E-Deposit in Academic Use

Marjan Gusev, Ljupco N. Antovski and Vangel V. Ajanovski

Institute of Informatics, Faculty of Natural Sciences and Mathematics, St. Cyril and Methodius University, Skopje, Macedonia

The e-Deposit is a deposit that can be managed in electronic form. The concept of a fund accessible for different financial transactions makes the e-Deposit appropriate for use at universities. The application is implemented in three-tier architecture. Because of the extensive exchange of financial data over the internet, the data integrity is secured. The business and data access tier are implemented in a modular manner helping the robustness of the application and reducing the risk of unwanted behavior. Spe cial modules are used, that enable the integra tion of an e-commerce scenario in an already ex isting university information system.

Keywords: e-Deposit, SSL, PKI, e-commerce, e-business, e-university, certificates, encryption, XML, database.

1. Introduction

The implementation of advanced security and non-repudiation algorithms for user's authentication on internet has caused the e-banking concept to start moving forward. One part of that concept is the e-Deposit.

Deposit account is a demand, time, savings, and passbook or similar account maintained with a bank, savings and loan association, credit union or like organization, other than an account evidenced by a certificate of deposit.

Many business companies sell prepaid services such as the telecom and mobile operators [5].

The e-Deposit is a deposit that can be managed in electronic form. It means that the access to financial funds is granted without physical authentication of the deposit loan owner in the organization that maintains the deposit. The user authenticates himself/herself by established security procedures over an insecure internet connection.

The e-Deposit is frequently used in everyday financial transactions. For example, the funds in the e-Deposit are used for distributed money transfer over the internet by e-trading, e-betting or e-auctions [5]. The e-Deposit is a guarantee for the financial liability of the deposit owner.

2. University Possibilities

The concept of a fund that can be accessed for different financial transactions makes the e-Deposit appropriate for use at universities. During their studies, students pay for different services offered by the university campus.

The question of effectiveness is especially expressed in last minute actions. With deposited prepaid finances, it takes just a button click to make a financial transaction can instead of bank transactions.

The e-Deposit payment can be used in different areas of students' life. The service offered by this approach implies administrative taxes like exam file, different certificates, semester scholarship, cantina, library, laboratory use and other.

It is clear that in this case, there are no transactions from one account to another, but the money is transferred to a specific account only at the beginning. Afterwards, every payment is recorded and the state of the deposit is updated. When the deposit is consumed, the student is verified to improve the fund in positive manner in order to be liable for further payments in the university campus.

3. Use-Case Scenarios

There are two major use-case scenarios. The first one is the student's one. The user interface offers different services. The use-case for a student is explained in the following steps:

- The student gets connected to the internet.
- He/She establishes a request for secure data transmission from the web server.
- The server authenticates itself.
- The user authenticates himself/herself and agrees on the algorithms used for further data protection.
- The user chooses the service and requests a transaction of funds from his/hers deposit account for the specific service.
- The server acknowledges the transaction if there are sufficient funds, or returns a warning message.

The second use-case is for the administrators (teachers, university financial authorities). The procedure follows:

- The administrator user gets connected to the internet.
- He/She establishes request for secure data transmission from the web server.
- The server authenticates itself.
- The user authenticates himself/herself and agrees on the algorithms used for further data protection.
- The user chooses the requested query and requests a return view from the server.
- The server acknowledges and returns the requested view.
- Optionally, the administrator can send an update for a specific student fund.
- The server acknowledges.

4. Implementation

4.1. Three-Tier Architecture

To maximize the functionality and security, the application is implemented in three-tier architecture [1,10,11]. The three-layer architecture

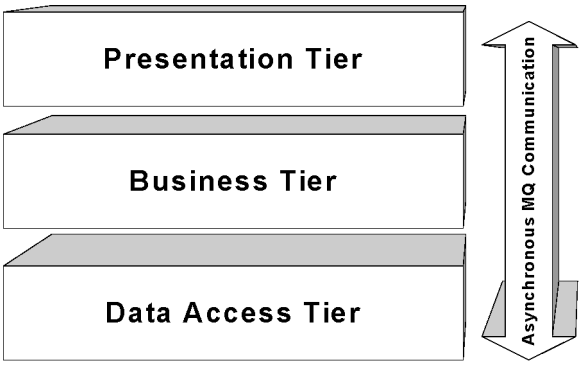

Fig. 1. Three-Tier Architecture.

given in Fig. 1 consists of the following components:

- Presentation Layer,
- Business Layer,
- Data Layer.

The presentation layer provides an interface to the end user into the services of the e-Deposit portal. This layer only encapsulates presentation of the information, but not the business logic [7]. The information received from the business tier is transformed in HTML format and presented to the user on the client's browser. All demands from the client are sent to the business tier for processing.

The business logic is processed in the business layer. All the elements of business logic, the rules and calculations are placed in this tier [2]. Every user authentication, transaction demand and verification is executed in this layer. It is a virtual interface among the presentation and the data layer.

The data layer implements a stateless object with generic procedures for connection with the physical database.

Because the processes in the e-Deposit environment are highly asynchronous, considering the distributed client, the business calculus and database communication, a system with asynchronous message queuing is introduced. This system incorporates three public message queues, each for every tier. The tiers communicate with each other in a hierarchical way with short predefined messages.

The messages are XML – (Extendable Markup Language) based. The main idea is to transfer information separated from presentation. This type of communication facilitates the burden on the business and data tiers and enables transformation at the presentation layer not only in HTML format, but also in whatever compatible format with the channel of communication in use.

With the use of message queuing and XML, the system event log enables tracking, diagnosing and recovery of every transaction put in question.

4.2. Network Implementation

The network implementation consists of the following items:

- Client computer,
- Web Server,
- Business Server,
- Database server.

The whole network implementation is given in Fig.2. The network is separated in two parts:

- Front End – the WEB server communicates with the client computer through SSL (Secure Socket Layer) internet.
- Back End – The Business and Database server protected from outside in private network accessed only by the WEB server in a restricted manner.

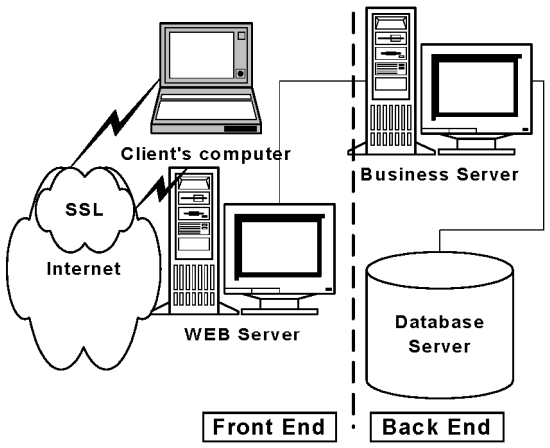

Fig. 2. Network Implementation.

Connection among the servers is secured and access is granted to a restricted list of users. Communication is available only between the client and the web server, the web server and the business server, the business server and the data server. It is asynchronous and through a system of message queuing. In the final version, the web and the business server are implemented on a single machine.

5. Security

Because there is an extensive exchange of financial data over the internet, integrity of the data must be secured. The employed security in the e-deposit solution is:

- Secure Socket Layer (SSL),
- Public Key Infrastructure (PKI),
- Firewall Security.

The Secure WEB Server uses the SSL protocol to create an encrypted communications channel between the client and server on the transport layer. SSL is a generic "pipeline" that secures the data [4].

Where non-repudiation is a key factor, on the application layer the Public Key Infrastructure (PKI) is introduced. PKI comprehensively satisfies the security requirements of e-Deposit.

Certification Authority (CA) is a certification-service-provider, which issues public key certificates [9]. The given university comes as a Certificate Authority. It issues certificates and confirms identity of the distributed users. The certification authority's key is used to sign the certificates and it is identified in the certificate as the issuer. Those certificates are used in the PKI infrastructure for the security procedures.

The PKI implementation covers the following aspects of secure transactions:

- Authentication,
- Confidentiality,
- Integrity,
- Non-Repudiation.

As a compliant security model, PKI provides establishment of a Trust chain, valuable in financial transactions [3,6].

The identity validation is established through various methods of identity check. The methods implemented are:

- User name and password validated on client's side with the use of the login media which encapsulates encrypted user information,
- Cookies,
- Digital Certificates stored on login media (smart cards or mini CD-s).

The installed firewall provides a high level of state-full security between the front-end server and the back-end database and business server. Specific policies are installed only to allow restricted communication.

6. Application and database structure

6.1. Application system structure

The overall e-Deposit application design consists of five independent functional units incorporating most of the application logic. All of these units work concurrently, while some of them are running all the time, and some only when triggered by special events.

The five main functional units, bearers of the application logic and design, are:

- Deposit management,
- Online shop,
- Request processing system,
- Service broker and
- Delivery system.

The role each of these functional units performs in the application system will be described in more detail in the following few paragraphs.

The **Deposit management system** takes care of the internal e-Deposit account of the clients (students) and of the payment processing. Its operation is mainly based on processing two payment transaction queues:

- e-Deposit Incomes and
- e-Deposit Expenses.

The Incomes queue is used for registering bank transactions regarding the transfer of finances from students' bank accounts to their University e-Deposit account. The Expenses queue keeps record of the transactions for University Services payments.

The **Online shop** is a web application style offering products for sale. This is done via a classical interface that features notions like "shopping cart" and "checkout". There is a list of products shown over different product categories. The student chooses a product to put into a shopping cart, with the possibility of returning the product in case he/she changes the mind. When the student has choosen all the products for buying, he/she proceeds to the checkout point. At the checkout, there is the possibility to authenticate and to buy the products provided the student's e-Deposit account has positive balance.

The cross-functional process style diagram depicted in Fig. 3, shows the division of the process activities and the functional units responsible for those activities. Each of the elements (rectangles, etc.) represents some activities to be performed by the corresponding functional unit, and the arrows define the order in which those activities should be performed. This flow diagram describes the conceptual level abstraction, instead of the lowest application level.

The main difference between this system and regular online shops is that products offered in the online shop are in fact University Services. For example, such services can be:

- registering for courses and exams,
- yearly enrollment for studies,
- library or lab usage,
- getting a course certificate or even receiving the final diploma.

Another difference compared to the way regular online shops work is the checkout process. In this application system, during the checkout process, terms of payment and shipping are not specified. This is the e-Deposit concept, since the funds are already transferred inside the institution.

Besides this, the "products" the money is paid for, are all in fact activities performed entirely

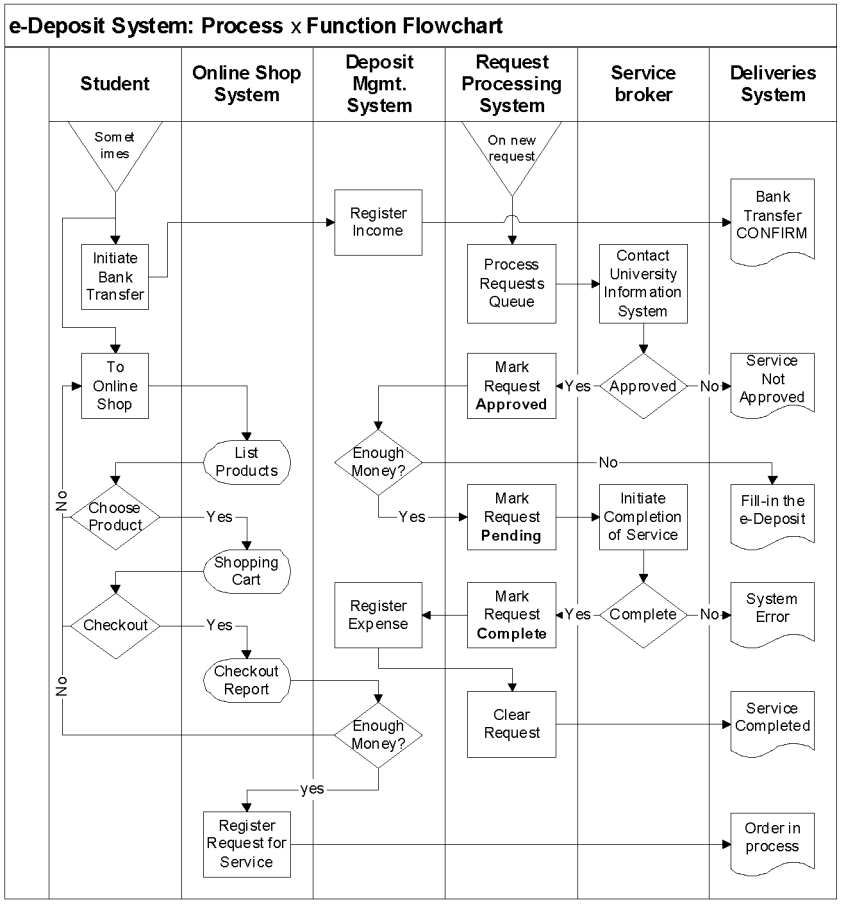

Fig. 3. Process and function diagram.

in the electronic domain for administrative purposes or e-Administration on a way to paperless electronic society.

The result of the student shopping process is a queue of requests for services processed by the **Request processing system**. Successful results are queued towards the Delivery system which informs the user when the service is completed and instructs the Deposit management to deduct the amount spent from the e-Deposit account.

Most universities are already using some kind of information system to maintain internal records. Such information systems usually work better if left untouched – so the only solution for collaboration of these two essentially different parts is via some kind of an interfacing system that transfers information back and forth the university information system.

The name of this subsystem is **Service broker**, and its main task is to get the operational status and results (reports, forms and documents) from internal components of the university information system. This is accomplished in concordance to the service codes and clients' authentication and authorization.

Another important task for the Service Broker is the initiation of all necessary actions that internal functional units should perform in order to produce the requested results. An example of operational status is the eligibility of a client for the listed service.

During shopping, the student accumulates a list of requests for service for which he/she is preliminary considered eligible (in a later process this fact is reassured). When moving products to the shopping cart, the request processing system processes each record in the list in a sequential manner (one record at a time) and in the exact order as it was specified inside the online shop. Failing to process the requests in such way can threaten the integrity and consistence of the data and data structures and could

potentially lead the student into an illegal status – depending on the chosen services.

For example, all the students can sign out of the university at any given moment. Most of the students are also eligible to register for different courses and exams. However, if one student makes a request consisting of these two activities in a sequence – register for an exam and sign out, it would be wrong to do them in the reverse way. Approving or denying the requests as a package, or establishing dependencies between them in order to find the right execution order is generally hard to accomplish, and out-of-order execution may lead to unwanted results.

The only real solution to this problem would be the introduction of a special status attribute for all requests. This attribute would state either “Approved”, “Pending” or “Completed” in correspondence with the status of each request. Processing of the requests should be done according to the rule – *do* the next request only if all the previous ones are marked as “Completed”.

Once a request is marked “Completed” it means that the service has finished and that the results are sent to the client (whether successful or not successful). The "Completed" status mark also means that the finances required for the service completion are subtracted from the e-Deposit account balance, in the amount specified by the university’s regulations. This mark also means that the request records will be cleared out of the Request for service queue once all of the requests in the package have been completed successfully.

After all these necessary steps, control over the information is transferred to the Deliveries subsystem - which acquires the results from the university information system and hands them over to the student. At this moment, the results are presented as internet documents. However, the possibility remains open for future expansions towards mobile or other wireless solutions.

6.2. Database schema design

The described application system structure manifests the need for a dedicated database management system in order to separate this application system structure and the complementary university information system. This implementation uses a relational database schema that is separate from the university information system database and thus highly scalable – as long as the interface to the university information system is maintained and managed to scale within the same factor.

All mentioned queues in the application design are implemented as database tables (as shown on the schema diagram on Fig. 4) – the list of offered Services, the Expenses and Incomes queue, as well as the Requests for Service queue – all implemented as separate independent database tables. Usage of special fields in the design enables distinction among the different states of activity of the queues.

Management of possible e-shopping requirements is the responsibility of the Online shop system. For this purpose, there is also the Shopping Cart table comprising different combinations of products a client has preselected for buying.

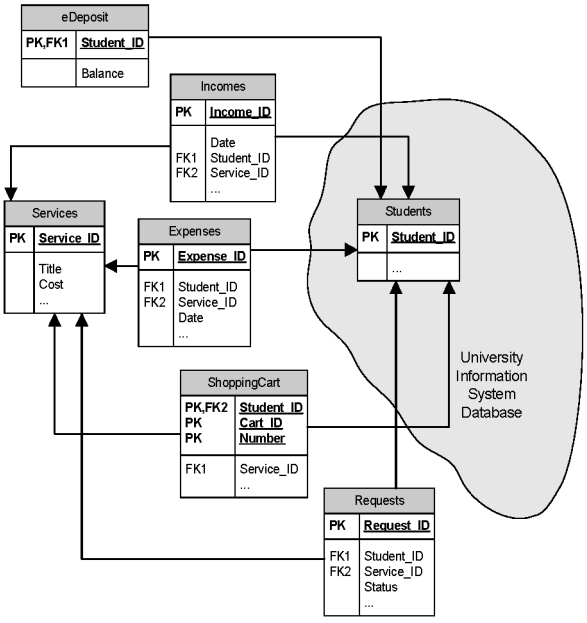

Fig. 4. e-Deposit relational schema.

The diagram presented in Fig.4 shows a conceptual level abstraction from the perspective of the application system logic, including the key parts of the e-Deposit relational database schema, but it is not a detailed relational database diagram.

The diagram in Fig.4 also shows parts of the e-Deposit system interaction with the university information system database schema.

7. Conclusion and Future Research

The e-Deposit is used in everyday financial transactions. The funds in the e-Deposit are used for distributed money transfer over the internet. This concept proves to be sustainable in the academic environment.

The main established goal is fast and secure transactions with prepaid service. This service takes an instance of time to transfer financial funds for any purpose in the university campus. This concept enables students to concentrate on the academic, not on the financial side of the university education.

With the procedures implemented in data manipulation, the system is designed to be data bullet proof. The three-tier architecture is upgradeable and scalable. The security measures undertaken make the system unbreakable in a lifetime.

This project is extendable to various means of communication, especially to mobile devices. Full integration in the e-commerce environment and enabling deposit payment for different kinds of goods and services that are not tightly connected to the university is a subject for further consideration.

References

[1] ANTOVSKI LJ., GUSEV M. Ebanking-Developing Future with Advanced Technologies. *In: Proc. of 2nd Conf. on Informatics and IT*; 20–23 December 2001; Bitola, Macedonia.

[2] EDWARD T. Transactional COM+− Building Scalable Applications. London; Addison- Wesley; 2000.

[3] ELISON C. AND SCHNEIER B. Ten Risks of PKI: What You Are Not Being Told About Public Key Infrastructure. *Computer Security Journal*, Vol.16, N.1, pp. 1–7.

[4] FREIER A., KARLTON P. The SSL 3.0 Protocol. Specification, Netscape Comm. Corp; 2001.

[5] GUSEV M. E-Commerce, a big step towards E-Business. *In: Proc. of 2nd SEETI Conf. on Trade Initiative and Commerce*; 8 November 2000; Skopje, Macedonia.

[6] HOWARD M. Designing Secure Web-Based Applications for Windows 2000. Microsoft Press, Washington.

[7] MICROSOFT & CISCO. E-Commerce Framework Architecture Document; White Paper; November 2000.

[8] MILOSEVIC A. Bank Information System- Bases, Architecture, Integration in Existing IS. In: *Mircetic M, editor. Proc. of New Banking Vision Conf*; Ohrid, 21–23 February 2001; Pexim, Skopje; pp. 61–67.

[9] RIVEST R., SHAMIR A. AND ADLEMAN L.A. Method for Obtaining Digital Signatures and Public-Key Cryptosystems. *Communications of the ACM*, February 1978; Vol. 21, pp. 120–126.

[10] Sun Microsystems. Scaling the N-Tier Architecture. White paper in Solaris Infrastructure Products and Architecture; December 2000; Sun, Palo Salto.

[11] TERPLAN K. OSS Essentials-Support System Solutions for Service Providers. February 2001; Wiley, Canada.

Received: June, 2002
Accepted: September, 2002

Contact address:
Marjan Gusev, Ljupco N. Antovski and Vangel V. Ajanovski
Institute of Informatics
Faculty of Natural Sciences and Mathematics
St. Cyril and Methodius University
Arhimedova b.b., PO Box 162
1000 Skopje, Macedonia
e-mail: marjan@ii.edu.mk
anto@ii.edu.mk
ajan@ii.edu.mk

MARJAN GUSEV was born in 1961. Since 1999 he is head of the Institute of Informatics at the Faculty of Natural Sciences and Mathematics, St. Cyril and Methodius University in Skopje, Macedonia, and manager of Wireless Application Laboratory. He completed his PhD studies in Loughborough, UK and Ljubljana, Slovenia in the field of parallel processing. He has published many papers in the field of parallel processing, e-business and mobile applications.

LJUPCO N. ANTOVSKI was born in 1977. Since 2001 he is with the Institute of Informatics at the Faculty of Natural Sciences and Mathematics, St. Cyril and Methodius University in Skopje, Macedonia. Currently he is a teaching/research assistant at the Institute of Informatics. He is a active member in the Wireless Application Laboratory. He graduated in electrical engineering at the Faculty of Electrical Engineering in 2001, major in computer science and automation. Mobile payments are the field of both his research and his graduate studies. He has published several papers concerning the aspects of m-payments, e-banking and e-commerce.

VANGEL V. AJANOVSKI was born 1975 in Skopje. He graduated from the Institute of Informatics, Faculty of Science and Mathematics, University St. Cyril and Methodius, Skopje, in 1999. Since 2000 he is attending graduate studies at the Institute of Informatics, in the area of artificial intelligence. The field of his master thesis is *Intelligent database and knowledge based systems in human-computer interaction*. He is currently working as a teaching and research assistant at the Institute of Informatics, on the courses such as "Databases", "Computer Graphics", "Operations Research", and other.