ISSN: 1545-679X

Information Systems Education Journal

Volume 7, Number 54 http://isedj.org/7/54/ June 11, 2009

In this issue:

Physical Level Systems Design: a Methodology for Inclusion in our Systems Analysis and Design Textbooks

Paul H. Rosenthal
California State University, Los Angeles
Los Angeles, CA 90032-8123, USA

L. Jane Park
California State University, Los Angeles
Los Angeles, CA 90032-8123, USA

Abstract: This paper presents a recommended information system (IS) physical level design and charting methodology for use in Systems Analysis and Design textbooks. The design method recommended is based on over fifty years of industry experience in systems design and in training IS undergraduate students. The methodology promotes student comprehension and rapid programmer implementation by reducing the time from assignment of system modules to start of programming by up to three weeks. Included in this paper are examples of the methodology including modifications of design charts appearing in several textbooks.

Keywords: Physical System Level Design, Physical Program Level Design, Systems Analysis and Design Textbooks

---

Recommended Citation: Rosenthal and Park (2009). Physical Level Systems Design: a Methodology for Inclusion in our Systems Analysis and Design Textbooks. *Information Systems Education Journal, 7* (54). http://isedj.org/7/54/. ISSN: 1545-679X. (A preliminary version appears in *The Proceedings of ISECON 2008:* §3112. ISSN: 1542-7382.)

This issue is on the Internet at **http://isedj.org/7/54/**

The **Information Systems Education Journal** (ISEDJ) is a peer-reviewed academic journal published by the Education Special Interest Group (EDSIG) of the Association of Information Technology Professionals (AITP, Chicago, Illinois). • ISSN: 1545-679X. • First issue: 8 Sep 2003. • Title: Information Systems Education Journal. Variants: IS Education Journal; ISEDJ. • Physical format: online. • Publishing frequency: irregular; as each article is approved, it is published immediately and constitutes a complete separate issue of the current volume. • Single issue price: free. • Subscription address: subscribe@isedj.org. • Subscription price: free. • Electronic access: http://isedj.org/ • Contact person: Don Colton (editor@isedj.org)

EDSIG activities include the publication of ISEDJ and JISAR, the organization and execution of the annual ISECON and CONISAR conferences held each fall, the publication of the Journal of Information Systems Education (JISE), and the designation and honoring of an IS Educator of the Year. • The Foundation for Information Technology Education has been the key sponsor of ISECON over the years. • The Association for Information Technology Professionals (AITP) provides the corporate umbrella under which EDSIG operates.

Physical Level Systems Design: A Methodology for Inclusion in our Systems Analysis and Design Textbooks

Paul H. Rosenthal
prosent@calstatela.edu

L. Jane Park
jpark@calstatela.edu

College of Business and Economics
California State University, Los Angeles
Los Angeles, CA 90032-8123, USA

Abstract

This paper presents a recommended information system (IS) physical level design and charting methodology for use in Systems Analysis and Design textbooks. The design method recommended is based on over fifty years of industry experience in systems design and in training IS undergraduate students. The methodology promotes student comprehension and rapid programmer implementation by reducing the time from assignment of system modules to start of programming by up to three weeks. Included in this paper are examples of the methodology including modifications of design charts appearing in several textbooks.

Keywords: Physical System Level Design, Physical Program Level Design, Systems Analysis and Design Textbooks

1. INTRODUCTION

This paper proposes an application system physical design approach that is easily understood by users, and easily used by programmer analysts during implementation. The system analysis and design textbooks reviewed approach only the individual program-level physical design process instead of including methodologies for overall system design. The approach presented in this paper is oriented toward the design of multi-program systems involving differing technologies, personnel, locations, and time periods.

Both overall systems level and program level physical designs are normally created from Data Flow Diagram (DFD) and Entity Relationship Diagram (ERD) based logical designs, by separating processes and data stores by time (such as daily vs. monthly, day vs. night), place (client or server, centralized vs. distributed), online vs. batch, and manual vs. automated.

None of these design decisions is fully described or illustrated in any of the textbook examples shown later in this paper. Additionally, proper separation of data flow vs. paper flows, and manual vs. computer processing is almost never mentioned.

The overall information system design process is shown in Figure1. Note the progression from logical business process design, to logical information systems design, and then to physical information systems design. This last step leads naturally to the topics of system estimating and implementations scheduling which with physical systems design are also almost never included in the texts.

2. CREATING A PHYSICAL DESIGN

Figure 2 presents an overall systems level physical design approach of a country club restaurant application using VISIO symbols. The application is modularized across time and should allow programmers to produce a well-structured program. Students presented with this type of chart have been able to easily create the four detailed program level designs needed to implement the system. This level of physical charting illustrates the recommended step needed between logical designs and programming.

The key point in the methodology shown is that users can immediately understand and approve it and programmers can immediately start program and procedure design. The business processes involved have been reduced to procedures with defined functions and interfaces.

The published methodology closest to that shown in Figure 1 (see appendix) is a "distributed systems architecture" approach presented in Whitten. He states that

> *"The use of logical DFD's to model process requirements is a fairly accepted practice. However, the transition from analysis-oriented logical DFD's to design oriented physical DFD's has historically been somewhat mysterious and elusive. We desire a high-level general design that can serve as application architecture for the system, and as a general design for the processes that make up the system. At the same time, we don't want to get caught up in a counterproductive modeling exercise that slows our progress in system design and rapid application development. Simply stated, we want a blueprint to guide us through detailed design and construction. The blueprint will identify design units for detailed specification or rapid system development, whichever is most productive in our project." (Whitten, pages 503-504)*

Whitten's methodology for producing 1) a network architecture, 2) a data distribution and technology assignment, 3) process distribution and technology assignments, and 4) person/ machine boundaries is applicable to this paper's methodology if care is taken to consider time, place, network structure, batch processing and other physical level requirements.

The type of charting shown in Figure 2 should be included in all systems analysis and design textbooks. They should illustrate application system level physical design to the point at which an implementation team can begin program design, and should use the picture type symbols shown since both users and managers can understand them.

3. PROGRAM LEVEL DESIGN EXAMPLE

Figure 3a illustrates an automated teller machine's (ATM) processing chart from Langer (page 72). It omits several key functions including the start and stop steps. It is incomplete in functional scope and difficult to understand because of the lack of separation of manual and automated processes.

Figure 3b adds the start and stop functions omitted from the previous illustration and separates manual and automated activities. It should be understandable by both users and implementers.

The textbook examples give students and business users a false sense of the complexity of ATM processing. The expanded chart with its startup process and separation of manual and computer processes will perhaps start answering the ever present question of "Why does it take so long and cost so much."

4. APPLICATION SYSTEM LEVEL DESIGN EXAMPLE

The following chart presents an example of a physical design from a popular textbook. It illustrates the simplification trend of text book charting illustrations that has made it difficult for students to learn to create realistic programmable designs.

Figure 4a is an illustration of a TPS application combing online and batch processing from Shelly (page 386). It omits the basic concept of differences in time that is fundamental to the separation of these processes.

Figure 4b illustrates this separation of batch and online as well as the idea of a time trigger. It is easier for students and users to understand the scope of implementation from Figure 4b than from the simplified version in Figure 4a.

5. INFORMATION SYSTEM LEVEL PHYSICAL DESIGN EXAMPLE

The Shelly and Cashman System Analysis and Design textbook included a country club case from which Figures 2 and 5 were developed. Figure 5 was used for this case as a systems design to assign student system development assignments. It is an example of the type of overall information system level physical design needed before detailed design and implementation can be started.

Student groups were assigned to implement each of the following sub-systems.

- Managers Office design
- Golf/Tennis Store design
- Restaurant design (see Figure 1)
- Batch Processing design
- MIS Systems design
- Architecture design

The student teams were able to produce quality forms and output designs, manual processing procedures and controls, and database and program design documentation.

6. SUMMARY

Several of the system analysis and design texts list in the bibliography have been used with mixed results. When assigned a fairly complex application such as the country club example for a term project, several weeks were required before the students were able to start program and manual procedure design. This was caused by the lack in the texts of procedures suitable for physical design of multi-program IS applications. Therefore, the authors' methodology shown in this paper has been effective as a supplement to the text approach. The key to the charting methodology's effectiveness (as illustrated in the examples) is the inclusion in the design of both manual and automated procedures and the separation of processes by time and place of actions. This type of charting appears to save several weeks of frustration for students, and is therefore recommended.

Decades of using these classical procedural approaches have shown that business users understand both what is being done and how it is being done.

7. ANNOTATED BIBLIOGRAPHY

This section includes a listing of selected System Analysis and Design textbooks including comments on their system level and program level coverage of logic level and physical level design charting.

Dennis, A., B. H. Wixom, and R. M. Roth (2006). Systems Analysis & Design: Third Edition. John Wiley & Sons.

Combines program level logical and physical design using DFD and Structure Charts.

Hoffer, J. A., J. F. George, and J. S. Valacich (2008). Modern Systems Analysis and Design: Fifth Edition. Pearson Prentice Hall.

Detailed description of physical design at the individual program level for interface, processing, and databases.

Kendall, Kenneth E., and Julie E. Kendall (2006). Systems Analysis and Design: Seventh Edition. Pearson Prentice Hall.

Presents several program level physical designs using DFD methodology.

Langer, Arthur M. (2005). Analysis and Design of Information Systems: Second Edition. Springer-Verlag, New York.

Uses DFD concepts and Object charting symbols.

Pressman, Roger S. (2004). Software Engineering: A Practitioner's Approach: 6 Edition, McGraw-Hill Companies.

This advanced systems analysis and design textbook's illustrations are at the programming level for mechanization type applications.

Satzinger, J. W., R. B. Jackson, and S. D. Burd (2005). Object-Oriented Analysis and Design with the Unified Process. Thomson Course Technology, Boston.

Has a primarily program logic level orientation.

Shelly, G. B., T. J. Cashman, and H. J. Rosenblatt (2006). Systems Analysis and Design: Sixth Edition. Thomson Course Technology, Boston.

Discusses both online and batch analysis and design; logical design includes primarily interface and data structures; physical design is very limited in scope.

Whitten, J. L., L. D. Bentley, and K. C. Dittman (2007). Systems Analysis and Design Methods: Seventh Edition. McGraw-Hill Irwin.

Presents a detailed online oriented physical design methodology. No illustration of batch physical design is presented. The online physical data flow diagram method demonstrates 1) network architecture, 2) data distribution and technology assignment, 3) process distribution and technology assignment, and 4) person/machine boundaries.

APPENDIX

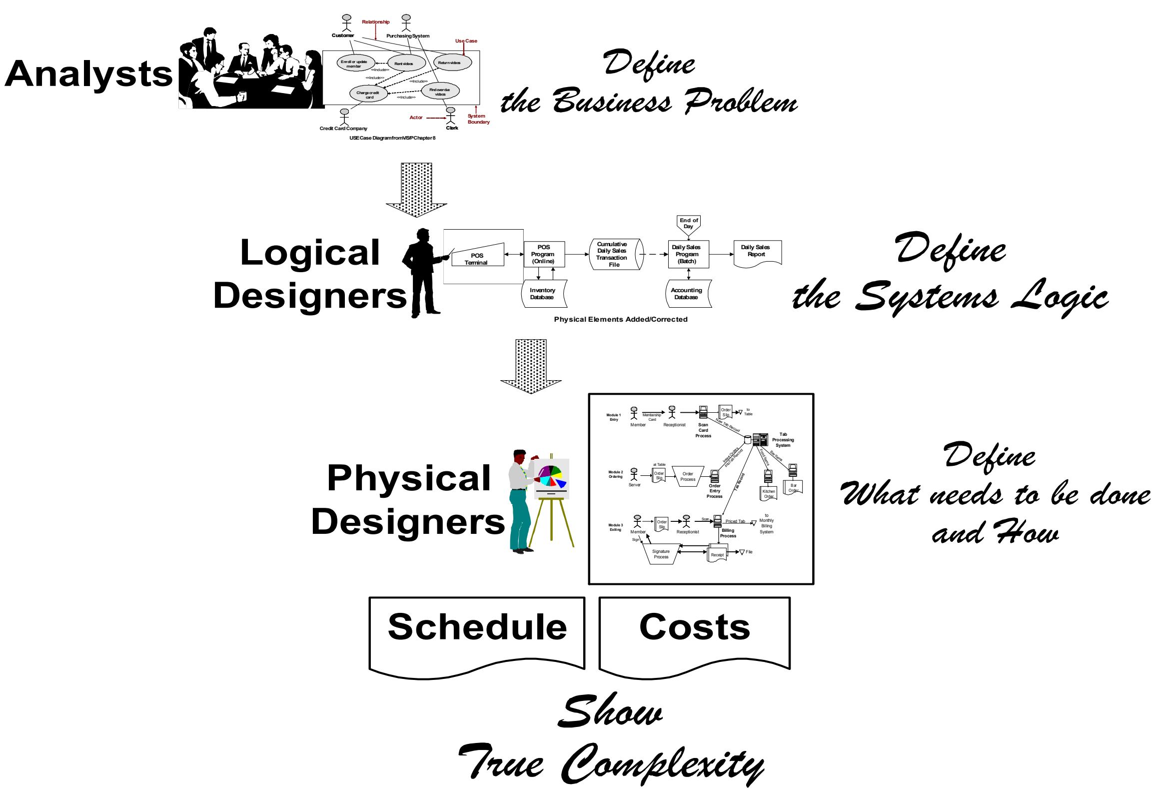

Figure 1: The Planning/Justification Life-Cycle

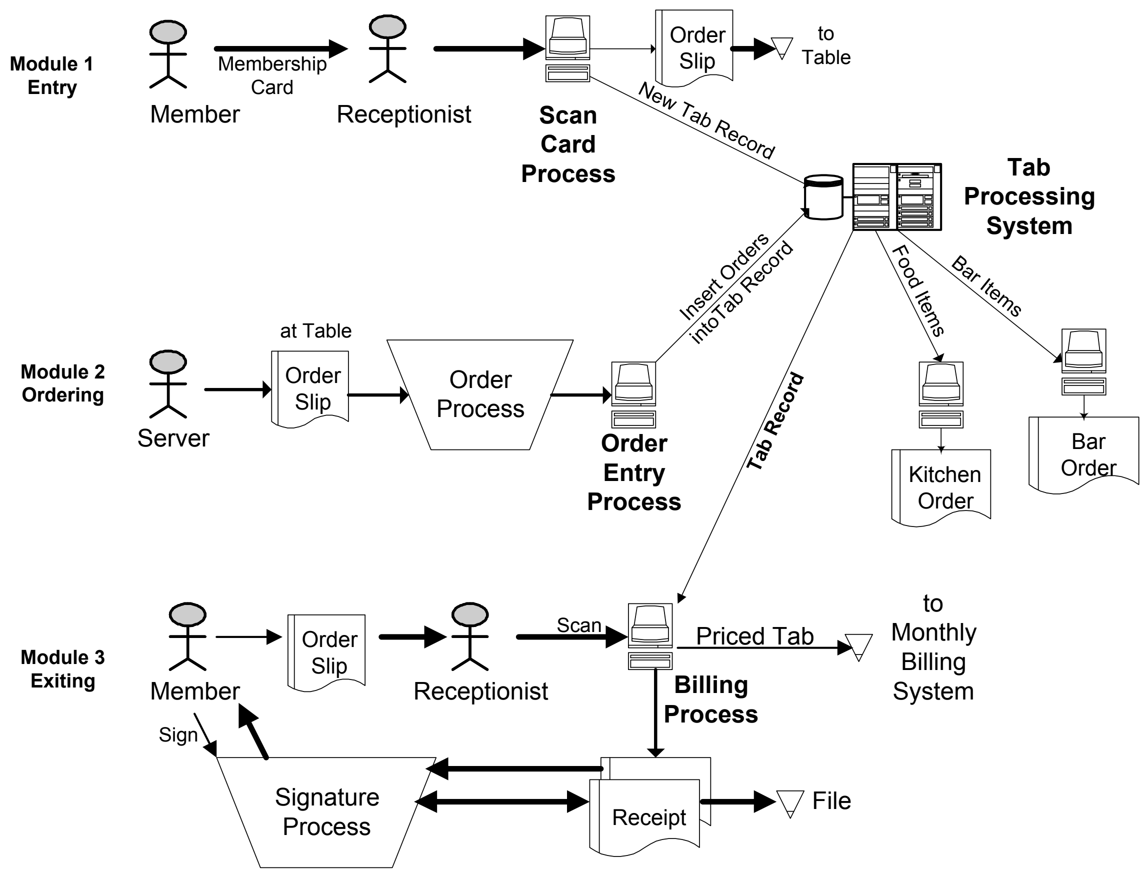

Figure 2: Physical design Example (Country Club Restaurant)

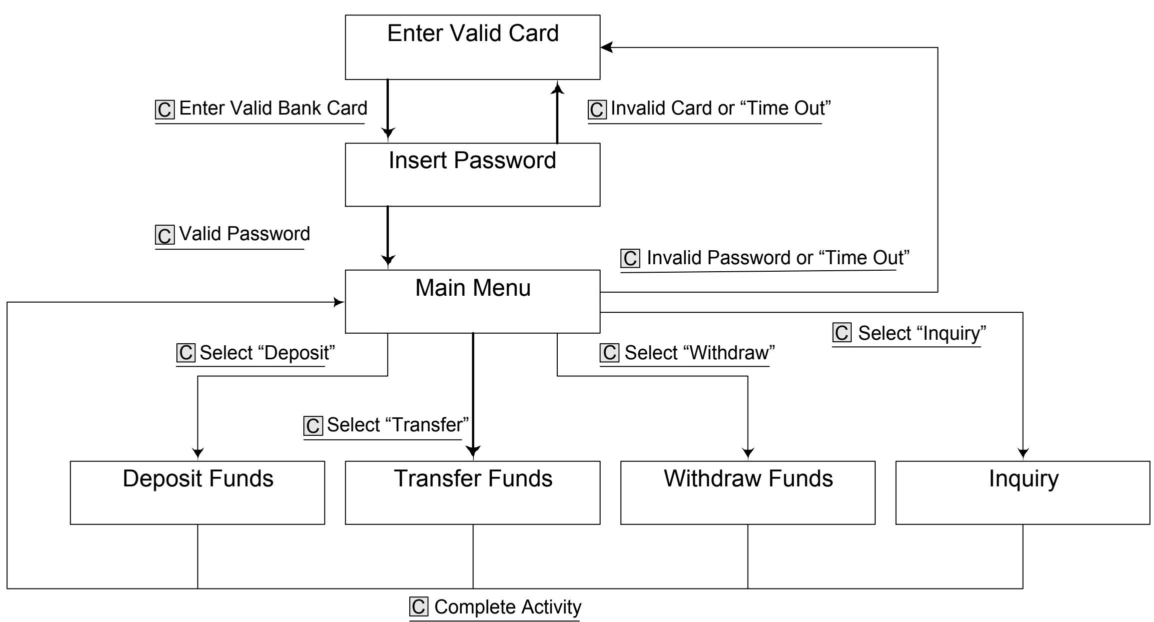

Figure 3a: ATM Process from Langer Text

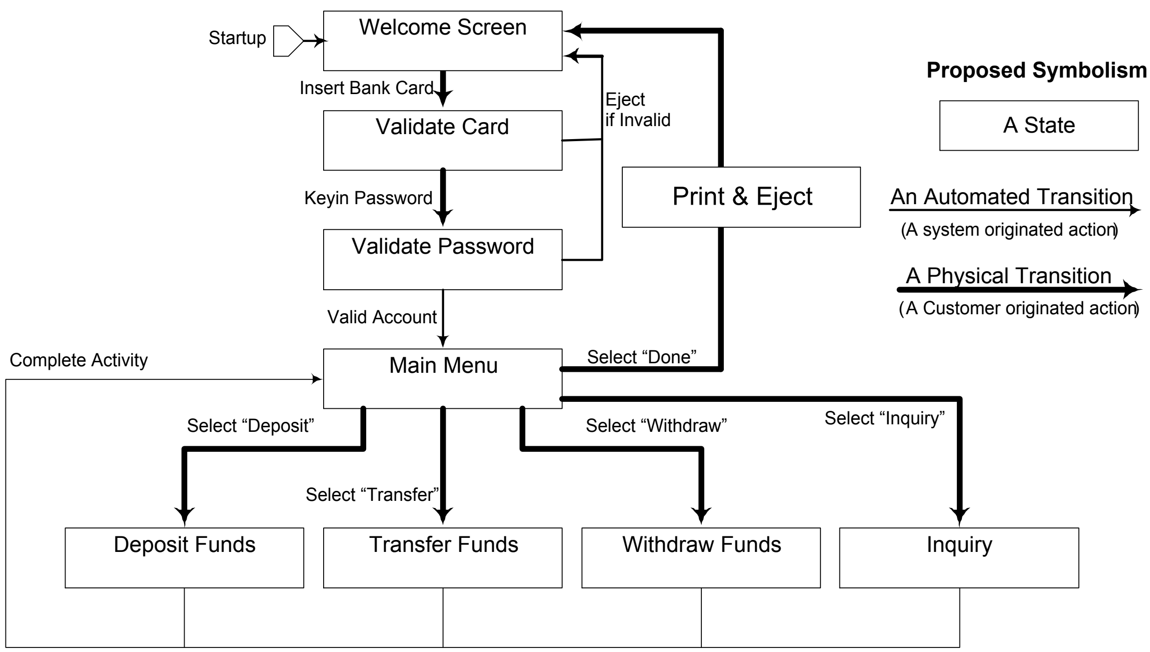

Figure 3b: Expanded ATM Process

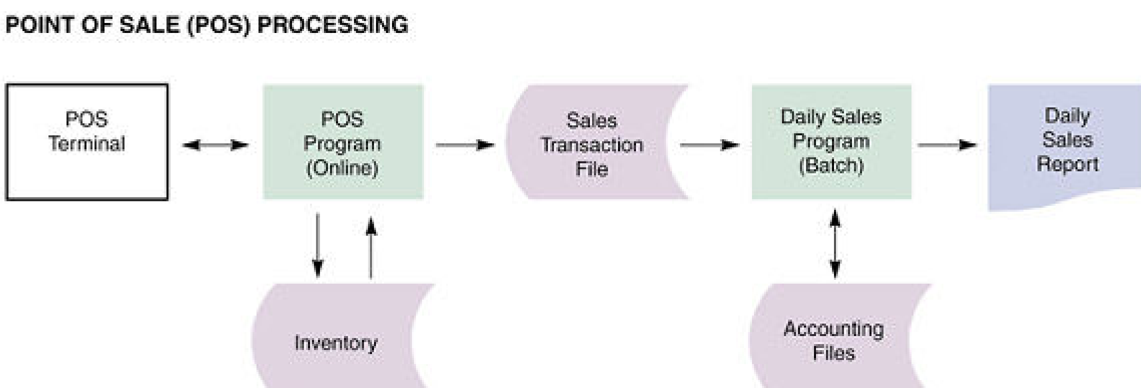

Figure 4a: Textbook Illustration of Combined Online and Batch

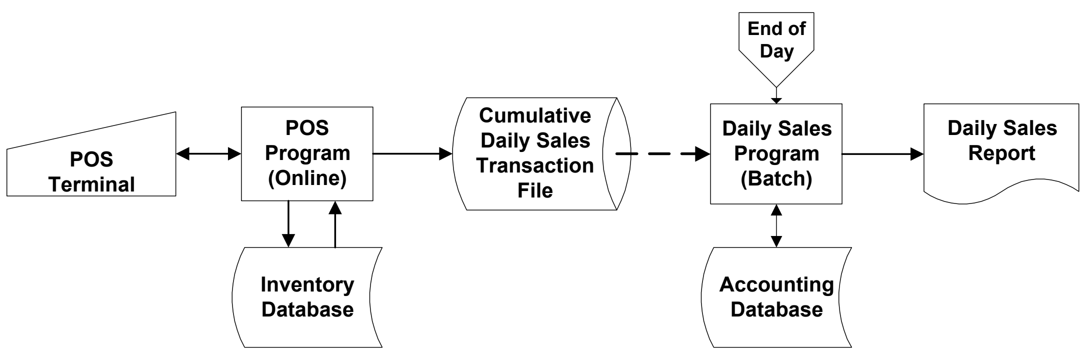

Figure 4b: Enhanced Illustration of Combined Online and Batch

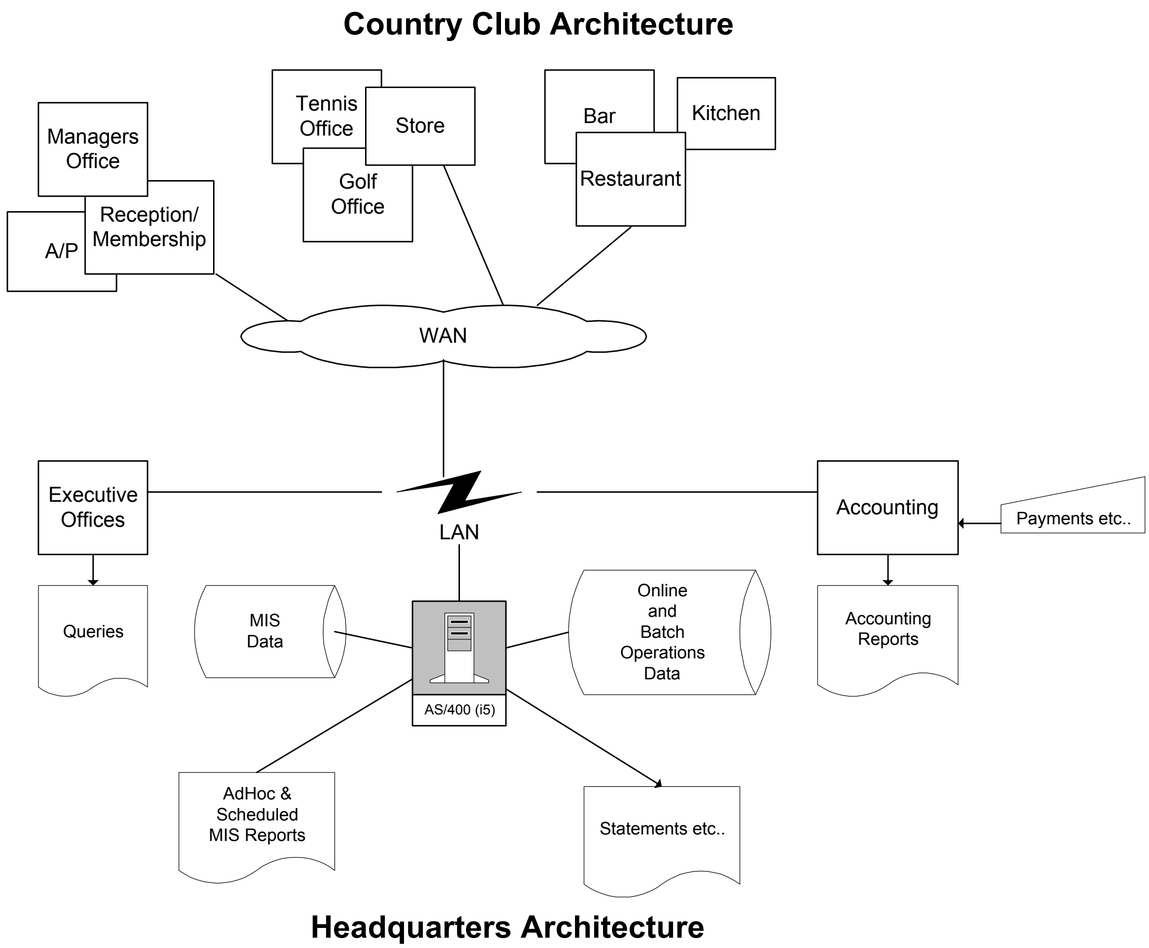

Figure 5: Country Club Billing Application Design