Penerbit Akademia Baru

Journal of Advanced Research Design

Journal homepage:
https://akademiabaru.com/submit/index.php/ard
ISSN: 2289-7984

Healthcare Data System for a Digital Twin of a Chemotherapy Centre

Piya Noeikham[1], Nikorn Sirivongpaisal[1,2,*], Dollaya Buakum[1,2], Phungern Khongthong[3]

[1] Department of Industrial and Manufacturing Engineering, Faculty of Engineering, Prince of Songkla University, Hatyai Campus, Songkhla 90110, Thailand
[2] Smart Industry Research Center, Department of Industrial and Manufacturing Engineering, Faculty of Engineering, Prince of Songkla University, Hatyai Campus, Songkhla 90110, Thailand
[3] Division of Medical Oncology, Department of Internal Medicine, HatYai-Namom Cancer Centre, HatYai Hospital, Hat Yai, Songkhla 90110 Thailand

ARTICLE INFO

Article history:
Received 20 January 2025
Received in revised form 21 February 2025
Accepted 24 May 2025
Available online 2 June 2025

Keywords:
Digital twin; data system; data model; healthcare service

ABSTRACT

The aim of this research is to enhance Digital Twin (DT) research, specifically in healthcare and smart health sectors. The main objective is to bridge the gaps in DT research in real-world healthcare units by proposing a new data model for a healthcare service DT. The research highlights the importance of secure and efficient database design through a case study at Hat Yai-Na Mom Hospital's chemotherapy centre. This innovation integrates a DT into healthcare and fills the gaps in literature on data systems for real healthcare units. The use of SQLite for database development adds uniqueness, making this research a pioneering effort in implementing DTs in a healthcare context. Overall, the research provides valuable insights into database design considerations and possible applications in service system monitoring.

1. Introduction

The present-day markets are characterized by intense competition, as they face new challenges due to the increased significance of software components. As a result, digitalization in manufacturing is regarded as a promising avenue for attaining heightened levels of productivity and service efficiency [1]. Consequently, through the utilization of digitalization technologies, virtual planning of products and processes has become feasible. Subsequently, simulation and optimization tools process, analyze, and evaluate the substantial volumes of data, thereby enabling real-time availability for planning purposes [2]. Michael Grieves and his collaboration with John Vickers of NASA are credited with the origin of the Digital Twin (DT) concept. The concept originated from a presentation at the University of Michigan in 2002, with the objective of establishing a Product Lifecycle Management (PLM) center in the industry. Since then, the fundamental concept of the DT model has remained relatively stable. It is based on the idea that a digital representation of a physical system can exist as a distinct entity, containing identical information to its physical counterpart. This digital

* *Corresponding author.*
E-mail address: nikorn.s@psu.ac.th

https://doi.org/10.37934/ard.133.1.92108

replica, referred to as the "twin," maintains a continuous connection to the physical system throughout its entire lifecycle [3].

Nowadays, research on DT has experienced exponential growth due to the rapid advancement of communication technology, sensor technology, big data analysis, the Internet of Things (IoT), and simulation technology [4,5]. Interest in the DT has grown significantly in academia and industry over the last ten years. Furthermore, the DT paradigm is currently garnering increased attention from researchers across various fields, with a particular emphasis on the realm of smart health. This focus has been greatly spurred by the rapid development of IoT [6-8]. Moreover, as highlighted by [9], the integration of DTs into smart health is still in its nascent stages, signifying that smart health is poised to become the primary application field for DTs in the future. In recent years, Karakra *et al.,* [10-13] developed a hospital DT model based on discrete event simulation. FlexSim Healthcare (HC), a 3D simulation and modeling tool, has been used to model an initial proof-of-concept of the proposed DT framework. They pointed out the process of developing a DT framework dedicated to real-time monitoring of patients' pathways and predicting their near future to handle irregular, unusual, and unexpected behaviors that may happen in hospitals. This helps make the right decision to mitigate unpredictable situations. In addition, they proposed algorithms for initializing and synchronizing the DT. However, an experimental platform has been developed using an emulator (replica hospital). Connecting the DT with the real world is subject to some limitations in their studies [10-13]. Therefore, future research in this field of DT in hospitals should apply with a real medical unit instead of an emulator to explore if any further issues could be found and need to be solved in the actual working environment. To apply the DT to the existent medical system, a healthcare data system is inevitably mandatory in fulfillment of the mission. A healthcare data system is a computer system that aims to make patients' data available for staff to address healthcare service problems. A data model is required to develop a DT for healthcare purposes in the real world. However, to our best knowledge, no past study focused on data systems for a DT of a real healthcare unit. To prevent the deficiency of a data system that could be a barrier to the successful implementation of a DT, the data system of the DT in a case study of the chemotherapy center department at Hatyai-Namom Cancer Centre is proposed herein. This model serves as an abstract representation that organizes data elements and determines their relationships with each other and real-world entities. Database design is the specific activity involved in this process [14].

A database is the one important component of the DT in terms of data connection. In certain situations, such as the implementation of a DT in healthcare, safeguarding sensitive personal patient information becomes crucial, particularly regarding unrelated or confidential data. In general, there are two types of databases: the standalone unit of the database and the organized database. The primary distinction between these databases lies in the fact that the standalone database is dedicated solely to storing and accessing real-time data for the development of a DT model. In contrast, the organized database stores various types of data, including human resources information, financial records, client data, and more. Due to the sensitivity of patient information, the organized database must implement information management or middleware to ensure security and privacy. In particular, the organization network's DT might establish Application Programming Interfaces (APIs) to securely and reliably handle the storage, management, and retrieval of data records [15]. On the other hand, the standalone database can effectively provide the necessary data with minimal risks in terms of data security. In this study, Structured Query Language (SQL)ite was employed to develop a database for a DT of a chemotherapy center. SQLite is a relational database that is open source and embedded in nature. It was initially introduced in 2000 with the primary purpose of offering a user-friendly method for applications to handle data without the typical complexities associated with dedicated relational database management systems. SQLite is renowned for its exceptional portability, simplicity, compactness, efficiency and reliability [16].

In this study, a novel data model for a DT in healthcare service is proposed to develop a DT in service system monitoring of a healthcare unit. The case study was conducted at the outpatient department chemotherapy center of Hat Yai–Na Mom Hospital, the hospital with the highest potential for treating patients in the lower southern region of Thailand.

2. Literature Reviews

In this section, the data model and data modelling methods are reviewed as follows. Data modelling is a critical process in database design and software engineering that involves creating an abstract representation of data and its relationships within a system. Within the domain of software engineering, data modelling denotes the systematic procedure of formulating a data model by employing formalized data model descriptions and specialized data modelling techniques. Data modelling serves as a methodological approach for delineating the operational prerequisites of a database. This practice is occasionally referred to as database modelling since the resultant data model eventually undergoes implementation within a database system [17]. The term "data model" encompasses dual interpretations. Initially, it denotes a data model theory, which serves as a formal representation detailing the organization and accessibility of data. Secondly, it denotes a data model instance, where a data model theory is implemented to generate a functional data model tailored for a particular application. A data model theory comprises three pivotal constituents: the structural component, which includes data structures employed in constructing databases representing the entities or objects under consideration; the integrity component, which consists of regulations that govern constraints to maintain structural coherence; and the manipulation component, incorporating operators for data modification and querying within the database [18].

According to West's 2011 analysis, essential data characteristics can be classified into three primary categories: (i) Attributes linked to data definition, (ii) attributes related to data content and (iii) attributes that pertain to both definition and content. Relevance, clarity, and consistency are attributes linked to data definition, while timeliness and accuracy are attributes related to data content. Completeness, accessibility and cost pertain to both definition and content, while cost evaluates the expenses associated with data acquisition and its readiness for utilization.

Ensuring that data is relevant is crucial for its applicability within a business context. Additionally, clarity is essential for a well-defined and universally understood data description. Consistency is necessary for data uniformity when sourced from various origins. Timeliness and accuracy are important for data availability and veracity, respectively. Completeness determines the extent to which required data is obtainable. Accessibility examines factors such as the locations, methods, and recipients of data availability, including security considerations. Lastly, the cost aspect evaluates the expenditures associated with data acquisition and its readiness for utilization [19].

To enhance data quality, data modeling commonly employs the Data-Flow Diagram (DFD), which is a graphical representation that delineates the path of data within an information system. A DFD is adept at visualizing data processing within the context of structured design by emphasizing data flow rather than program control flow. Starting with the creation of a context-level data-flow diagram, which illustrates the interactions between the system under examination and external entities, an established practice involves elaborating upon this initial depiction to reveal finer system intricacies and illuminate the intricate data flows that interconnect these constituent parts [20].

There are many types of database models, including the Object-Oriented Database model and the Network model. However, the Relational Database model, developed by Dr. Codd, has been the most widely used and popular choice for managing transactional data in various information systems for decades [21]. A relational database is essentially a collection of tables, or relations and the interconnecting relationships that link them [22]. Each table in a relational database has data fields, rows, a primary key and foreign keys. These tables store records in a two-dimensional structure with columns (representing data fields) and rows, which are called tuples. Importantly, these tables represent entities within the semantic or conceptual model of the database. Each column denotes an attribute of the respective entity, and the primary key, usually made up of either a single attribute or a combination of attributes within the table, uniquely identifies each row. Tables in a relational database model establish and maintain relationships among themselves to represent relevant system-related information. The use of foreign keys facilitates this relationship management by establishing and maintaining references.

The difference between SQL and SQLite lies in their nature and functionality. SQL is a standardized programming language used for managing and manipulating relational databases. It serves as a language for defining, querying, and manipulating data in various relational database management systems (RDBMS), such as MySQL, PostgreSQL, Microsoft SQL Server and others [23]. On the other hand, SQLite is a specific implementation of an RDBMS. It is characterized by being serverless, self-contained, and requiring minimal configuration. SQLite is designed for simplicity and ease of use, making it a popular choice for embedded systems, mobile applications, and scenarios where a standalone database server is impractical. Unlike SQL, which is a language, SQLite is a software library that provides a relational database management system [24-26]. The key differences between SQL and SQLite can be summarized in Table 1.

Table 1
Comparison feature of SQL and SQLite

Feature	SQL	SQLite
Nature	Language	Specific RDBMS implementation as a software library
Deployment	Client-server model	Serverless, embedded within applications
Scope	Standardized language for RDBMS	Self-contained, standalone database engine
Compatibility	Implemented by various RDBMS	Specific implementation with its own characteristics
Installation/Configuration	Separate server installation and setup	Minimal setup, serverless, suitable for embedded scenarios
Concurrency/Transactions	Robust features for concurrency, transactions	Supports transactions and concurrency with some limitations
Use Cases	Wide range of applications, scalable	Embedded systems, mobile applications, lightweight scenarios
Performance	Optimized for high-concurrency scenarios	Performant, suitable for scenarios with lower concurrency
Portability	Language supported across various systems	Lightweight, portable, easy to integrate into applications

3. Data modelling method

A relational database design process [27] was adopted to develop data modelling for a DT of a chemotherapy centre. Several steps need to be followed: gather system requirements using a contextual diagram, design the data input system, and create the physical database. Figure 1 describes the data modelling process.

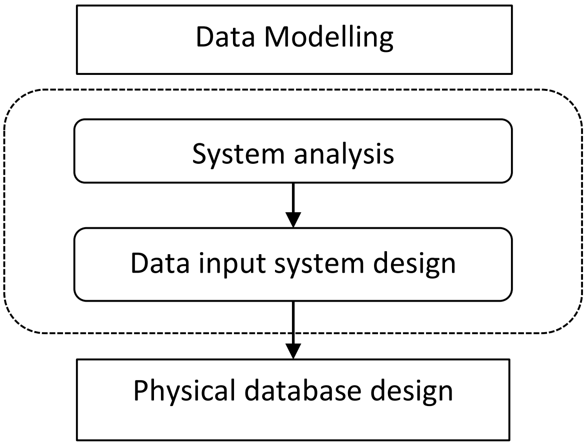


Fig. 1. Data modelling for a DT of a chemotherapy centre

Each step can be summarized as follows.

3.1 Gather System Requirements

Gathering system requirements for data modelling in database design is a critical step in the database development process. Accurately capturing these requirements ensures that the database will meet the needs of the organization and its users. In this research, the data flow diagram, a graphical representation of the "flow" of data through an information system, was used to gather system requirements for outpatient care at the chemotherapy centre of Hat Yai–Na Mom Hospital in Thailand. The data flow diagram is designed to illustrate how the system is divided into smaller components and to emphasize the flow of data between these parts. This diagram is then further detailed to provide a more comprehensive representation of the system being modelled.

To accurately map out the the analysis of the system states, there are 2 main phases for creating Data Flow Diagrams of outpatient care at the chemotherapy centre.

Phase I: Understanding System Requirements

Before creating the data flow diagram, it is crucial to gather information on key stakeholders, including patients, medical staff (e.g., doctors, nurses, nurses and assistants), and IT teams. It is also important to identify data sources such as electronic health records (EHR), IoT sensors, and patient inputs. Additionally, key processes—such as patient data collection, service workflows, and DT processing—must be clearly defined. Data storage and flow mechanisms, including both physical and informational flows, should also be assessed. To effectively gather these requirements, methodologies such as stakeholder interviews and a review of existing system documentation should be employed.

Phase II: Developing the data flow contextual diagram

In this phase, the data flow contextual diagram was designed to illustrate both physical and informational flows in patient care. First, a single process in the system and key entities—such as patients, nurses, doctors, and pharmacists—were defined. Next, data interactions between these entities and each process in the system were mapped. Finally, logical continuity was ensured, verifying that both physical and informational flows progressed smoothly from one process to another without missing connections.

3.2 Data Input System Design

Designing a data input system plays a vital role in building a database that relies on user-generated data. The goal is to create a user-friendly and efficient interface for users to input data accurately and securely. After clearly defining the requirements for the data input system, including the types of data to be collected, the frequency of data input, and any specific data or security requirements, we considered the needs and preferences of users. Consequently, an interface that is intuitive, user-friendly, and aligns with their workflow was designed.

3.3 Physical Database Design

Conceptual data modelling is about understanding the organization and gathering the right requirements. Physical database design, on the other hand, is about creating stable database structures that correctly express the requirements of a database management system (DBMS). Physical design is where we translate the expected schemas into actual database structures.

In this research, the physical database design process ensures real-time data storage and retrieval for patient management and chemotherapy treatment tracking. After gathering all the necessary information to be recorded in the database by analysing the data flow contextual diagram, the database is then designed using SQLite as the Database Management System (DBMS), following these steps:

Step I: Divide the Information into Tables

First, the information items are categorized into major entities such as patient data, room status, bed status, etc. These entities are then mapped to separate tables to ensure structured data storage.

Step II: Define Columns for Each Table

Next, the specific information that needs to be stored on each table is determined. Each data item becomes a field, represented as a column in the table. For example, a patient data table might include fields such as hospital number, queue position, and arrival time.

Step III: Establish Table Relationships

Finally, the relationships between tables are defined. The data in one table may be related to another through common fields. If necessary, additional fields or new tables are created to clarify these relationships and ensure data integrity.

4. Results

4.1 System Analysis

The critical design factors for the DT in a healthcare unit are the physical entities and information existing within an operational environment. To gather the relationship between the physical entities and information in the environmental conditions and constraints, the data flow diagram was adopted. Figure 2 presents the data flow contextual diagram of outpatient care in the chemotherapy centre at Hat Yai–Na Mom Hospital in Thailand.

Based on Figure 2, data will be generated when a patient arrives at each station and receives the required service. Furthermore, data will flow between the workstations. For instance, at station 1.0, patients will be assigned to a queue and measured for their weight, height, heart rate, and blood pressure. This data will then flow as a patient flow from station 1.0 to station 2.0, and so on until the patient finishes treatment.

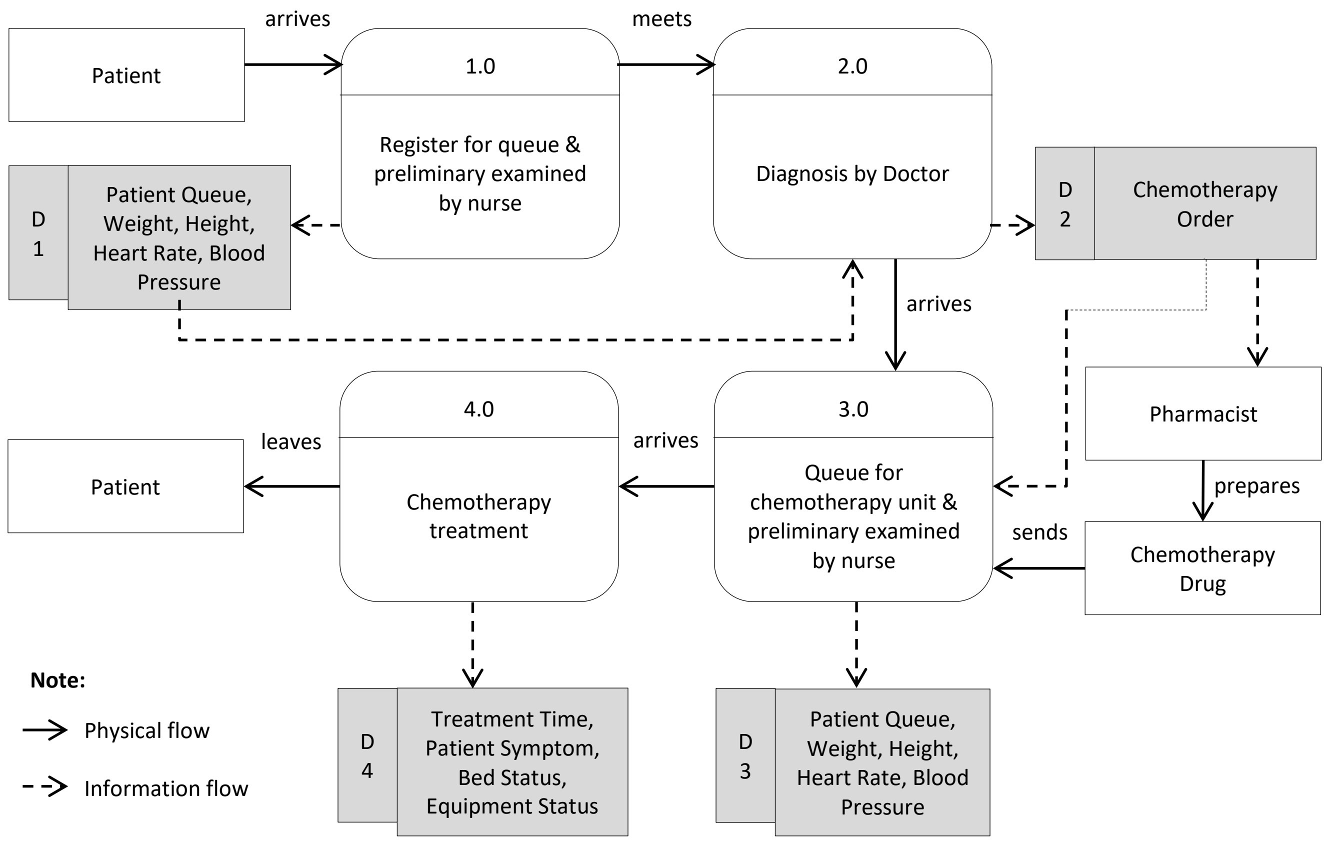


Fig. 2. Data flow diagram of a chemotherapy centre

4.2 Data Input System Design

In this research, the standalone database was applied since it can effectively provide the necessary data with minimal risks in terms of data security. SQLite was employed to develop a database for a DT of a chemotherapy center. As a result of analysing the data flow contextual diagram, eight main stations will be needed to input the necessary data, connected by a local Wi-Fi network for database connection. All relevant data are input into the database manually, adhering to the terms of data security and privacy, as illustrated in Figure 3.

According to the complexity of healthcare-related issues, a particularly large and multidimensional amount of data must be collected and organized for creating DTs and keeping them updated. However, this study focuses on a DT in monitoring service systems in a healthcare unit. Thus, the scopes of the proposed data model are listed as follows:

- The data that needs to be collected in a database for the construction of a DT in a healthcare unit includes data related to physical entities such as patients, arrival time-date, queue, treatment protocol, and bed status.
- The focus group consists of outpatients receiving chemotherapy, encompassing their entire process within the outpatient healthcare system, from initial entry to eventual discharge.
- The procedure for dispensation and payment processing is not included.

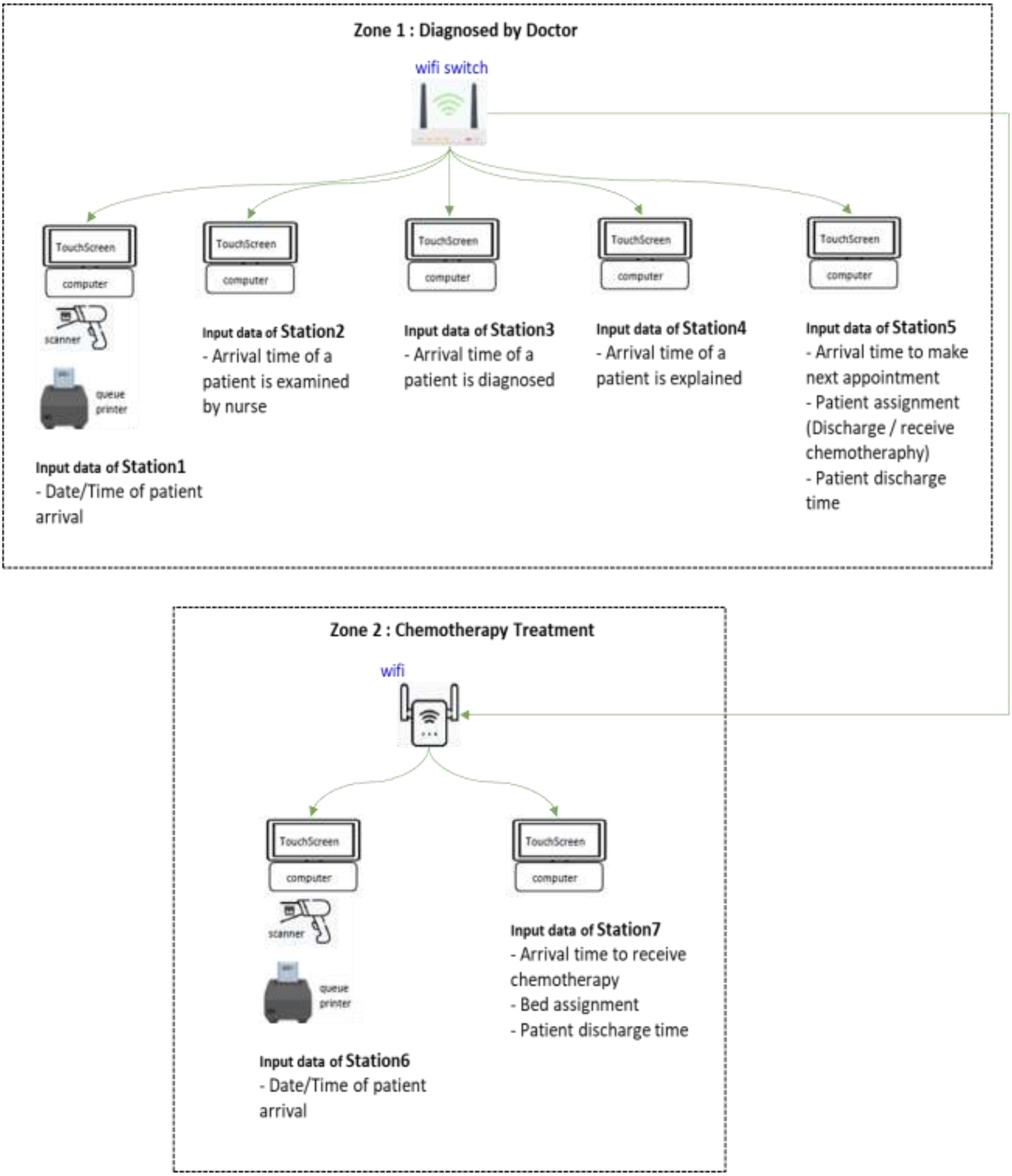


Fig. 3. Data input system

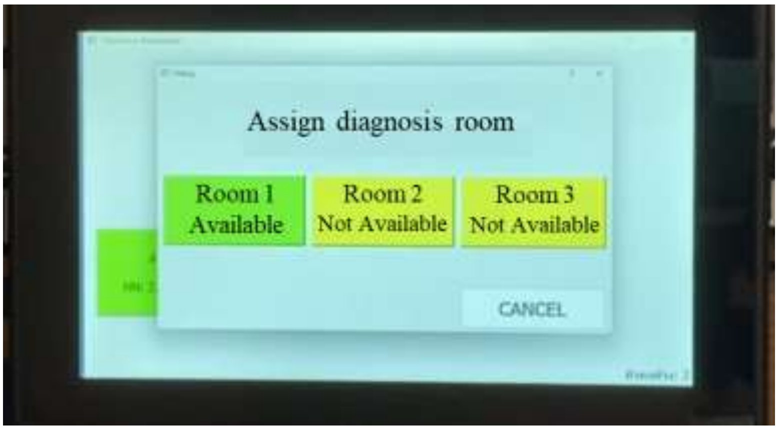
4.3 Physical Database Design

During the physical database design process, all input data, including entities, are mapped to a table. In this system, a single table is created to store data that is input by users through the application. For example, at Station 1, when patients scan the barcode on their appointment, a new row is inserted into the table and the hospital number (HN) is entered through barcode reading. Then, the queue is updated accordingly to the previous row, and the arrival time in Section 1 is updated with a timestamp. It's worth noting that both the date and time are assigned as text to facilitate connection with other software, such as FlexSim Health Care, for developing a DT. Figure 4 displays the equipment for inputting data and the physical database design at Station 1.

Service_log	
Date	text
Hospital number	text
Queue	text
Arrival time	text

Fig. 4. Data Input and storage setup at station 1

Occasionally, certain staff members at the workstation may have to input data under specific conditions. For instance, they may need to assign patients to available diagnosis rooms, chemotherapy beds, or chairs. As a result, additional tables have to be created in the database to support these decisions. However, these tables are independent of one another and don't have any relation. For example, Station 3 updates the arrival time with a timestamp and the room number is entered through an application that can only select available rooms. Hence, it's necessary to create a table for room status in this station. Figure 5 gives an illustration of the application for data input and the physical database design in Station 3.



Service_log	
Arrival time	text
Room number	integer

Room status	
Room 1	text
Room 2	text
…	
Room n	text

Fig. 5. Data Input and storage setup at station 3

The database structure of the case study in this research is illustrated in Figure 6. The primary table is the 'Service_log' table, while the other tables are additional individual tables for supporting patient assignment to rooms, beds, and chairs. Consequently, it is not required to establish relationships between the tables or specify primary keys and foreign keys.

Service_log	
Date	text
Hospital number	text
Queue	text
Arrival time Zone 1	text
Arrival time to examined	text
Arrival time to diagnosed	text
Room number	integer
Arrival time to pharmacist	text
Patient assignment	text
Discharge time Zone 1	text
Arrival time Zone2	text
Queue	text
Arrival time to examined	text
Arrival time to chemotherapy	text
Bed number	integer
Discharge time Zone2	text

Room status	
Room 1	text
Room 2	text
…	
Room n	text

Bed status	
Bed 1	text
Bed 2	text
…	
Bed n	text

Chair status	
Chair 1	text
Chair 2	text
…	
Chair n	text

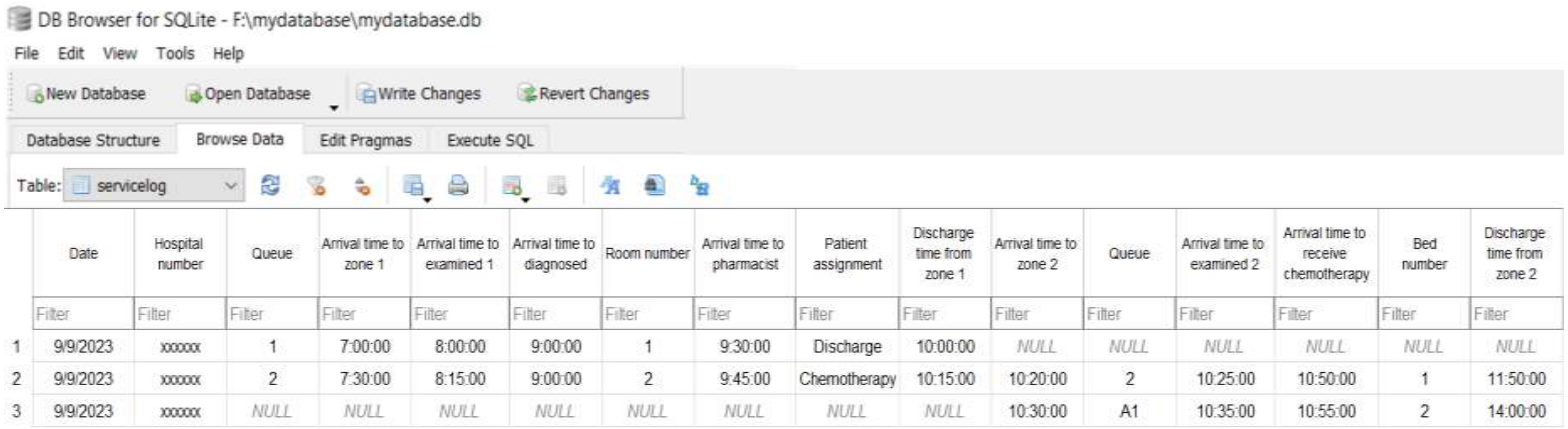
Fig. 6. The database structure of a Chemotherapy Centre

In this step, a database was created on the local network system to store and access real-time data electronically. The database was developed using SQLite for a DT of a chemotherapy centre. A table with 16 columns, corresponding to input data, was generated in the SQLite database, as shown in Figure 7. Each column corresponds to a specific attribute, while the rows represent individual patients. After scanning the patient's barcode from the appointment form, the data was entered into the table *via* a touchscreen application that generated transaction ID1. If a patient does not receive chemotherapy, certain data required for the chemotherapy treatment process is marked as "NULL" in the table, as shown in transaction ID1. Similarly, for patients who only enter the system for chemotherapy, certain data needed for the diagnosis process could be marked as "NULL," as seen in transaction ID3.

4.4 Statistical Analysis

Currently, the health data systems of the chemotherapy department at Hatyai-Namom Cancer Centre rely on an organized database, with data input performed manually. However, the current data systems do not support DT development, as the input data does not cover the entire service process. Therefore, in this section, only diagnosis time is used for explicit comparisons between the current health data systems and the proposed system. Statistical analysis to validate the effectiveness of the proposed system is provided in this section. Following this, a thorough analysis of the data entry error rate is conducted, and a hypothesis test is performed to determine whether the population proportions of the two groups differ, using the normal approximation test. The testing hypothesis is outlined as follows:

Null hypothesis H_0: $p_1 - p_2 = 0$
Alternative hypothesis H_1: $p_1 - p_2 > 0$



	Date	Hospital number	Queue	Arrival time to zone 1	Arrival time to examined 1	Arrival time to diagnosed	Room number	Arrival time to pharmacist	Patient assignment	Discharge time from zone 1	Arrival time to zone 2	Queue	Arrival time to examined 2	Arrival time to receive chemotherapy	Bed number	Discharge time from zone 2
	Filter	Filter	Filter	Filter	Filter	Filter	Filter	Filter	Filter	Filter	Filter	Filter	Filter	Filter	Filter	Filter
1	9/9/2023	xxxxxx	1	7:00:00	8:00:00	9:00:00	1	9:30:00	Discharge	10:00:00	NULL	NULL	NULL	NULL	NULL	NULL
2	9/9/2023	xxxxxx	2	7:30:00	8:15:00	9:00:00	2	9:45:00	Chemotherapy	10:15:00	10:20:00	2	10:25:00	10:50:00	1	11:50:00
3	9/9/2023	xxxxxx	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	10:30:00	A1	10:35:00	10:55:00	2	14:00:00

Fig. 7. The data within the database table

Minitab software computes the P-value for the test. If the obtained *P*-value is greater than 0.05, the null hypothesis (H_0) is not rejected, indicating no significant difference in the data entry error rate between the current health data systems and the proposed system. Conversely, if the obtained P-value is less than or equal to 0.05, the null hypothesis (H_0) is rejected, indicating the data entry error rate by current health data systems (p_1) is significantly higher than the data entry error rate by the proposed system (p_2). The results of the statistical tests for the data entry error rate of diagnosis time between the current health data systems and the proposed system are shown in Table 2.

Table 2
The statistical tests for the data entry error rate of diagnosis time

Descriptive Statistics	Sample	N	Event	Sample p
	Sample 1	535	214	0.400
	Sample 2	263	23	0.087
Estimation for Difference	Difference		95% Lower Bound for Difference	
	0.313		0.267	
	CI based on normal approximation			
Test	Method		Z-Value	P-Value
	Normal approximation		11.40	0.000

According to the population proportions of the data error rate and their corresponding P-values in Table 2, we can claim that the proposed system outperforms current health data systems, primarily due to its capacity to reduce data entry error rate.

5. Discussion and Conclusions

Comparing SQLite to server SQL database engines is not appropriate as it doesn't require extensive administration. It works well in systems that require no expert human support. The suggested data model proposes using a single database table for a DT of a chemotherapy center, eliminating intermediate steps in the database design process, as shown in Figure 8. In Figure 8(a), it can be seen the typical procedure for designing a relational database, which includes several sequential steps such as collecting system requirements for the database, creating the conceptual model (such as an Entity-Relationship Diagram or ER diagram), mapping the conceptual data model to a logical data model, and translating the logical data model into the physical database design. This type of design process suits complex data systems that demand complex relationships in a database and require expertise from the developer. On the other hand, Figure 8(b) proposes a data model that only needs user participation in extracting domain knowledge, which even non-technical users can quickly handle. However, the accuracy of the results depends on the data inputted by healthcare staff.

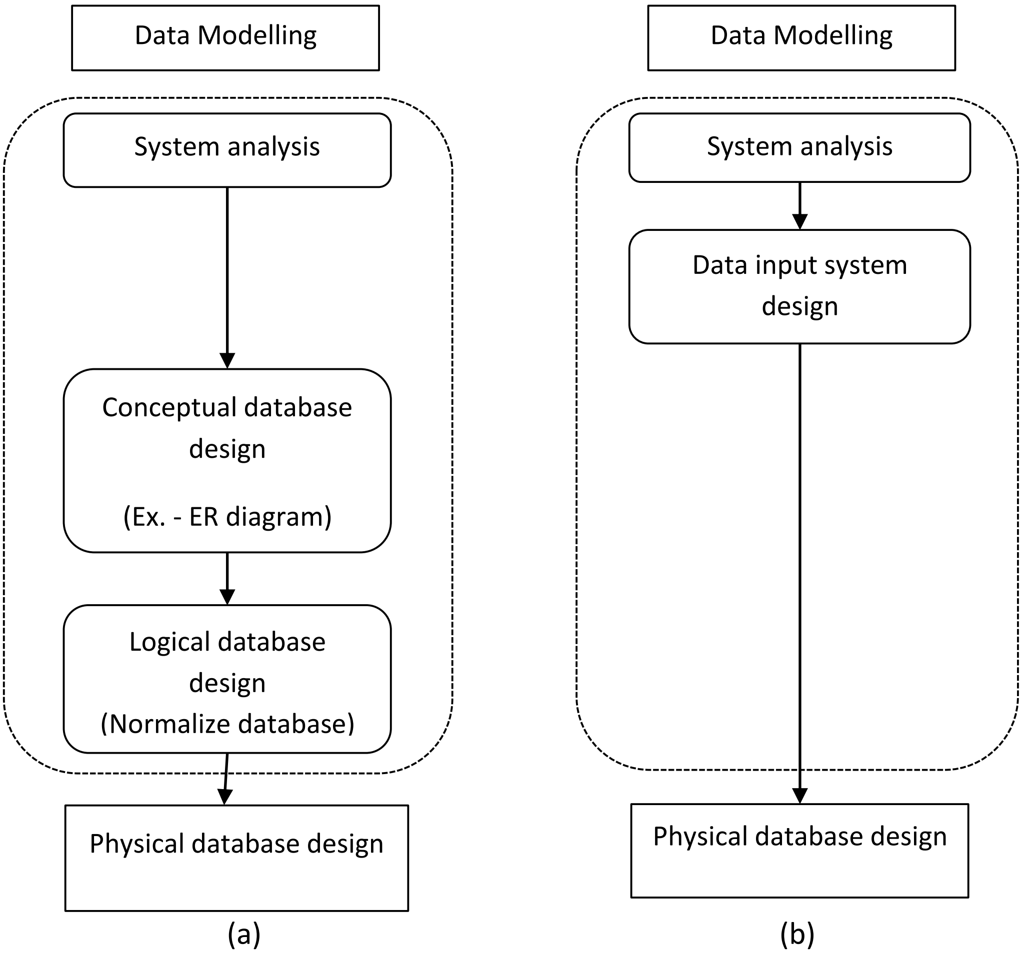
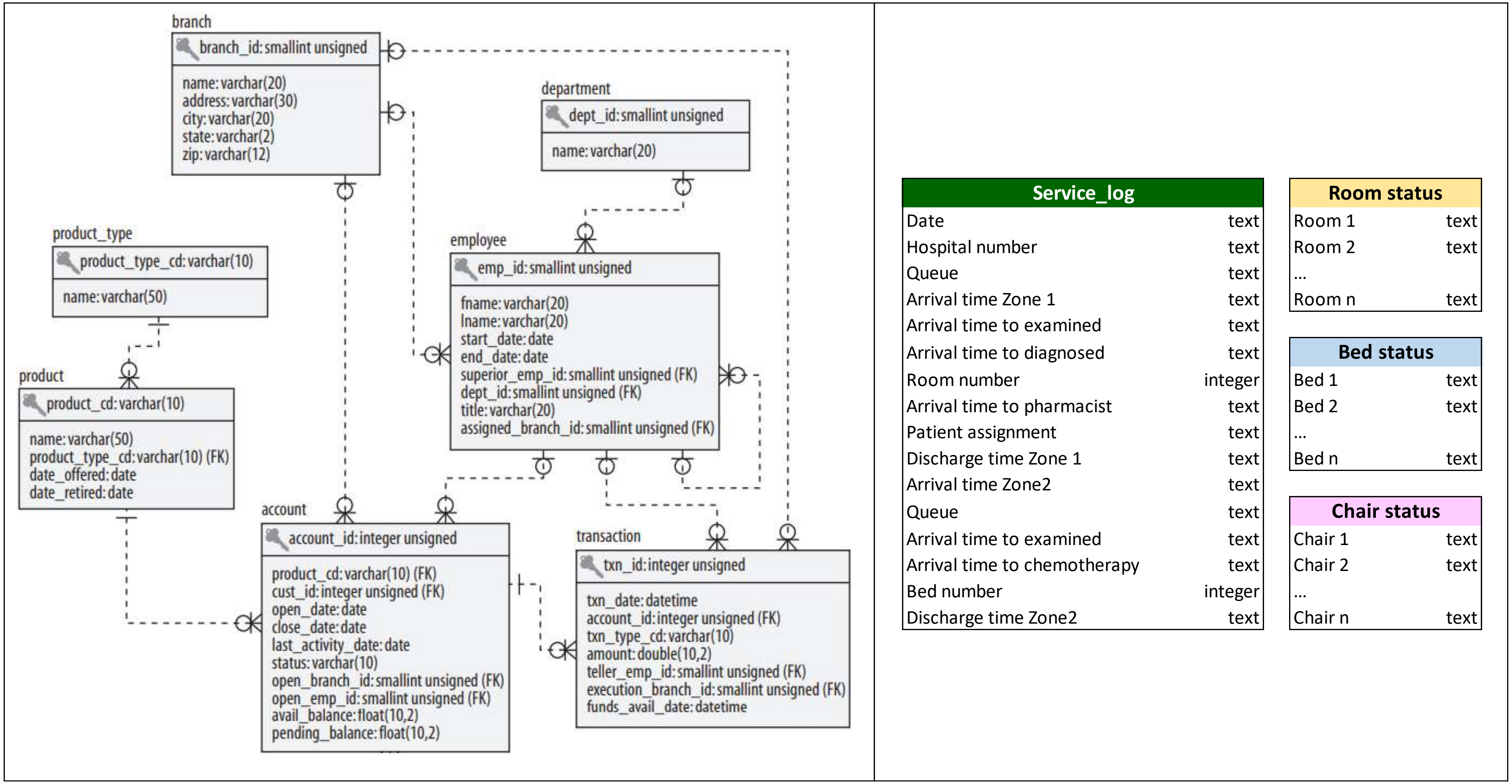


Fig. 8. Data modelling for a DT of a chemotherapy centre (a) Relational Database Design Process [27] (b) The proposed data model

The typical procedure for creating a database structure can result in a complex structure with many tables, relationships, primary keys and foreign keys, as shown in Figure 9(a). This type of database is mainly used for storage, so the schema of entities can be used to identify their data type. In contrast, the proposed procedure creates a simpler database structure, as it doesn't require the identification of entities, primary keys, foreign keys, and table relationships, as shown in Figure 9(b). Dates and times are assigned the data type of text to make it easier to integrate with other software for developing a DT, such as FlexSim Healthcare. In summary, the proposed approach simplifies dynamic data generation without user intervention in technical tasks which is particularly beneficial for non-technical individuals in small and medium-sized healthcare units and support further development of the DT model. The novelty of this research is accredited to the healthcare digital transformation with a DT to remote monitoring of the chemotherapy unit through the integration of undemanding applications, including devices and collective information. Furthermore, the proposed data model addresses the gap in literature regarding data systems for a DT in an existent healthcare unit. The utilization of SQLite for database development adds to the creativity, given its open-source, embedded nature, and suitability for the healthcare context.



(a) (b)

Fig. 9. The comparison of database structures (a) Example of database structure by the typical procedure [28] (b) Database structure by proposed procedure

5.1 Research Limitations

The data in this study is manually entered into the database, which can lead to incorrect data if the staff are not following the rules. Although the proposed system in this study outperforms current health data systems, as its data entry error rate is significantly lower, it still has an error rate of approximately 8%. However, in the healthcare industry, acceptable data entry error rates vary depending on the specific process and its impact on patient care. For instance, the Healthcare Information and Management Systems Society (HIMSS) typically aims for an error rate below 1% in critical healthcare systems, while the American Health Information Management Association (AHIMA) recommends data accuracy levels of at least 98–99% for medical records. Therefore, deploying digital technologies for data input, such as smart sensors and the IoT, paves the way for improved data accuracy and system efficiency. Nowadays, IoT is used to input data into applications. For example, an IoT-based GPS tracking system can help parents monitor their children's safety during commutes [29] and an IoT-based monitoring system can track ammonia, nitrate, pH and temperature in real-time for aquaponic systems [30]. Nevertheless, there are some limitations in implementing these technologies in the healthcare sector, which are outlined in the following main points:

1) **Cost:** The initial setup and maintenance costs of implementing and maintaining these technologies can be significant.
2) **Data Security and Privacy:** The use of automated data input technologies, such as IoT for data collection, raises concerns about the security and privacy of patient data. Ensuring the security, privacy, and resilience of interconnected devices and systems is crucial [31].
3) **Ethical:** Ethical considerations may arise concerning the continuous monitoring of patients and the potential misuse of collected data. Legal frameworks regarding patient consent and data ownership need to be considered carefully.

5.2 Further Applications

The establishment of the data system for the healthcare DT of a Chemotherapy Centre in this research can be deemed as the initiative. The DT concept is still in its early stages in the healthcare domain. So, the ramifications of this technology are just around the corner. The applications of the DT can emerge in other healthcare sectors in two aspects, encompassing a real-time monitoring system and a decision support tool. In the real-time monitoring aspect, the utilization of the DT was primarily directed towards the management of healthcare units. Specifically, monitoring healthcare service systems aimed at identifying bottlenecks in which specific locations within the operational process and devising strategies for operational enhancement. Furthermore, real-time monitoring played a pivotal role in identifying unexpected events from the current state of the real world.

In the context of decision support, a discrete event simulation model built up in the DT was employed to ascertain optimal scenarios applicable to real healthcare units. This involved comprehensive consideration of various factors, including manpower, risks and finances, among others. This enhanced the visibility of implications before decision-makers put plans into action.

Acknowledgement
This research was funded by a grant from Office of the Permanent Secretary, Ministry of Higher Education, Science, Research and Innovation (OPS MHESI), Thailand Science Research and Innovation (TSRI) (Grant No. RGNS 65-228).

References

[1] Uhlemann, Thomas H-J., Christoph Schock, Christian Lehmann, Stefan Freiberger, and Rolf Steinhilper. "The digital twin: demonstrating the potential of real time data acquisition in production systems." *Procedia Manufacturing* 9 (2017): 113-120. https://doi.org/10.1016/j.promfg.2017.04.043

[2] Boschert, Stefan, and Roland Rosen. "Digital twin—the simulation aspect." *Mechatronic futures: Challenges and solutions for mechatronic systems and their designers* (2016): 59-74. https://doi.org/10.1007/978-3-319-32156-1_5

[3] Grieves, Michael, and John Vickers. "Digital twin: Mitigating unpredictable, undesirable emergent behavior in complex systems." *Transdisciplinary perspectives on complex systems: New findings and approaches* (2017): 85-113. https://doi.org/10.1007/978-3-319-38756-7_4

[4] Li, Lianhui, Bingbing Lei, and Chunlei Mao. "Digital twin in smart manufacturing." *Journal of Industrial Information Integration* 26 (2022): 100289. https://doi.org/10.1016/j.jii.2021.100289

[5] E. Glaessgen and D. Stargel, "The DT paradigm for future NASA and US Air Force vehicles," in 53rd AIAA/ASME/ASCE/AHS/ASC structures, structural dynamics and materials conference 20th AIAA/ASME/AHS adaptive structures conference 14th AIAA, 2012, p. 1818. https://doi.org/10.2514/6.2012-1818

[6] Barricelli, Barbara Rita, Elena Casiraghi, and Daniela Fogli. "A survey on digital twin: Definitions, characteristics, applications, and design implications." *IEEE access* 7 (2019): 167653-167671. https://doi.org/10.1109/ACCESS.2019.2953499

[7] Leng, Jiewu, Douxi Yan, Qiang Liu, Kailin Xu, J. Leon Zhao, Rui Shi, Lijun Wei, Ding Zhang, and Xin Chen. "ManuChain: Combining permissioned blockchain with a holistic optimization model as bi-level intelligence for smart manufacturing." *IEEE Transactions on Systems, Man, and Cybernetics: Systems* 50, no. 1 (2019): 182-192. https://doi.org/10.1109/TSMC.2019.2930418

[8] Zhang, Jun, Lin Li, Guanjun Lin, Da Fang, Yonghang Tai, and Jiechun Huang. "Cyber resilience in healthcare digital twin on lung cancer." *IEEE access* 8 (2020): 201900-201913. https://doi.org/10.1109/ACCESS.2020.3034324

[9] Yang, Daoguang, Hamid Reza Karimi, Okyay Kaynak, and Shen Yin. "Developments of digital twin technologies in industrial, smart city and healthcare sectors: A survey." *Complex Engineering Systems* 1, no. 1 (2021): N-A. https://doi.org/10.20517/ces.2021.06

[10] Karakra, Abdallah, Franck Fontanili, Elyes Lamine, Jacques Lamothe, and Adel Taweel. "Pervasive computing integrated discrete event simulation for a hospital digital twin." In *2018 IEEE/ACS 15th international conference on computer systems and Applications (AICCSA)*, pp. 1-6. IEEE, 2018. https://doi.org/10.1109/AICCSA.2018.8612796

[11] Karakra, Abdallah, Franck Fontanili, Elyes Lamine, and Jacques Lamothe. "HospiT'Win: a predictive simulation-based digital twin for patients pathways in hospital." In *2019 IEEE EMBS international conference on biomedical & health informatics (BHI)*, pp. 1-4. IEEE, 2019. https://doi.org/10.1109/BHI.2019.8834534

[12] Karakra, Abdallah, Elyes Lamine, Franck Fontanili, and Jacques Lamothe. "HospiT'Win: a digital twin framework for patients' pathways real-time monitoring and hospital organizational resilience capacity enhancement." In *9th International Workshop on Innovative Simulation for Health Care, IWISH*, vol. 20202020, pp. 62-71. 2020. https://doi.org/10.46354/i3m.2020.iwish.012

[13] Karakra, Abdallah, Franck Fontanili, Elyes Lamine, and Jacques Lamothe. "A discrete event simulation-based methodology for building a digital twin of patient pathways in the hospital for near real-time monitoring and predictive simulation." *Digital Twin* 2 (2022): 1. https://doi.org/10.12688/digitaltwin.17454.1

[14] Forlizzi, Luca, Ralf Hartmut Güting, Enrico Nardelli, and Markus Schneider. "A data model and data structures for moving objects databases." *ACM SIGMOD Record* 29, no. 2 (2000): 319-330. https://doi.org/10.1145/335191.335426

[15] Harper, K. E., S. Malakuti, and C. Ganz. "DT architecture and standards." (2019).

[16] Owens, Mike, and Grant Allen. *SQLite*. New York: Apress LP, 2010.

[17] Bentley, Lonnie D., Kevin C. Dittman, and Jeffrey L. Whitten. *Systems analysis and design methods*. Irwin/McGraw Hill, 2000.

[18] Beynon-Davies, Paul. "Database Systems and Electronic Business." In *Database Systems*, pp. 62-75. Palgrave, London, 2004. https://doi.org/10.1007/978-0-230-00107-7

[19] West, Matthew. *Developing high quality data models*. Elsevier, 2011.

[20] Azzolini, John. "Introduction to Systems Engineering Practices." In *System Engineering Seminar*. 2000.

[21] Sugumaran, Vijayan, and Veda C. Storey. "The role of domain ontologies in database design: An ontology management and conceptual modeling environment." *ACM Transactions on Database Systems (TODS)* 31, no. 3 (2006): 1064-1094. https://doi.org/10.1007/978-0-230-00107-7

[22] Elmasri, Ramez, and Shamkant B. Navathe. "The relational data model and relational database constraints." *Fundamentals of Database Systems* (2013).

[23] I. O. f. Standardization, ISO/IEC 9075, Information technology-Database languages-SQL: international standard. Internat. Inst. for Standardization, 1992.

[24] Feiler, Jesse, and Jesse Feiler. "Understanding What SQLite Is." *Introducing SQLite for Mobile Developers* (2015): 9-14. https://doi.org/10.1007/978-1-4842-1766-5_2

[25] Bhosale, Satish Tanaji, Tejaswini Patil, and Pooja Patil. "Sqlite: Light database system." *Int. J. Comput. Sci. Mob. Comput* 44, no. 4 (2015): 882-885.

[26] Gaffney, Kevin P., Martin Prammer, Larry Brasfield, D. Richard Hipp, Dan Kennedy, and Jignesh M. Patel. "SQLite: past, present, and future." *Proceedings of the VLDB Endowment* 15, no. 12 (2022). https://doi.org/10.14778/3554821.3554842

[27] Uduwela, Wasana C., and Gamini Wijayarathna. "An Approach To Automate The Relational Database Design Process." *Int. J. Database Manag. Syst. (IJDMS)* 7, no. 6 (2015): 49-56. https://doi.org/10.5121/ijdms.2015.7604

[28] Beaulieu, Alan. *Learning SQL: master SQL fundamentals*. O'Reilly Media, 2009.

[29] Hanafi, Hafizul Fahri, Muhamad Hariz Adnan, Miftachul Huda, Wan Azani Mustafa, Miharaini Md Ghani, Mohd Ekram Alhafis Hashim, and Ahmed Alkhayyat. "IoT-Enabled GPS Tracking System for Monitoring the Safety Concerns of School Students." *Journal of Advanced Research in Computing and Applications* 35, no. 1 (2024): 1-9. https://doi.org/10.37934/arca.35.1.19

[30] Sarnin, Suzi Seroja, Mohd Rizal Dohad, Norlela Ishak, Mohd Nor Md Tan, and Ghulam E. Mustafa Abro. "IoT-Based Monitoring and Optimization of Nutrient Film Technique Aquaponic Systems: Influence of Bio-Mechanical Filtration on Growth Rates of Oreochromis Niloticus and Brassica Rapa." *International Journal of Advanced Research in Computational Thinking and Data Science* 4, no. 1 (2024): 10-18. https://doi.org/10.37934/ctds.4.1.1018

[31] Ahmed, Moustafa Abdelrahman Mahmoud, Syahril Anuar Idris, Nur Arzilawati Md Yunus, Fazlina Mohd Ali, and Hazrina Sofian. "CyberShield Framework: Attacks and Defense Modelling for Cybersecurity in Internet of Things (IoT)." *International Journal of Computational Thinking and Data Science* 4, no. 1 (2024): 30-39. http://doi.10.37934/ctds.4.1.3039