they can access. A key element within the data level privileges is table ownership. Each table has a user owner, and that user may grant and revoke privileges to other users for the data items he owns.

The Views mechanism is an important one within DAC. A view is a virtual table, which constrains a user's access to data. Necessary information can be presented while

details in underlying tables are hidden. Views have several advantages in that they require little storage room while constraining user access but can provide a customized data presentation. However, they can degrade system performance because the view must generated "on-the-fly" once the user logs in.

- Strengths:

  1. System performance is enhanced because user authentication and authorization occur at login only.

  2. It is a flexible, simple-to-implement method. Users are assigned privileges, and a database view, as they are assigned to the system.

  3. The views mechanism within DAC enforces user access rights.

- Weaknesses:

  1. Data and user authorization rules are stored, and maintained, within the system tables of the database system. This metadata would have to be queried upon user action to validate it. However, unlike MAC which requires rules to be checked for both data item and user classification, DAC requires only the user's classification be checked.

  2. The access control has little granularity with access predicated only on the user level and not the data.

  3. The model does little to guard against malicious internal actors.

  4. The model does not guard against storage channel hacks. A storage channel hack works in the following way. User_One, who is not allowed to see all patient records, types in a patient's name that he thinks is in the hospital (a celebrity, for

example). The system responds, "File in use" and User_One can infer the individual is under care.

5. DAC is vulnerable to the Trojan horse security hole. A Trojan horse is "an apparently useful and innocent program containing additional hidden code which allows the unauthorized collection, exploitation, falsification, or destruction of data."[40] Suppose User_One wants access to a table with sensitive data that User_Two can access. By writing a program with a Trojan horse, she could achieve that access:

- User_One creates a table Dummy_Table and gives write privileges to User_Two (who is unaware of User_One's true intentions)
- User_One gives herself read privileges to Dummy_Table.
- User_One copies HighlySensitive_Table to Dummy_Table.
- User_One now has access to the contents of HighlySensitive_Table.
- User_One then gets User_Two to run the program. User_One will get access to the contents of HighlySenstive_Table.[41]

4.2 Mandatory Access Control (MAC)

Also known as the lattice based access control model, the mandatory access control approach assigns security classifications to both users and data. Typical security classes are top secret (TS), secret (S), confidential (C), and unclassified (U) with TS being the highest security. Two types of restrictions are enforced for data access based on the user and data classifications.

- The simple security property: A user is not allowed read access to a data item unless the security classification of the user is greater than or equal to the security classification of the data item. The logic for this restriction is straightforward.
- The star property: A user is not allowed to write to a data item unless the security classification of the user is less than or equal to the security classification of the data item. The logic for this restriction is less intuitive. The restriction prevents a user with a high security classification from accessing a data item, making a copy, and then assigning a lower security classification to the copy. "A user with TS clearance may make a copy of an object with classification TS and then write it back as a new object with classification U, thus making it visible throughout the system."[42] This is called a "write-down" and violates a tenet of security that information should not be allowed to flow from higher to lower classification levels.

Classifications on data items can be placed upon rows or individual data items. Here is an example of classified and secret classification upon the data item level:

1. Table without Security Classifications: Patient (PatientID, Fname, Lname, BirthDate)
2. Table with Security Classifications: Patient (PatientID, $Classified_{PatientID}$, Fname, $Classified_{Fname}$, Lname, $Classified_{Lname}$, BirthDate, $Secret_{BirthDate}$)

These constraints are mandatory and automatic. The system must review these constraints each time it encounters a request for a read or write.

Polyinstantiation is a key element under MAC. A row will have different attribute values for users at different classification levels. Let us assume we have a hospital ward clerk with a classified security level, an attending physician with a top-secret classification level, and a nurse with a secret classification level. Our entire data set, with its classification levels, is below.

PatientID	LastName	DOB	PrimaryDiagnosis
111 S	Singh C	12/12/50 TS	AIDS TS
222 TS	Johansen TS	08/05/72 TS	Cystic Fibrosis TS
333 S	Kelley C	02/11/35 S	Heart Failure S

The clerk will see the last names in rows one and three, and null values in all other fields. The attending physician will see all data, and the nurse will see the following:

PatientID	LastName	DOB	PrimaryDiagnosis
111 S	Singh C	Null	Null
333 S	Kelley C	02/11/35 S	Heart Failure S

- Strengths:

  1. Security is very granular with application at user and data levels.

  2. DAC's vulnerabilities to Trojan horses and storage channel hacks are not possible under MAC.

  3. Polyinstantiation provides a mechanism for enforcing varying user access rights to data.

- Weaknesses:

  1. The overhead of checking both user and data privileges can degrade system performance.

2. The model does not guard against covert channel disclosures. A covert channel disclosure can occur in this way: A poorly designed system allows some users to see all reasons for admission in the EMR database. Other users are prevented from seeing the data if the reason for admission is for psychiatric disease or sexual assault. Each time a patient is admitted and the second group of users see a NULL admission reason, they can infer the reason was either assault or psychiatric.

3. This model allows "blind writes." A user can "write up" and not be able to read what she written. Data access rules are unaffected but data integrity, however, can be compromised because inaccurate data can be inserted during the write.[43]

4. The model can be inflexible and difficult to implement. It requires significant administrative overhead.

4.3 Role Based Access Control (RBAC)

RBAC is more recent than DAC and MAC and is an evolution from those older policies. The main concept under RBAC is that privileges are encapsulated into roles. Users are then assigned to roles, and acquire those privileges.[44] A role is defined as "an explicit (i.e., named) representation of a collection of privileges which are defined and used by system administrators and users."[45] With RBAC, database administrators may create roles, assign privileges to those roles, and then assign users to roles based on their specific job responsibilities and roles.

MAC and RBAC models have been used in conjunction with one another. The data items are assigned to classification levels (as in MAC) and the user privileges are concatenated into roles (as in RBAC).

- Strengths:

  1. RBAC can simplify user account and security management. Users are grouped into roles and have aggregate privileges assigned.

  2. The role can be activated upon user login, and data view enforced. Further system calls are not required.

  3. Because roles represent organizational structure and functions, RBAC can support organization-specific security policies.

  4. RBAC can be used along with DAC or MAC models.

- Weaknesses:

  1. Roles may have to be defined very granularly. For example, a nurse should be able to modify only the records of patients with whom he has worked. However, if one role of nurse is defined and all nurses assigned to it, all of those individuals could change any record.

  2. RBAC, by itself, provides little protection against internal actor attacks. However, this threat is removed if RBAC is used in conjunction with MAC models.

4.4 Analysis of Comparisons Between Access Models and Mobile Database Characteristics

In this section, we compare the three secure access models against selected characteristics of the mobile database design. These design characteristics chosen reflect key facets of the mobile database design. The first facet is the design's security requirements, as presented in Section 3.1.4. If the design requires certain security features, it is logical to test the models against the features and see how secure access models do, and do not, fulfill these security requirements. The second facet concerns the potential threats to the EMR system, as presented in Section 3.1.2. A secure access model, as its very name indicates, is intended to provide secure access and minimize threats to the system. We will test these potential threats against the access models to if, and how, the threats are minimized. Lastly, we consider some of the denormalization measures taken to make the database design function more efficiently in the mobile computing environment. We determine if those denormalization steps work in collaboration or in conflict with the secure access models. [46]

Our testing begins with the comparison of the secure access models against the potential system threats. Table 9 contains a summary of this testing information with each table square noting whether the access model has positive (Pos.), negative (Neg.), or Mixed (Mix.) interaction with the potential threat. Two of the threats, loss of confidentiality, and privacy have a very mixed interaction with the three security models. The DAC model provides some protection against confidentiality loss through its user classification and views mechanism. However, its lack of classification for data items

provides little protection against unauthorized access. Under MAC, data items are classified which aids confidentiality and blocks Trojan horse attacks but covert channel attacks are still possible. RBAC provides little to no protection and should be avoided for protecting against this specific threat. Combining models, unfortunately, does not bolster security more than the individual models.

The loss of privacy has a similar mixed interaction with the secure access models as the loss of confidentiality threat. Given that privacy and confidentiality are closely related to one another, it is not too surprising the two threats interactions with the secure access models are similar. Neither individual models nor model combinations provides very stringent security against the privacy threat. MAC provides the most stringent security but covert channels attacks are still possible.

The loss of system responsiveness, system integrity and data integrity threats fares better in this testing round against the access models. Because of their simplicity and little overhead, DAC and RBAC do little to impede the system and degrade its responsiveness. The tradeoff is the simplicity of the models comes at a cost of providing less than robust security. MAC does provide stringent security but will degrade system performance. Combining models reflects that tradeoff as well. A RBAC or DAC and MAC combination would enhance security but still generate overhead to the system. Combining RBAC and DAC would provide no enhanced security to the system.

Combining the MAC model with DAC provides strong protection against the threats of data or system integrity loss. The combination of the two models makes the protection sufficiently granular for both user and data item classification. Moreover, the

blind write problem under MAC is minimized. User's write privileges are dictated by their user account, and an individual would not be able to blind write dirty data into a security classification higher than his own.

The major observation from this testing of potential threats against the secure access models is that some combination of models generally provides better protection than a model working alone. With each of the five losses, some combination of DAC, MAC, or RBAC could provide protection, control access, and provide sufficient security granularity. The primary downside is that robust security has the potential to downgrade system performance as seen with the loss of system responsiveness threat.

At this point, we consider the secure access models against system security requirements. Table 10 provides a summary of this information with positive, negative, and mixed interactions noted. As with potential security threats, the access models have a varied interaction with the security requirements. DAC's emphasis on user privileges works well with the access control and user authentication requirements. The access model's simplicity of use, and minimal overhead, works well with the data availability requirement because data delivery is not impeded or system performance degraded.

The limitations of DAC become more apparent as the other security requirements are considered. User authorization and non-repudiation require tracking data items in addition to user privileges. Because DAC does not impose any security upon these items, it cannot determine or track if users are accessing unauthorized data items. In preventing privilege level changes, DAC's views mechanism enforces access privilege rules.

However, privilege rules may be circumvented by the Trojan horse problem to which DAC is vulnerable.

The MAC model does much better than DAC in satisfying these security requirements. Its stringent security on both users and data items has a positive interaction with all but two of the security requirements. With data consistency and integrity, the blind write problem under MAC can be exploited to allow for surreptitious dirty data inserts into the EMR. The MAC model "falls down on the job" in attempting to satisfy the data availability requirement. As with the system responsiveness in the potential threat section, the overhead and inflexibility of this security model can lessen data availability.

"Indifferent" would best describe the record of RBAC in this testing round. It has little overt negative interaction to the various security requirements but also has little overt positive interaction. Its lack of security granularity precludes it from satisfying any of the security requirements. Overall, its simplicity and flexibility of use alone provide a positive interaction with data availability.

The major observation from this second testing series is similar to the observation made with the potential threats. Some combination of models generally provides better protection than a model working alone. With five of the seven security requirements, some combination of DAC, MAC, or RBAC could control and track user action on the database and provide views or polyinstantiation. Unlike the potential threats example, however, combining security models did not enhance the security of two requirements.

Neither data integrity nor non-repudiation could enjoy more stringent security from a combination than under MAC alone.

We then consider the secure access models against select denormalization steps. As discussed and demonstrated throughout this research, designing databases for mobile computing environments requires significant alterations for those databases to work effectively. However, care must be taken these denormalization steps do not promote effectiveness and usability at the cost of security. Five denormalization steps were selected from this research to see if their effects upon the database complement, or hinder, secure data access. The summary data is presented in Table 11.

For each of the denormalization steps, the interaction with the secure access models yielded mixed results. This observation is not surprising given that these steps were largely designed to accommodate the constraints of the mobile database setup and minimize communication overhead and traffic volume. Security, on the other hand, generates overhead in its effort to conceal and protect data.

Concluding this analysis section is an overall evaluation of how the secure access models fared in their testing against the mobile database characteristics. We begin by studying the performance of each access model. Of the three secure access models, RBAC performed the least well. It lacks the granular control necessary to enforce user access rules or protect against unauthorized entry by external actors. RBAC's most attractive aspect is its simplicity and flexibility but that benefit has clearly come at the cost of stringent security. MAC contains a trade-off almost opposite to that of RBAC. It provides robust security by classifying both users and data items. However, this dual

classification makes the security model inflexible and difficult to implement. Moreover, in the mobile computing environment, the overhead generated by MAC can tax an already constrained system and degrade system performance. DAC provides a "middle-of-the-road" alternative to RBAC and MAC. Like RBAC, DAC is flexible and does not impede system performance. Unlike RBAC, DAC's emphasis on user privileges provides some granularity. This granularity allows DAC to satisfy some security requirements, such as user authentication, that RBAC cannot meet. Like MAC's polyinstantiation, DAC's views mechanism prevents inappropriate data access and disclosure. However, DAC's inattention to data item classification makes its overall security far less robust than the MAC access model.

The second point within this overall evaluation is some combination of the access models generally performed better than the access models by themselves, with one exception. In all three testing categories (system potential threats, system security requirements, and select denormalization steps) a combination of models provided more rigorous security than an individual access model could provide. For example, combining MAC and DAC models provides robust security while preventing the blind write problem MAC possesses. The one exception deals with the overhead generated by MAC model. As noted in the previous paragraph, MAC's overhead can tax already constrained mobile systems and none of the other secure access models was able to ameliorate this fact.

Table 9: Comparison of Secure Access Models Against System Potential Threats

	DAC	**MAC**	**RBAC**	**Combination of Models**
Loss of confidentiality	1. Mix.: Authorized system users access somewhat limited by classification. Trojan horse attack still possible, however. 2. Neg.: Lack of data item classification can allow unauthorized intruder access to data. 3. Pos.: Views will enforce appropriate user access	1. Pos.: Legitimate system users access limited by classification. Trojan horse attack not possible. 2. Neg.: Covert channel disclosure possible.	1. Neg.: Model does not have sufficient granular control over users to prevent loss.	1. Mix.: If RBAC & DAC used, system will have sufficient granularity to control user access. Data items still have no classification though. 2. Mix.: If DAC & MAC, data items themselves have protection but covert channels still possible.
Loss of privacy	1. Mix.: Authorized system users access somewhat limited by classification. Trojan horse attack still possible, however. 2. Neg.: Lack of data item classification can allow unauthorized intruder access	1. Pos.: Polyinstantiation will prevent inappropriate data access and disclosure. 2. Neg.: Covert channel disclosure possible.	1. Neg.: Model does not have sufficient granular control over users to prevent loss.	1. Mix.: If RBAC & DAC used, system will have sufficient granularity to control user access. Data items still have no classification though. 2. Mix.: If DAC & MAC, data items themselves have protection

	to data. 3. Pos.: Views will enforce appropriate user access			but covert channels still possible.
Loss of system responsiveness	1. Pos.: Model's simplicity little impedance to system.	1. Neg.: Amount of overhead generated by model can impede data availability and degrade timeliness	1. Pos.: Model's simplicity little impedance to system.	1. Pos.: If RBAC & MAC used, security is enhanced but amount of overhead generated by model may impede data availability and degrade timeliness
Loss of data integrity	1. Neg.: Model provides little protection against malicious internal actors.	1. Mix.: Data items have their own classification and some integrity protected. System vulnerable to blind write problem, however.	1. Neg.: Model does not have sufficient granularity to protect against malicious internal actors.	1. Pos.: If DAC & MAC used, system will sufficient granularity to protect against most integrity losses. Blind write problem averted because user write privileges limited by account or role.
Loss of system integrity	1. Mix.: Model provides some protection against privilege status changes. 2. Neg.: Model contains has no temporal aspect. Status	1. Pos.: Model prevents data from being "written-down". 2. Neg.: Model contains has no temporal aspect. Status cannot be changed temporarily.	1. Neg.: Model contains has no temporal aspect. Status cannot be changed temporarily.	1. Pos.: If DAC & MAC used, system will sufficient granularity to protect against most integrity losses. Blind write problem averted because

	cannot be changed temporarily.			user write privileges limited by account or role.

Table 10: Comparison of Secure Access Models Against System Security Requirements

	DAC	**MAC**	**RBAC**	**Combination of Models**
Access Control	1. Pos.: Model provides for this.	1. Pos.: Model provides for this.	1. Mix.: Model provides for some control with roles but little user granularity	1. Pos.: If RBAC & MAC & DAC, there would be sufficient granular control over users, simplicity of use in role creation, and data item protection through MAC.
User Authentication	1. Pos.: Model provides for this.	1. Pos.: Model provides for this.	1. Mix.: Model provides for some control but little granularity	1. Pos.: If RBAC & DAC, there would be sufficient user control and simplicity of role management.
User Authorization	1. Mix.: Views mechanism provides some protection but data items have no classification	1. Pos.: Polyinstantiation provides for this.	1. Mix.: Roles provide for some authorization rules but lack granularity for individual	1. Pos.: If RBAC & MAC, there would be sufficient user control with polyinstantiation to provide for

	rules.		users.	data protection.
Privilege Level Changes	1. Mix.: Model provides some protection against status changes but Trojan horse attack possible.	1. Pos.: Model provides protection against both user and data privilege changes.	1. Mix.: Model provides some protection against status changes.	1. Pos.: If RBAC & MAC, protection would exist against changes in user and data privileges. Trojan horse attack not possible.
Data Consistency & Integrity	1. Mix.: Model provides minimal protection. Data items possess no protection.	1. Mix.: Model provides protection with exception of blind writes.	1. Neg.: Model provides for little granular control over users. Data items possess no protection.	1. Mix.: No combination provides for more protection than MAC alone.
Data Availability	1. Pos.: Simplicity of model provides little impedance to system.	1. Neg.: Overhead and inflexibility of model can impede system performance.	1. Pos.: Simplicity of model provides little impedance to system.	1. Pos.: If RBAC & DAC, simplicity would provide little impedance but provide granular user control.
Non-repudiation & Accountability	1. Mix.: Model provides some protection but Trojan horse attack can blur accountability.	1. Pos.: Model provides for this.	1. Mix.: Model provides minimal protection for this. Individual user action can be concealed in role.	1. Mix.: No combination provides for more protection than MAC alone.

Table 11: Comparison of Secure Access Models Against Select Denormalization Steps

	DAC	MAC	RBAC	Combination of Models
Data Allocation and Caching on Access Points	1. Mix.: User classifications under model provide some protection to data on access points. Data items themselves have no classification, however.	1. Pos.: Model provides protection to data on access points through user and data item classifications. 2. Neg.: Overhead generated by this model can tax the limited memory of the access point.	1. Mix.: User classifications under model provide some protection to data on access points. User access control has little granularity and data items themselves have no classification, however.	1. Mix.: If DAC & MAC, data would be protected through user and data item classifications. Significant overhead could be generated for access point, however.
Caching Validation Mechanism (i.e., server messaging)	1. Pos.: When server sends updated data to access points and devices, the data does not require its own classification information. Model provides little impedance to system.	2. Neg.: The server sending updated data must send the data and its classification level. Model may provide significant impedance to system.	1. Pos.: When server sends updated data to access points and devices, the data does not require its own classification information. Model provides little impedance to system. 2. Neg.: Model provides little explicit connection between user	1. Mix.: No combination of models provides more stringent security than MAC. 2. Neg.: No model provides security while minimizing overhead.

			and data.	
Pessimistic Concurrency Control Model	1. Pos.: Views mechanism under DAC reinforces strict user access rules and allowable transactions.	1.Pos.: Polyinstantiation under MAC reinforces strict user access rules and allowable transactions. 2. Neg.: Blind write problem under MAC undermine strict concurrency control.	1. Neg.: Model provides little user granularity to reinforce allowable transactions.	1. Mix.: No combination of models provides more stringent security than MAC. 2. Neg.: No model provides security while minimizing overhead.
Four Mode Locking Structure (i.e., read, write, intent to read, intent to write)	1. Pos.: Views mechanism, under DAC, and this locking structure both enforce user access and privilege rules.	1.Neg.: The locking structure was designed to minimize overhead and communication during locking. MAC generates overhead in contrast.	1. Neg.: RBAC not sufficiently granular to link user locking data to a specific account.	1. Pos.: If DAC & RBAC, model would have sufficient granularity to enforce user access while allowing flexible role management.

5. Conclusion

The use of mobile databases is growing within healthcare, while simultaneously, the need for secure data access grows as well. At this intersection of this trend is the need for secure mobile databases. Yet, because of the newness of this trend, and the complexity of mobile database design, several questions remain on how best to secure these data applications.

The work to answer those questions began with a delineation of a prototype mobile database in the preliminary analysis and requirements definition sections. The main observation here is that the issues to be considered are many, the details within those issues are multiple, and the scope difficult to delineate. Preparatory work for designing a mobile database requires consideration of factors ranging from Wi-Fi specifications, to the risk analysis of potential security threats, to how patient medical information is used within a hospital. Building on the work of these earlier sections, a conceptual database design for the mobile EMR was constructed and reviewed. Emphasis was placed on designing a system that could be accessed by multiple users with varying access privileges. Views were constructed throughout the database to enforce and maintain user access to the data.

The physical database design section represents the application of some of the mobile computing characteristics identified in the early paper sections to the conceptual database design. Specifically, the effect mobile computing has upon data allocation and caching and transaction management design issues were studied. From these studies, alternative design approaches to resolving these issues in the mobile environment were

evaluated and approaches selected. Once a working model for a mobile database was constructed, the model was then tested against alternative secure access models. In this testing, specific facets of the design model were tested against characteristics of the access models to determine if, and how, the models worked together. Notes were made as to where the models appeared to work collaboratively with one another or at cross-purposes.

None of the secure access models, by itself, provided rigorous security to all facets of the database design. In addition, many characteristics of the access models were in conflict with characteristics of the mobile design. Some security model combinations worked more collaboratively with the design models but they, too, had conflicts. In this research, it was evident that strong security for mobile databases cannot be dependent upon access control alone. Just as it is unwise to rely solely on wireless channel encryption or network security mechanisms, securing mobile databases well requires a layering of access control, encryption, and additional security measures.

6. References

[1] "2001 Technology Trends," Health Information and Management Systems Society, November 2001, < http://www.himss.org/2001Survey/keytrends/keytrends.htm> (1 December 2001).

[2] Committee on Improving the Patient Record, Institute of Medicine. 1997, *The computer-based patient record*, eds. Richard S. Dick, Elaine B. Steen, Don E. Detmer, Washington, D.C.: National Academy Press.

[3] "Handhelds and Security," SANS Institute, 2001, <http://www.sans.org/infosecFAQ/PDAs/hand_sec.htm> (2 December 2001).

[4] Stammer, L. 2001. A show of handhelds. *Healthcare Informatics,* April, 37-44.

[5] Tabar, P. 2000. Data security. *Healthcare Informatics,* February, 28-33.

[6] Amatayakul, M. 2000. Security measures required for HIPAA privacy. *Journal of Healthcare Information Management,* 14, 4-9.

[7] "Security Analysis of the Palm Operating System," @Stake Inc., 2001, <http://www.atstake.com/research/reports/index.html#security_analysis_palm> (August 2001).

[8] Baker, James. (2001, September). *Handhelds and Security: An Oxymoron?* Paper presented at a meeting on the use of handhelds within healthcare, Chapel Hill, NC.

[9] "Company Backgrounder," Palm, Inc., 2001, <http://www.palm.com/about/background.html> (3 December 2001).

[10] Joshi, J.B.D., Aref, W., Ghafoor, A., and E.H. Spafford. 2001. Security models for web based applications. *Communications of the ACM,* 44, 38-44.

[11] Pernul, G., Min Tjoa, A., and W. Winiwarter. 1998. Modelling data secrecy and integrity. *Data Knowledge & Engineering,* 26, 291-308.

[12] Myers, A. and B. Liskov. 2000. Protecting privacy using the decentralized label model. *ACM Transactions on Software Engineering and Methodology*, 9, 410-442.

[13] Forman, G.H. and Z. Zahorjan. 1994. The challenges of mobile computing. *Computer,* 27, 38-47.

[14] Alonso, R., and H.F. Forth, H.F. 1993. Database system issues in nomadic computing. *ACM SIGMOD Record*, 21, 388-392.

[15] Lam, K.Y., Li, G.H., and T.W. Kuo. 2001. A multi-version data model for executing real-time transactions in a mobile environment. *Proceedings of the Second ACM International Workshop on Data Engineering for Mobile and Wireless Access*, 345-352.

[16] Lubinski, A. 2000. Database security meets mobile requirements. *Proceedings of the IEEE International Symposium on Database Technology and Software Engineerin*g, 445-450.

[17] Son, S.H. and C. Chaney. 1998. Supporting the requirements for multi-level secure and real-time databases in distributed environments. In *Database security: status and prospects,* eds. T.Y. Lin and S. Oian, 73-91. New York: Chapman and Hall Publishing.

[18] Blobel, B. 2001. A systematic approach for analysis and design of secure health information systems. *International Journal of Medical Informatics,* 62, 51-78.

[19] Naiburg, E.J. and R.A. Maksimchuk. 2001. *UML for Database Design*. Boston: Addison Wesley.

[20] Naiburg, *UML for Database Design*, 45.

[21] Dunham, M. and A. Helal. 1995. Mobile computing and databases: anything new? *ACM SIGMOD Record,* 24, *4-14.*

[22] "An overview of threat and risk assessment", James Bayne, January 2002, <http://rr.sans.org/audit/risk.php>, (6 June 2002).

[23] "The devil you know", Ronald Mendell, February 2002, <http://online.securityfocus.com/infocus/1543>, (6 June 2002).

[24] Schneier, B. 2001. *Secrets & lies: digital security in a networked world*. New York: John Wiley & Sons.

[25] IEEE 802.11 standards as reported in Phifer, L. "Improving WLAN Security", <http://www.80211-planet.com/tutorials/article/0,,10724_953651,00.html>, (12 July 2002).

[26] "Wireless security", Richard Wagner, December 2001, <http://www.fcc.gov/realaudio/presentations/ 2002/042902/wagner.pdf>, (12 July 2002).

[27] "Security of the WEP algorithm", Nikita Borisov, <http://www.isaac.cs.berkeley.edu/isaac/wep-faq.html>, (20 July 2002).

[28] Naiburg, *UML for Database Design*, 54.

[29] Elmasri, R. and S. Navanthe. 2000. *Fundamentals of database systems*. Boston: Addison Wesley.

[30] Elmasri, *Fundamentals of database systems*, 538.

[31] Mazumdar, S., Mateusz, M. and P.J. Chrysanthis.. 2001. Mobile computing: caching constrained mobile data. *Proceedings of the ACM Tenth International Conference on Information and Knowledge Management*, 442-449.

[32] Jing, J., Helal, A. and A. Elmagarmid. 1999. Client-server computing in mobile environments. *ACM Computing Surveys,* 31*, 117-157.*

[33] Chan, B., Leong, H. Si, A. and K.F. Wong. 1999. MODEC: a multi-granularity mobile object-oriented database caching mechanism, prototype and performance. *Distributed and Parallel Databases*, 7, 343-372.

[34] Barbara, D. and T. Imielinski. 1994. Sleeper and workaholics: caching strategies in mobile environments. *Proceedings of the ACM SIGMOD International Conference on Management of Data,* 1-12.

[35] Tewari, R. and P. Grillo. 1995. Data management for mobile computing on the Internet. *Proceedings of the ACM Annual Conference on Computer Science,* 246-252.

[36] Pitoura, E. and B. Bhargava. 1994. Building information systems for mobile environments. *Proceedings of the ACM Third International Conference on Information and Knowledge Management,* 371-378.

[37] Seydim, A.Y. (Date). An overview of transaction models in mobile environments. *JOURNAL, The paper is prepared for Distributed Database Management course, November 1999*

[38] Elmasri, *Fundamentals of database systems*, 542.

[39] Alonso, R. and H.F. Korth. 1993. Database issues in nomadic computing. *ACM SIGMOD Record,* 22, 388-392

[40]"NSA glossary of terms used in security and intrusion detection", SANS Institute, <http://www.sans.org/newlook/resources/glossary.htm#T>, (20 July 2002).

[41] Phillips, C.E., Ting, T.C. and S.C. Demurjian. 2002. Mobile and cooperative systems: information sharing and security in dynamic coalitions. *Proceedings of the Seventh ACM Symposium on Access Control Models and Technologies,* 87-96.

[42] Elsmari, *Fundamentals of database systems*, 411.

[43] Oh, S. and R. Sandhu. 2002. Role administration: a model for role administration using organization structure. *Seventh ACM Symposium on Access Control Models and Technologies,* 155-162.

[44] Giuri, L. and P. Iglio. 1997. Role templates for content based access control. *Proceedings of the Second ACM Workshop on Role Based Access Control, 153-159.*

[45] Baldwin, R.W. 1990. Naming and grouping privileges to simplify security management in large databases. *Proceedings of the IEEE Symposium on Research in Security and Privacy,* 224-230.

[46] Bertino, E., Catania, B. Ferrari, E. and P. Perlasca. 2001. A logical framework for reasoning about access control models. *Proceedings of the Sixth ACM Symposium on Access Control Models and Technologies,* 41-52.

7. Appendices

Appendix A - Use Cases

This appendix pertains to Section 3.2, Requirements Definition. The use cases below are the detailed expositions of the users and functional requirements of the mobile database system being modeled in this research. The body of text contains a discussion of the requirements but the detailed flow of information and processes are contained within these use cases.

The system users fall into three roles with varying system privileges. In the first role are the attending physician and nurse who have full access, and read/write privileges, upon the EMR data. The second role consists of the consulting physician, other clinician, laboratory staff, and pharmacy staff. These users have limited access, and limited read/write privileges, upon data that pertains to their specialties. The ward clerk is in the last role. This user has limited access, and limited read/write privileges, to patient demographic and administrative data. This user is barred from accessing data with medical observations such as medications and laboratory test values.

Note that all EMR tables do not have an accompanying use case. Access, Update, and Close Use Cases for the remainder of the entities are the same as those for Patient Treatment Plan, and were not written.

Use Case - Provide Medical Care Overview

Use Case Name: Provide Medical Care Overview

Use Case Purpose: The purpose of this use case is to demonstrate the overall conduct of the system as the actor initiates a session, requests a medical record, reviews the medical record, possibly updates the record, and submits and closes the record.

Actors: Attending Physician, Nurse One, Nurse Two, Consulting Physician, Other Clinician, Ward Clerk, Laboratory Staff, Pharmacy Staff

Pre-Conditions: Actor has a legitimate user account and patient identification number with which to query the medical record database.

Post-Conditions: None

Limitations: Queried patient must have a record to view and manipulate.

Event Flow:

A. Medical Record User (MRU) initiates a legitimate access session to the EMR.
B. MRU queries for a specific patient record with patient id.
C. MRU reviews patient demographic info.
D. MRU selects another entity within EMR to view.

E. MRU updates info contained within EMR.
F. MRU submits and closes patient's EMR.
G. MRU logs off EMR system.

Alternate Flow: None identified

Use Case - Access Medical Record Overview
Use Case Name: Access Medical Record

Use Case Purpose: The purpose of this use case is to demonstrate how the medical record is accessed by a medical record user.

Actors: Attending Physician, Nurse One, Nurse Two, Consulting Physician, Other Clinician, Ward Clerk, Laboratory Staff, Pharmacy Staff

Pre-Conditions: The medical record must not have a write lock upon it. Read locks will not affect other users from reading the medical record.

Post-Conditions: The accessed medical record will be inaccessible until the write lock is removed, or until the lock is invalidated because the transaction is judged to be aborted.

Limitations: Write access to the medical record will be denied to other Medical Record Users
if another user has a write lock on the record. Read access will remain available.

Event Flow:
A. The Medical Records User attempts to login to the EMR system.
B. VARIES: Perform Security Verification and view construction use case.
C. The Medical Records User selects the EMR by patient identification number.
D. The patient demographics table, and a listing of the other EMR entities, is made available to the user to read.
E. If the Medical Record User wants to update the record, see Update Medical Record use case.

Alternative Flows:
Condition Triggering Alternate Flow A: The requested medical record is already being used and has a read lock.
A. The MRU is informed there is a read lock on the record, but the user may still view the record.

Condition Triggering Alternate Flow B: The requested medical record is already being used and has a write lock.
A. The MRU is informed there is a write lock on the record and it is unavailable for viewing at this time.

Condition Triggering Alternate Flow C: The MRU is viewing a medical record that another MRU has a read lock upon. The second MRU decides to convert his read lock to a write lock.

A. The first MRU is informed a write lock is now upon the medical record and data elements may be changed.
B. To forestall the possibility of "dirty read" problems, once the EMR is written to and the write lock released, cascading updates will occur on devices with read locks on the particular EMR.

Condition Triggering Alternate Flow D: The EMR System does not recognize Medical Record User as having permission to access Medical Record.
A. Three attempts to login to the EMR system are given. If none are successful, access to the EMR system is denied.
B. The EMR System returns to step A in Event Flow.

Use Case - Update Medical Record Overview
Use Case Name: Update Medical Record

Use Case Purpose: The purpose of this use case is to demonstrate how the medical record is updated by a medical record user.

Actors: Attending Physician, Nurse One, Nurse Two, Consulting Physician, Other Clinician, Ward Clerk, Laboratory Staff, Pharmacy Staff

Pre-Conditions: The medical record user must have a write lock upon the record before updating it.

Post-Conditions: The accessed medical record will be inaccessible to other MRUs until the write lock is removed, or until the lock is invalidated because the transaction is judged to be aborted.

Limitations: Access to the medical record will be denied to other Medical Record Users because another user has a write lock on the record. MRUs with read locks prior to the write lock being granted will follow the procedures outlined in Access Medical Record Overview Use Case, Condition Triggering Alternate Flow C.

Event Flow:
A. The Medical Record User has logged into the EMR system and has selected a patient record (See Access Medical Record Overview Use Case.)
B. The MRU selects the data item to be updated and inserts a new value.
C. The MRU selects the submit button to update the database.

Alternate Flow: None identified.

Use Case: Close Medical Record Overview
Use Case Name: Close Medical Record Overview

Use Case Purpose: The purpose of this use case is to demonstrate how the medical record is closed by a medical record user.

Actors: Attending Physician, Nurse One, Nurse Two, Consulting Physician, Other Clinician, Ward Clerk, Laboratory Staff, Pharmacy Staff

Pre-Conditions: The Access Medical Record use case has occurred. The MRU must have obtained either a read or write lock upon the record before closing it.

Post-Conditions:
A. Closing a medical record will be considered a commit and will follow the unilateral commit protocol outlined in Section 3.4.2.

Limitations: None identified.

Event Flow:
A. The Medical Record User has logged into the EMR system and has selected a patient record (See Access Medical Record Overview Use Case.)
B. The MRU selects the submit button to update the database.
C. The system will then follow the unilateral commit protocol to insert any updated data into the database.

Alternate Flow:
Condition Triggering Alternate Flow A: Because of device disconnection, memory loss, or loss in network connectivity, the commit protocol ends during steps one through three of the protocol.
A. The Medical Record User has logged into the EMR system and has selected a patient record (See Access Medical Record Overview Use Case.)
B. The MRU selects the submit button to update the database.
C. The commit protocol ends during steps one through three.
D. The Coordinator broadcasts an abort, the locks are released, and the transaction is rolled back.

Use Case: Security Verification and View Construction
Use Case Name: Security Verification and View Construction

Use Case Purpose: The purpose of this use case is to demonstrate how the system blocks unauthorized access and verifies authorized access. Upon accessing the system, authorized users interact with views dependent upon their access rights.

Actors: Attending Physician, Nurse One, Nurse Two, Consulting Physician, Other Clinician, Ward Clerk, Laboratory Staff, Pharmacy Staff

Pre-Conditions:
A. Login attempts must be made from a mobile device whose IP and MAC addresses are known to the system.

Post-Conditions:
A. If an unauthorized individual attempts to access this system, this login attempt will be written to a system security log.
B. If an authorized user accesses the system, this login will be noted and access logged.

Limitations: None identified.

Event Flow:
A. Via the device, an individual enters a username and password to access the system. Three attempts are given to login to the system.
B. The individual is recognized as an authorized user and admitted to system.
C. Dependent upon the user's access rights, a view to the database is constructed and enforced for the user.

Alternative Event Flow:
Condition Triggering Event Flow A: System does not recognize individual as authorized user.
A. The individual is not recognized as an authorized user and is not admitted to system.
B. No further login attempts are allowed.
C. The login attempts are written to a security log.

Use Case - Access Patient Demographics
Use Case Name: Access Patient Demographics

Use Case Purpose: The purpose of this use case is to demonstrate how the patient demographics table is accessed by a medical record user.

Actors: Attending Physician, Nurse One, Nurse Two, Consulting Physician, Other Clinician, Ward Clerk, Laboratory Staff, Pharmacy Staff

Pre-Conditions:
A. The MRU has successfully logged onto the EMR system.

B. The MRU has a queried for a patient with a patient identification number recognizable to the system.
C. The MRU has been granted a read or write lock upon a patient's EMR.

Post-Conditions:
A. The accessed medical record will be inaccessible, for write operations, until the write lock is removed, or until the lock is invalidated because the transaction is judged to be aborted.
B. The accessed medical record will be accessible, to other users, if a read lock is held upon it.
C. The MRU will have the option to select other tables within the patient's EMR in addition to reading the patient demographic table information.

Limitations:
A. Write access to the medical record will be denied to other Medical Record Users if another user has a write lock on the record.
B. All authorized system users can access this table.

Event Flow:
A. The Medical Records User has logged on to the EMR system.
B. The MRU submits a query for a patient's EMR with a patient identification number.
C. The patient's demographic table is sent to the MRU, along with the option to select other tables within the patient's EMR.

Alternative Event Flow: None identified.

Use Case - Update Patient Demographics
Use Case Name: Update Patient Demographics

Use Case Purpose: The purpose of this use case is to demonstrate how the patient demographics table is updated by a medical record user.

Actors: Attending Physician, Nurse One, Nurse Two, Consulting Physician, Other Clinician, Ward Clerk

Pre-Conditions: The MRU has successfully logged onto the EMR system, and the Access Patient Demographics use case has occurred. The MRU user has a write lock upon the record.

Post-Conditions:
A. The accessed medical record will be inaccessible until the write lock is removed, or until the lock is invalidated because the transaction is judged to be aborted.
B. The MRU will have the option to select other tables within the patient's EMR in addition to reading the patient demographic table information.

Limitations:
A. Write access to the medical record will be denied to other Medical Record Users if another user has a write lock on the record. Read access will remain available.
B. All authorized system users can update this table

Event Flow:
A. The Medical Records User has logged on to the EMR system.
B. The MRU submits a query for a patient's EMR with a patient identification number.
C. The patient's demographic table is sent to the MRU, along with the option to select other tables within the patient's EMR.
D. The MRU selects the data item to be updated and inserts a new value.
E. The MRU selects the submit button to update the database.

Alternative Event Flow: None identified.

Use Case - Access Patient Medications
Use Case Name: Access Patient Medications

Use Case Purpose: The purpose of this use case is to access the medications list for the patient.

Actors: Attending Physician, Nurse One, Nurse Two, Consulting Physician, Other Clinician, Pharmacy Staff, Ward Clerk

Pre-Conditions:
A. The Select Patient and the Access Patient Demographics use cases have successfully occurred.
B. The MRU has selected the patient medications table after reviewing the list of available tables.

Post-Conditions: The MRU must obtain a read lock, at minimum, for the medications table.

Limitations:
A. Medical Record Users will have varying access privileges upon the medications entity. All users, except the Ward Clerk, can fully access the table. The Ward Clerk user can access a subset of the entity's data elements.

Event Flow:
A. The Medical Records User has logged on to the EMR system.
B. The MRU submits a query for a patient's EMR with a patient identification number.
C. The MRU selects the patient medications table after reviewing the list of available tables.

Alternate Flow: None identified

Use Case - Update Patient Medications
Use Case Name: Update Patient Medications

Use Case Purpose: The purpose of this use case is to update the medications list for the patient.

Actors: Attending Physician, Nurse One, Nurse Two, Consulting Physician, Other Clinician, Pharmacy Staff, Ward Clerk

Pre-Conditions: The Select Patient, Access Patient Demographics, and Access Medications use cases have successfully occurred. The MRU has a write lock upon the record.

Post-Conditions:
A. The accessed medical record will be inaccessible until the write lock is removed, or until the lock is invalidated because the transaction is judged to be aborted.

Limitations:
A. Medical Record Users will have varying write privileges upon the medications entity. All users, except the Ward Clerk, can fully write to the table. The Ward Clerk user can write to a subset of the entity's data elements.

Event Flow:
A. The Medical Records User has logged on to the EMR system.
B. The MRU submits a query for a patient's EMR with a patient identification number.
C. The patient's demographic table is sent to the MRU.
D. The patient's medications table has been selected.
E. The MRU selects the data item to be updated and inserts a new value.
F. The MRU selects the submit button to update the database.

Alternate Flow: None identified

Use Case - Access Patient Laboratory
Use Case Name: Access Patient Laboratory

Use Case Purpose: The purpose of this use case is to update the laboratory values for the patient.

Actors: Attending Physician, Nurse One, Nurse Two, Consulting Physician, Other Clinician, Laboratory Staff, Ward Clerk

Pre-Conditions: The Select Patient and Access Patient Demographics use cases have successfully occurred. In addition, the MRU has selected the patient medications table after reviewing the list of available tables.

Post-Conditions: The MRU must obtain a read lock, at minimum, for the laboratory table.

Limitations: Medical Record Users will have varying update privileges upon the medications entity.

Event Flow:
A. The Medical Records User has logged on to the EMR system.
B. The MRU submits a query for a patient's EMR with a patient identification number.
C. The MRU selects the laboratory table after reviewing the list of available tables.

Alternate Flow: None identified

Use Case - Update Patient Laboratory
Use Case Name: Update Patient Laboratory

Use Case Purpose: The purpose of this use case is to update the laboratory values for the patient.

Actors: Attending Physician, Nurse One, Nurse Two, Consulting Physician, Other Clinician, Laboratory Staff

Pre-Conditions: The Select Patient, Access Patient Demographics, and Access Laboratory use cases have successfully occurred.

Post-Conditions:
A. The accessed medical record will be inaccessible until the write lock is removed, or until the lock is invalidated because the transaction is judged to be aborted.

Limitations:
A. Medical Record Users will have varying write privileges upon the laboratory entity. All users, except the Ward Clerk, can fully write to the table. The Ward Clerk user can write to a subset of the entity's data elements.

Event Flow:
A. The Medical Records User has logged on to the EMR system.
B. The MRU submits a query for a patient's EMR with a patient identification number.
C. The patient's demographic table is sent to the MRU.
D. The patient's laboratory table has been selected.

E. The MRU selects the data item to be updated and inserts a new value.
F. The MRU selects the submit button to update the database.

Alternate Flow: None identified

Use Case - Access Patient Treatment Plan
Use Case Name: Access Patient Treatment Plan

Use Case Purpose: The purpose of this use case is to access the treatment plan for a patient.

Actors: Attending Physician, Nurse One, Nurse Two, Consulting Physician, Other Clinician

Pre-Conditions: The Select Patient and Access Patient Demographics use cases have successfully occurred.

Post-Conditions:
A. The accessed medical record will be inaccessible until the write lock is removed, or until the lock is invalidated because the transaction is judged to be aborted.

Limitations: Medical Record Users will have varying access privileges upon the treatment plan entity.

Event Flow:
A. The Medical Records User has logged on to the EMR system.
B. The MRU submits a query for a patient's EMR with a patient identification number.
C. The MRU selects the patient treatment plan table after reviewing the list of available tables.

Alternate Flow: None identified

Use Case - Update Patient Treatment Plan
Use Case Name: Update Patient Treatment Plan

Use Case Purpose: The purpose of this use case is to update the treatment plan for a patient.

Actors: Attending Physician, Nurse One, Nurse Two, Consulting Physician, Other Clinician

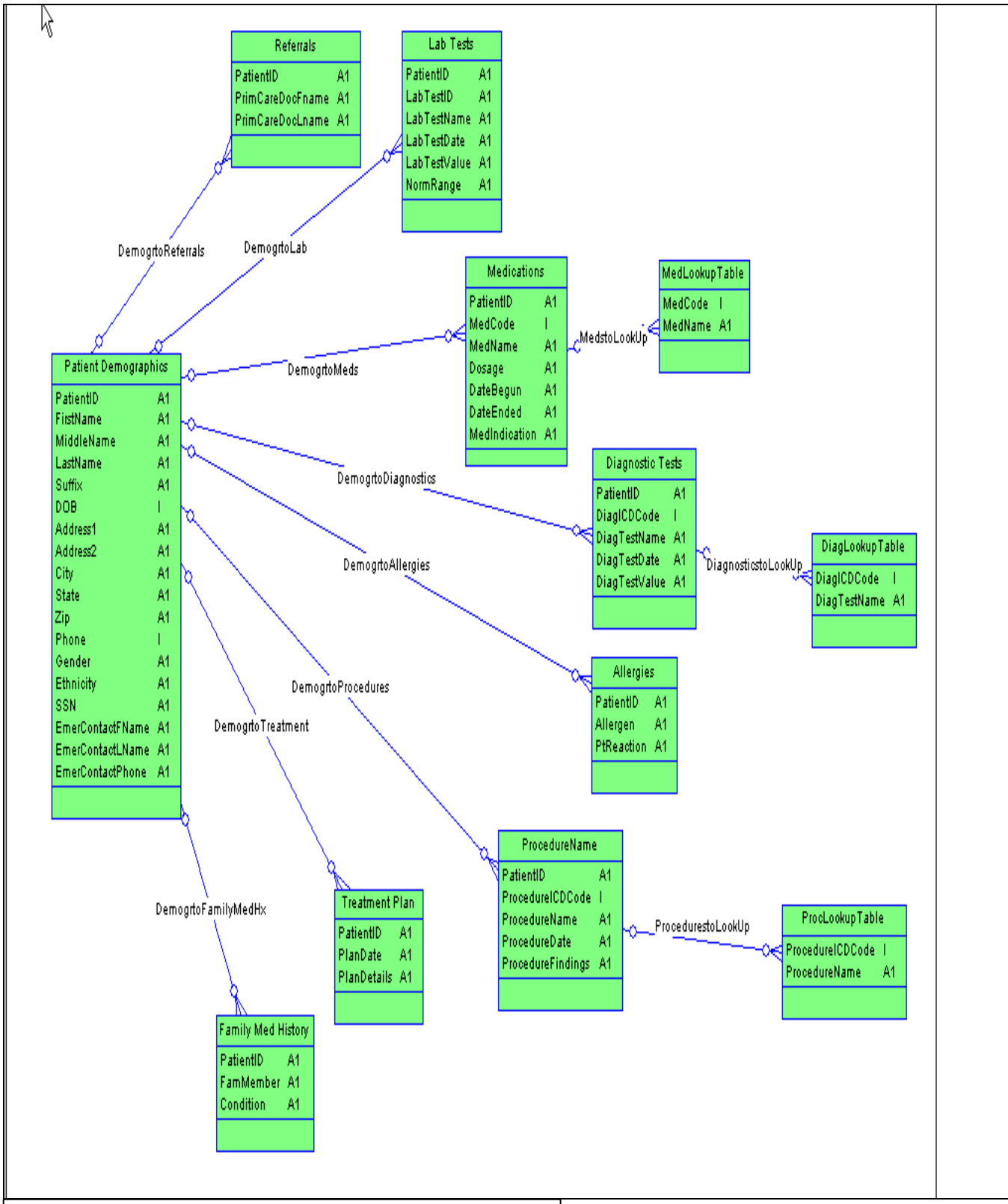
Pre-Conditions: The Select Patient and the Access Treatment Plan use cases have successfully occurred.

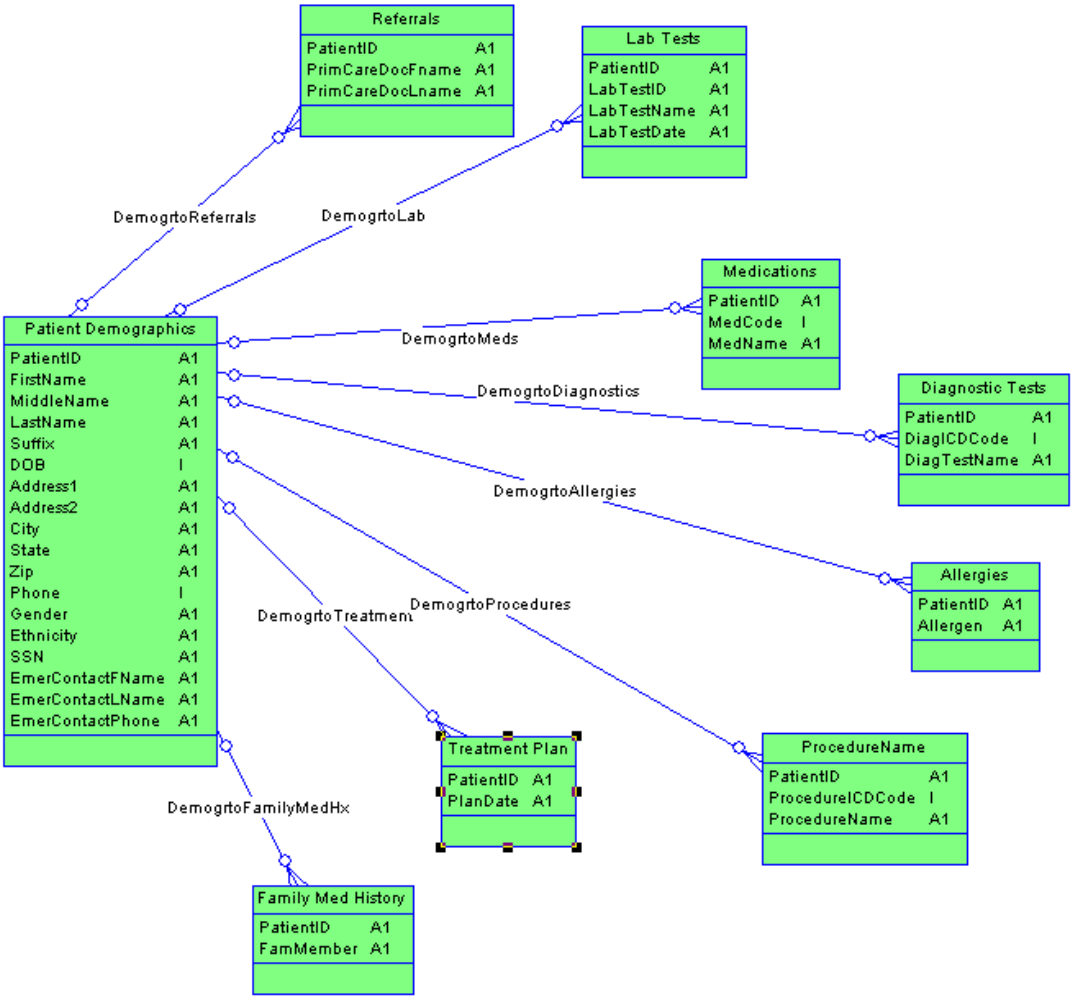
Post-Conditions:
A. The accessed medical record will be inaccessible until the write lock is removed, or until the lock is invalidated because the transaction is judged to be aborted.

Limitations:
A. Medical Record Users will have varying write privileges upon the patient treatment plan entity. All users, except the Ward Clerk, can fully write to the table. The Ward Clerk user can write to a subset of the entity's data elements.

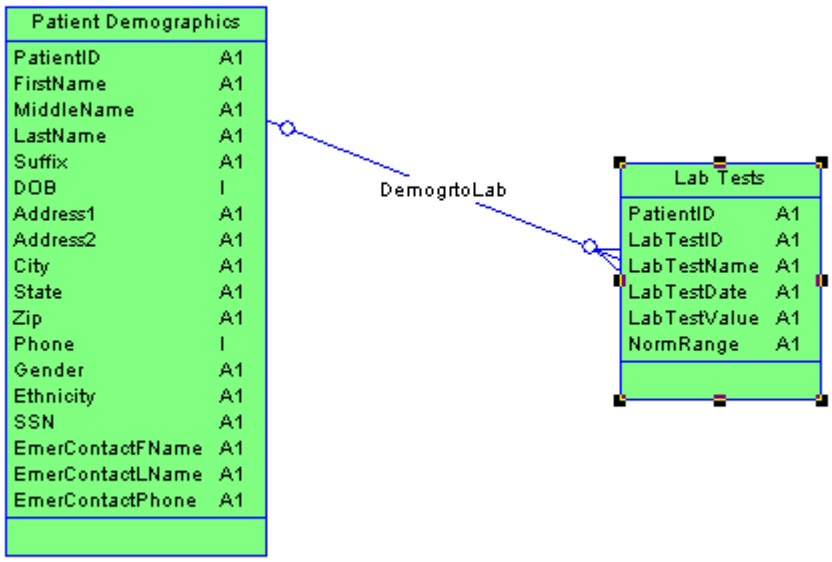
Event Flow:
A. The Medical Records User has logged on to the EMR system.
B. The MRU submits a query for a patient's EMR with a patient identification number.
C. The patient's demographic table is sent to the MRU.
D. The patient's treatment plan table has been selected.
E. The MRU selects the data item to be updated and inserts a new value.
F. The MRU selects the submit button to update the database.

Alternate Flow: None identified

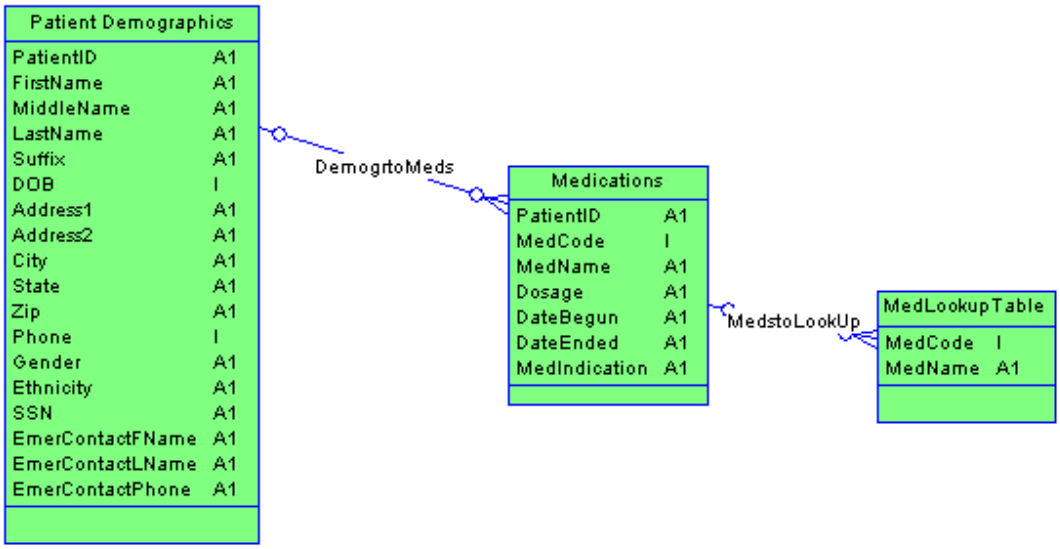

Referrals
PatientID A1
PrimCareDocFname A1
PrimCareDocLname A1
Lab Tests
PatientID A1
LabTestID A1
LabTestName A1
LabTestDate A1
LabTestValue A1
NormRange A1
DemogrtoReferrals
DemogrtoLab
Medications
PatientID A1
MedCode I
MedName A1
Dosage A1
DateBegun A1
DateEnded A1
MedIndication A1
MedLookupTable
MedCode I
MedName A1
MedstoLookUp
Patient Demographics
PatientID A1
FirstName A1
MiddleName A1
LastName A1
Suffix A1
DOB I
Address1 A1
Address2 A1
City A1
State A1
Zip A1
Phone I
Gender A1
Ethnicity A1
SSN A1
EmerContactFName A1
EmerContactLName A1
EmerContactPhone A1
DemogrtoMeds
DemogrtoDiagnostics
DemogrtoAllergies
DemogrtoProcedures
DemogrtoTreatment
DemogrtoFamilyMedHx
Diagnostic Tests
PatientID A1
DiagICDCode I
DiagTestName A1
DiagTestDate A1
DiagTestValue A1
DiagnosticstoLookUp
DiagLookupTable
DiagICDCode I
DiagTestName A1
Allergies
PatientID A1
Allergen A1
PtReaction A1
ProcedureName
PatientID A1
ProcedureICDCode I
ProcedureName A1
ProcedureDate A1
ProcedureFindings A1
ProceduretoLookUp
ProcLookupTable
ProcedureICDCode I
ProcedureName A1
Treatment Plan
PatientID A1
PlanDate A1
PlanDetails A1
Family Med History
PatientID A1
FamMember A1
Condition A1



Appendix C: Clerk's View of Entity-Relationship Model



Appendix D: Laboratory Staff Views of Entity-Relationship Model



Appendix D: Pharmacy Staff Views of Entity-Relationship Model