

**Tangible and Beneficial Course Objective
and Teaching Materials Proposed
for Management- and Information
Science-Oriented Students (Part II)**

能 登 宏

Khaled BUGRARA

Tangible and Beneficial Course Objective and Teaching Materials Proposed for Management- and Information Science-Oriented Students (Part II) ^{†)}

Hiroshi NOTO¹⁾ Khaled BUGRARA

Contents

1. Introduction
2. Basic Concepts
3. Course Objectives
4. Management of Class and Laboratory
5. Contents and Teaching Materials
6. Questionnaire for the Students' Feedback
7. Conclusion

[Abstract]

Innovative approaches are presented for teaching software application development to undergraduate students who major in Management and Information Sciences, and graduate students who major in Information Systems. The students should be able to deal with both business model construction and management information system development equally well. Selected for our courses is NetBeans IDE (Integrated Development Environment) on the Java platform for software development, with UML (Unified Modeling Language) plugged in for business modeling, in order to make full use of methodology in business modeling and in information system developing. The core of our innovative approach relies on "visualizing logical thinking" that designs abstractions and complexity. The visual practices and work of building and manipulating business models on such a platform help make the students interested in and feel challenged by software development. The curriculum contents and the teaching materials are set forth based on our basic concepts and methodology, along with examples of laboratory work in one graduate school course and in one undergraduate course.

5. Contents and Teaching Materials

In this section we present several examples of contents and teaching materials in our graduate school and our undergraduate school.

◇ Graduate Schools

Department of Information Systems, Graduate School of Engineering, Northeastern University.

For the sake of discussion, the following course contents relevant to our undergraduate courses are elaborated. This course covers how to leverage object-oriented techniques as well as user interface design principles to engineering multi-role business applications.

Key words : Business Model Construction, Management Information System Development, NetBeans IDE (Integrated Development Environment), UML (Unified Modeling Language), Logical Thinking Visualization

Students learn how to incorporate such features as configuration management, user administration, and role-based-access control.

Course: Application Engineering and Development (ISYG100)²⁾

Throughout the course dealt with are the items on the portal site for sales persons and sales managers in business as the examples of teaching materials and homework assignments for the students.

We remark here only the important and innovative points in our contents. By solving assignments one by one, the students understand and handle the concepts, methods and techniques on the Java NetBeans platform³⁾:

- Package, Class and Object, Instantiation, Event, Table, ArrayList, Hierarchical Structure (Package, Class), Generic Class
- Java/Swing, JFrame, JPanel, initComponents(), jTable, jScrollPane, Split Pane, ArrayList <ClassName>, CardLayout.

Shown in Table 5-1 are business related items and their realization in Java Programming or Object Model that comprise the main contents of their Lab practices.

Table 5-1. The correspondence between Business-related Items and their realization in Java Programming or Object Model

Business-related Items	Java Programming or Object Model
Catalog, Account	Directory
Multi Role	Categories
Business Model	Hierarchical Structure
Screen	Interface, Modularity
Navigation of Screens	Object Model, Relationship
Work Area	Event, Sequence of Screens
Role of Processes	Use Case
Real Time Update	Dynamical Plug-in

Summarized and shown in Table 5-2 are the complete contents, themes, concepts, methods, techniques and assignments of the teaching materials of our course.

Table 5-2. Contents of "Application Engineering and Development" course

Period	Theme	Concept	Notion / Subject	Method	Technique	Interface	Management	Information	Level (15)	Assignment
Lectures Practices										
Lab1	Demonstrate how to create Java/Swing application in NetBeans.	Java: Package, Class and Object. Instantiation, Event	Instantiation, Pass Object (data) between Class Object, Event Screen Change	Access Property of Variable, Class Constructor, Instantiation, Event Listener	Java/Swing JFrame, JPanel, initComponents () ⇐ visual components	Split Pane, Label, Button, Text Field	Product data ⇒ display	User Interface, Class and Object, Pass and get the data, Display	3	So many works are included in this Lab. They have to be carried out in several steps.
Lab2	Create and display many Products using jTable and ArrayList	Table, Hierarchical Structure (Package, Class), Generic class	Tabulated data, Update and display tabulated data. Screen Change (pass Catalogue).	Table, ArrayList Update data Reference Variable (Pointer), import	jTable, JScrollPane, ArrayList(Class Name), Screen Change. Enhanced for loop, Method managing Products	Table, Screen Change (pass Catalogue).	Tabulated Presentation, Product Catalogue, Manage Products Directory.	Table Object, Array/ List, Information Model (Relation)	4	So many works are included in this Lab. They have to be carried out in several steps.
Lab3	User Process Sequencing in Java/Swing.	Sequence of Screens	Work Area: Sequence of Screens, Search and Find in jTable.	Pass data between JPanels Product Catalogue ⇔ Product. Search and Find in jTable.	Create and Populate JPanels	Sequence of Screens, Search and Find in jTable.	Work Area, Work Responsibilities, Product Search and Product Manager Role Find in jTable	Product Catalogue, based on BaseProject teaching material	4	
Lab4	Implement Multi-Party Relationships.	Hierarchical Structure. Directory ⇒ Catalog	Hierarchical Structure ⇒ Business Model, Work Area	Event, Sequence of Screens	Split Pane, CardLayout	Work Area, Directory, Product Catalogue	Multi-Party Relationships, Business Model, Role of Players	Sequence of Screens, Event ⇒ pass Objects, Hierarchical Structure ⇒ Business Model.	4	based on BaseProject teaching material
Lab5	Application assigns Users. Business Model.	Business Model, Modularity, Personnel ⇔ Accounts, Categories (Roles)	Business Model, Information ⇒ Modules, Categories (Roles), Combo Box	Directories ⇒ Objects, Directory ⇒ ArrayList, Combo Box, add, get, display in Combo Box.	Hierarchical Packages, Directory Classes, User Screens, Static Method, toString (), if ()	Login Screen, Directory Screen, Combo Box	Users, Business Model, Categories (Roles)	Object-Oriented Program (Module), Pass Objects between Class ⇔ Class.	5	based on BaseProject teaching material. Several steps should be set forth.
Lab6	CASE Study (Order booking)	Screen Representation, Sales Console, Sales Model	Sales Process Use Case, Navigation Sequence, Sales Model creates Orders.	User Login, Manage Customer Screen, Serve Customer, Sales Order History, Sales Commission, Create Orders	Order Creation: created and kept in ArrayList.	Screen Representation, Sales Console	Sales Model, Sales Commission	Sales Process Use Case, Navigation Sequence, Manage Customer Screen, Manage ArrayList.	5	Several steps should be set forth + a student's own design
Final Project	Portal Sites for Sales Person & Manager. Takes Product Orders. Sales Commission calculations.	Product Orders, Booking Application, Sales Commission, Object Model using UML	Price Range Information, "Sales Commission Rule", Definition Screen, Object Model (UML), Screens logically connected, Generics	Multiple Product Catalogs, Negotiate prices, Take Product Orders, Sales Commission Rule, Object Model (UML), Screens logically connected, Generics	Realtime Information about the Commission, Commission Rules, Generics, Dynamical plug-in	Product Orders, Application "Sales Commission Rule" Definition Screen.	Negotiate prices, Multiple Product Catalogs, Sales Commission Rule.	Realtime "Sales Commission Rule", Definition Screen, Screen logically connected, Dynamical plug-in	5+	Comprehensive Assignment
Credit & Score										

◇ Undergraduate Schools

Department of Management and Information, School of Economics, Hokusei Gakuen University.

We are going to describe the course “Application II”/“seminar II” as a typical example of our Web application development. This course covers “Web application on Java Platform with Unified Modeling Language(UML)^{5,6)} plugged in”^{17,18)} which employs the Java EE (Enterprise Edition) category and the Enterprise Java Beans(EJB) technique with the JSF (Java Server Faces), JSP(Java Server Pages) Frameworks along with the MVC(Model-View-Controller) design pattern in its contents^{15,16,19)}.

Course: Application II (2014) / Seminar II (2014)

In this semester we set forth the “Bulletin Board System (BBS)” construction as its teaching materials in several steps each of which has its own theme based on the course objectives.

Throughout the course the emphasis is on the Object-Relational Mapping (ORM) that automates the mapping between entity (object) model and data model and the first workout of the web application system: basically creating several Java Server Pages (JSPs). Another emphasis is on the MVC framework or design pattern that provides clear separation of three artifacts, i.e. acting as data, displaying the data and manipulating the data and transferring control to the view. One deliverable of this approach is to transform the authentication function of BBS to the MVC framework with Java Beans structure. The students are gradually to understand and get used to what the Java web application looks like.

The table 5-3 shows the Lab assignments, the subjects or functions of BBS and their methods and techniques, and therein the complete contents, themes, concepts and problems are realized.

Shown below in Fig. 5-13 through Fig. 5-27 are the examples of the screen shots of Lab assignments of the course.

Running the BBS Web application with the JSF user interface, the application is deployed on the GlassFish server and the Initial Menu Screen appears in the default browser as shown in Fig. 5-13, where the index.xhtml is invoked which is written in facelets tags. On this screen we click on the [Sign-In] button to register ourselves with our login names and email addresses and the like if necessary.

Table 5-3. Correspondence between Lab assignments and the themes or functions of the BBS with their methods and techniques

Lab Assignments	Functions of BBS	Methods and Techniques
Lab 1	Creation of databases in MySQL / Java DB	MySQL / Java DB Server, SQL, JSTL Code, JDBC Technology, SQL Language
Lab 2	Initial Menu	Enterprise Application, JSF, Facelets, EJB Module, Web Archive (WAR) Module, Managed Bean, JSP, GlassFish Server
Lab 3	Send Email from BBS	jMail, JavaMail API, sendMail (), Simple Mail Transfer Protocol (SMTP)
Lab 4	User Account and Password	
Lab 5	User Login and User Authentication	interface HttpServletRequest to provide request interface HttpServletResponse to send a response
Lab 6	File Uploading	HTML Form, POST Method
Lab 7	Bulletin Board (Sending Messages)	JSF, Facelet, EJB Module, Entity Class, Message-Driven Bean, SessionFaçade, Web Module, Singleton Session Bean, Servlet, JMS Resources
Lab 8	Logout	
Lab 9	MVC Design and Java Beans	MVC Design Pattern

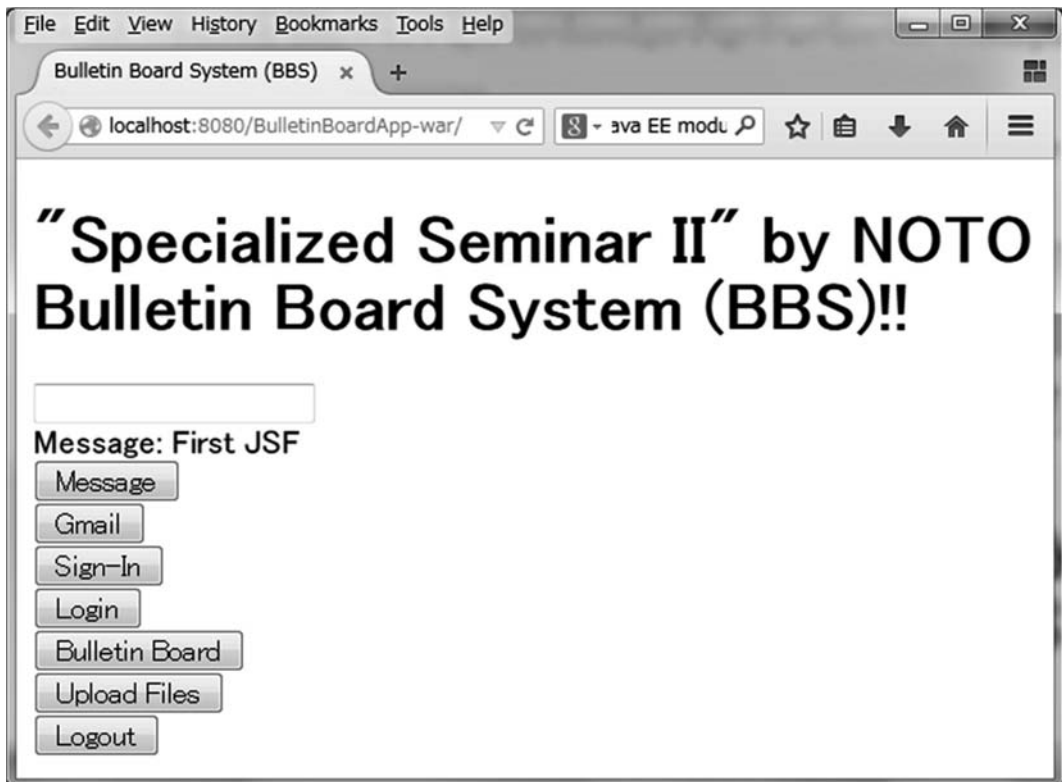


Fig. 5-13. Initial Screen of our BBS (Bulletin Board System)

The “Register User” page appears in Fig. 5-14 where one enters one’s Login Name, Nickname and Email Address and presses the [Register] button.

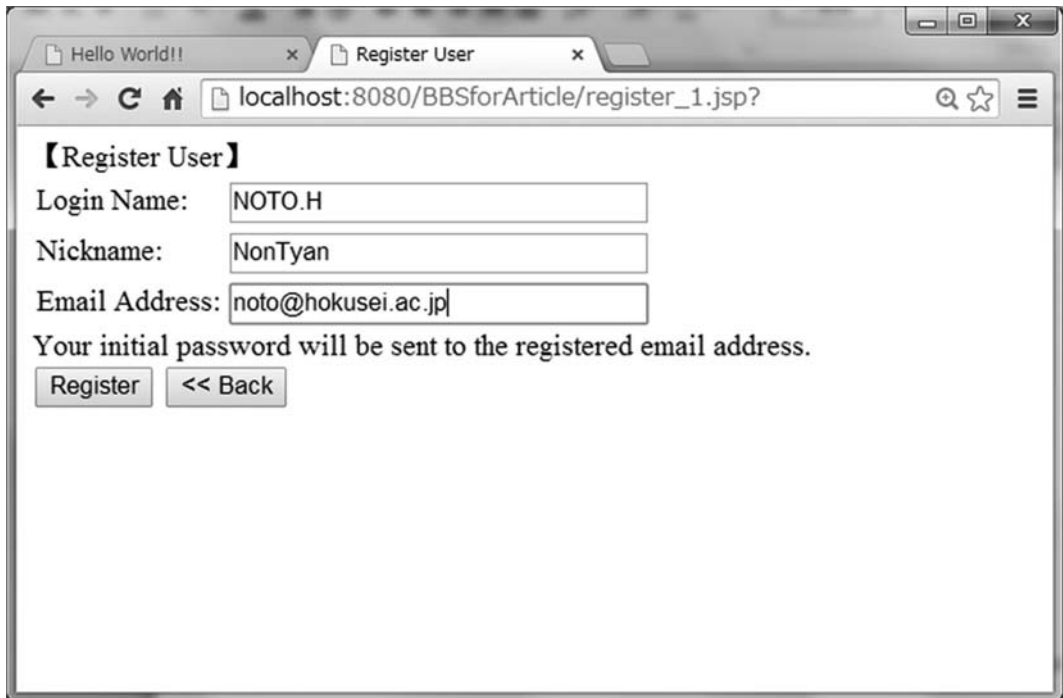


Fig. 5-14. User Registration by entering Login Name, Nickname and Email Address

Fig. 5-15 shows that the User Registration is completed: all the information is stored in the “user” table of the “mybbs” database on MySQL server (See Fig. 5-16.) and that the initial password is sent to the user by email. We need to supply our Google username and password as the administrator of the Google SMTP server. In our BBS we define a JavaBean called “SimpleEmail” which realizes the business model for the user authentication in our MVC architecture. The “register-2.jsp” page creates an instance of the JavaBean and sends the user’s initial password (See Fig. 5-15)

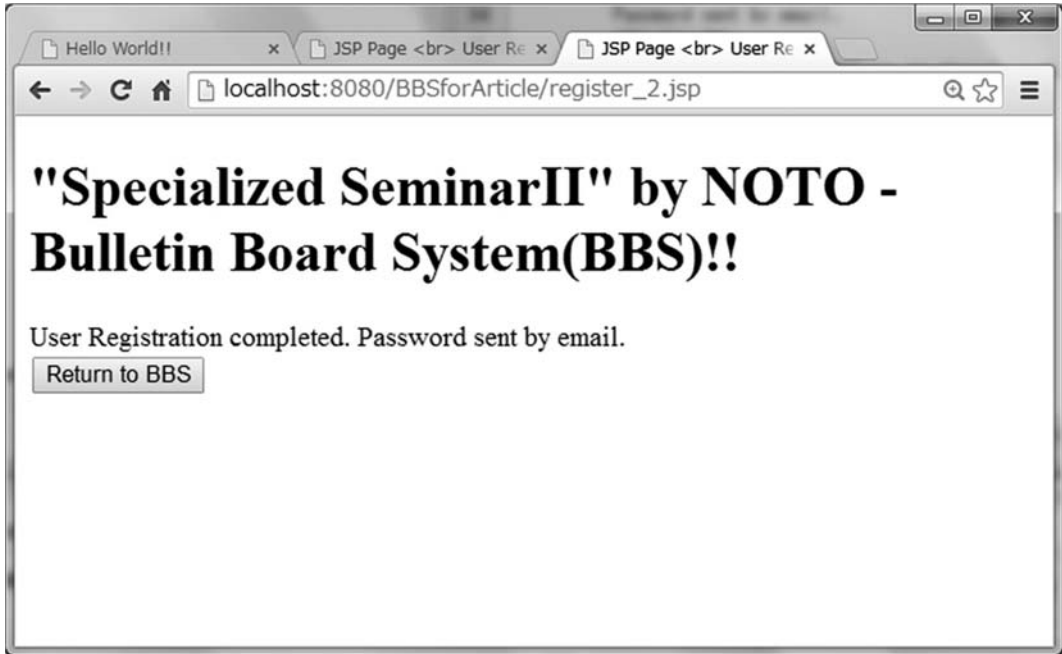


Fig. 5-15. User Registration is completed and the initial password is sent to the user.

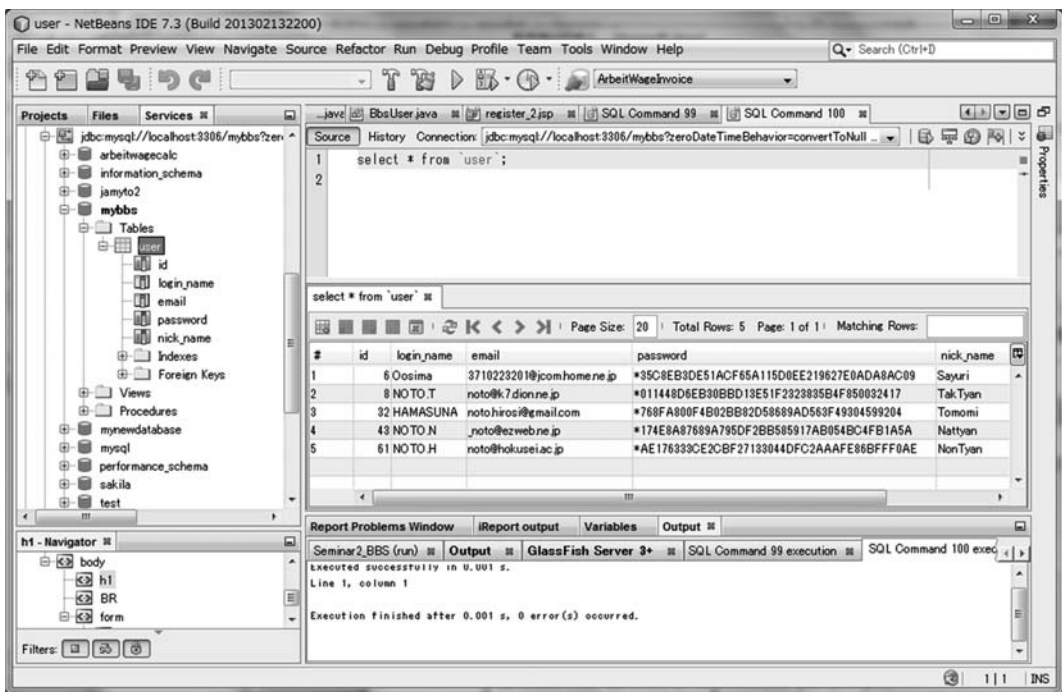


Fig. 5-16. "MyBBS" table stores all the data of user registration on MySQL server.

Clicking on [Return to BBS] button in Fig. 5-15 we reach again the previous BBS screen in Fig. 5-17, where we can login the BBS by clicking on the [Login] button.



Fig. 5-17. BBS Screen again, where we can login this time.

We are now on the login screen in Fig. 5-18 where we are supposed to enter the Login Name that we gave in Fig. 5-14 and the Password that was sent from the server.

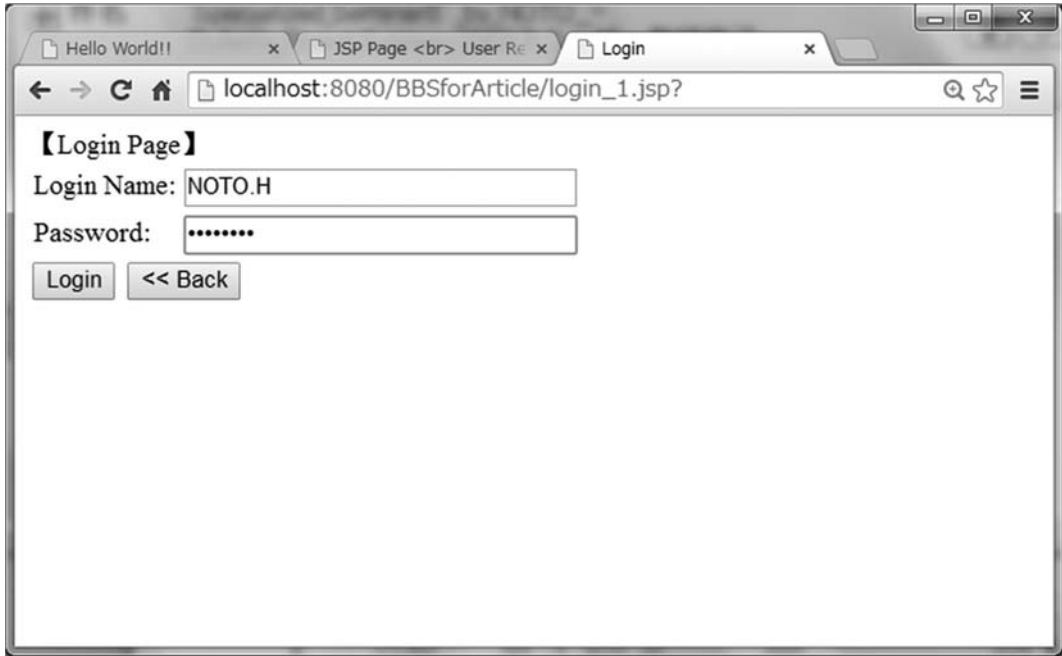


Fig. 5-18. User logs in with Login Name and Password sent from the server.

Fig. 5-19 shows that we succeeded in login: we are authenticated now.

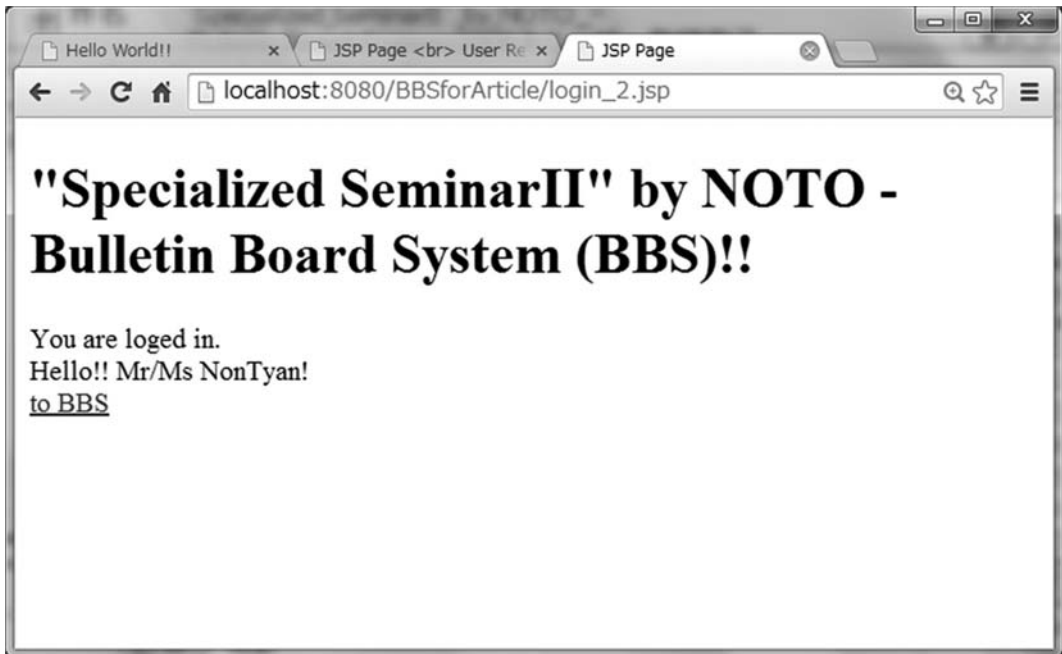


Fig. 5-19. User succeeded in logging in.

Clicking on the [to BBS] link in Fig. 5-19 we get back to the initial BBS Menu Screen

in Fig. 5-13.

Now let's try uploading files by clicking on the [Upload Files], which leads to Fig. 5-20, where we select action for [Upload File].



Fig. 5-20. BBS "File Upload" Screen

In our BBS the "Upload File" function consists of a servlet called "MyFileLoad.java" which invokes the `getPart()` method with the object that the "upload-1.jsp" page makes a file upload request for. The example has a very simple HTML form of [Choose File] button in Fig. 5-21 that enables us to browse the local file system to select a file, say, "C:\¥noto.txt".

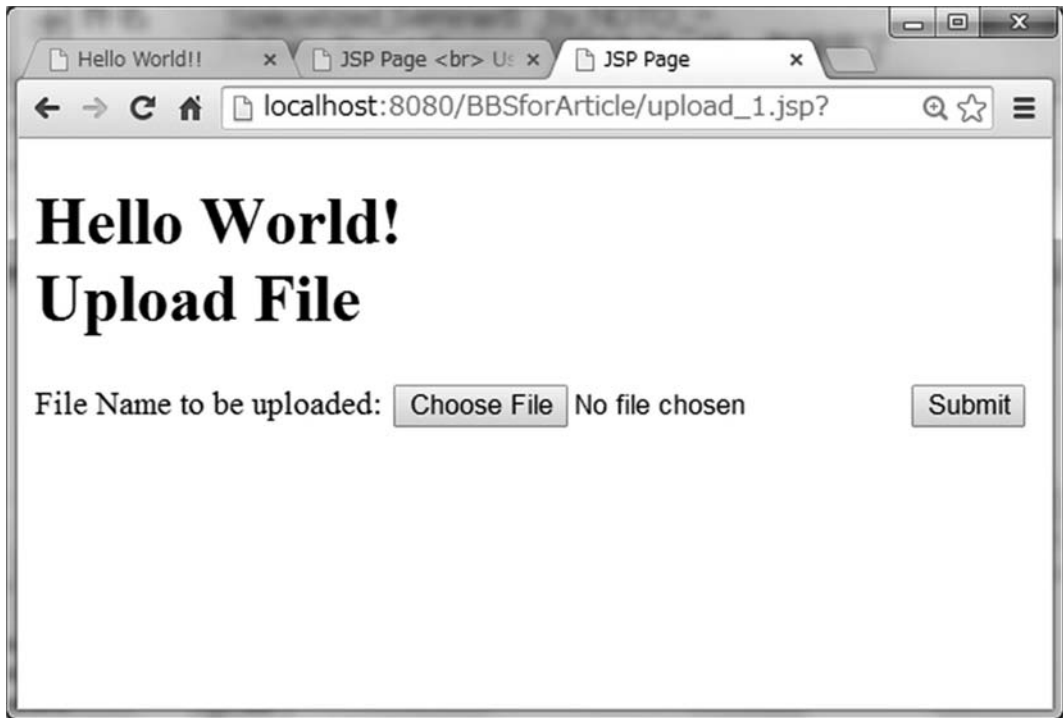


Fig. 5-21. File Uploading Screen appears.

When the file is selected, it is sent to the server as a part of a POST request by clicking on [Submit] button in Fig. 5-22.



Fig. 5-22. The selected file is about to be uploaded.

Fig. 5-23 shows that the selected file (“C:¥noto.txt”) is uploaded successfully, which is confirmed in Fig. 5-24 where the uploaded file is stored as “noto.txt” in the D:¥Drive on the server.



Fig. 5-23. The file is successfully uploaded.

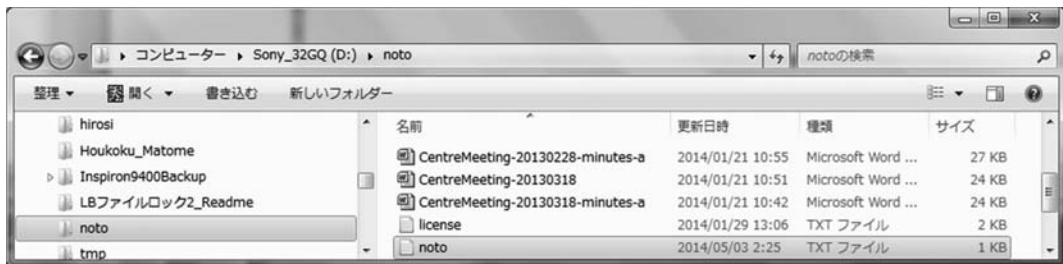


Fig. 5-24. The uploaded file is saved in the server folder “D:\noto.”

Getting back to the BBS Menu (Fig. 5-13) again by clicking on the [to BBS] link in Fig. 5-24, we click on the [Bulletin Board] button to send messages.²⁰⁾ The ListNews servlet opens the Initial Message Screen in Fig. 5-25 where a list of the messages and their contents in the database is displayed. We can click [Add New Message] to send a message with the PostMessage servlet.

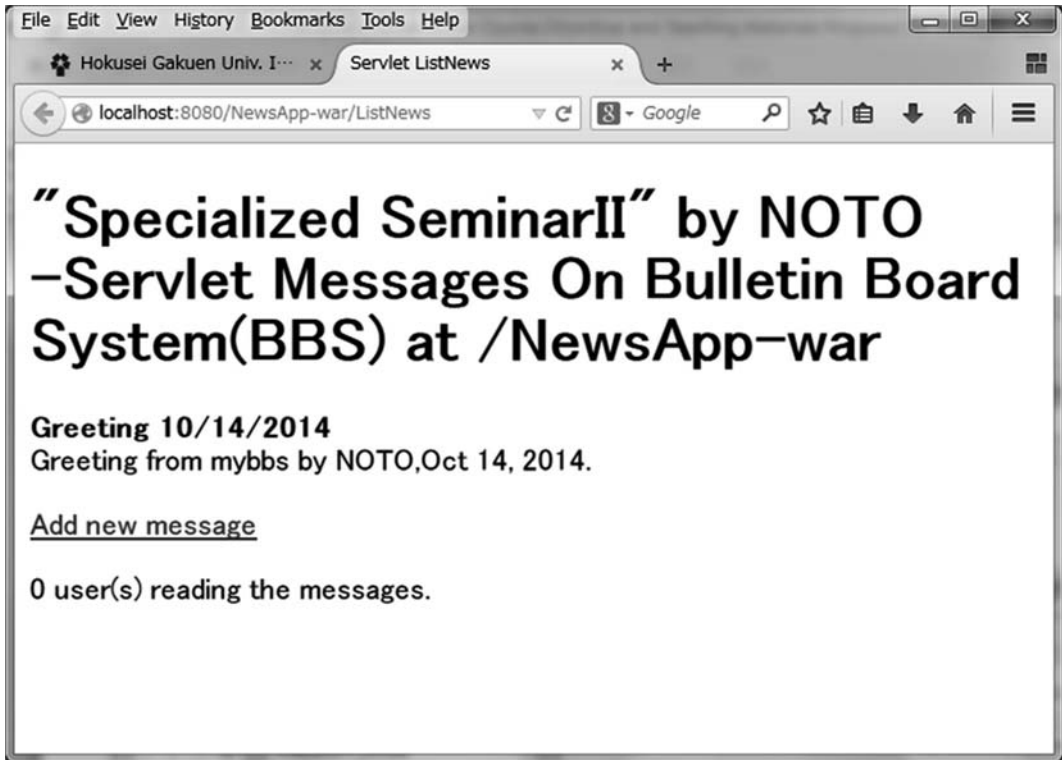


Fig. 5-25. The Initial Message Screen on the BBS system displays a list of the messages in the database "NEWSENTITY."

When we click on the [Submit Query] button in Fig. 5-26, the message is sent to the message-driven bean for writing to persistent storage and the message screen appears again in Fig. 5-27. The list of messages in the database retrieved by ListNews often might not yet contain the new message because our message service is asynchronous. In that case we can click on [Reload the current page] in the URL box in the menu bar.



Fig. 5-26. The [Submit Query] button sends a message to the message-driven bean.



Fig. 5-27. The list of messages in the database contains the new message.

Fig. 5-28 shows the records of the persistent storage, i.e. the table “NewsEntity” in the Java DB database “sample.”

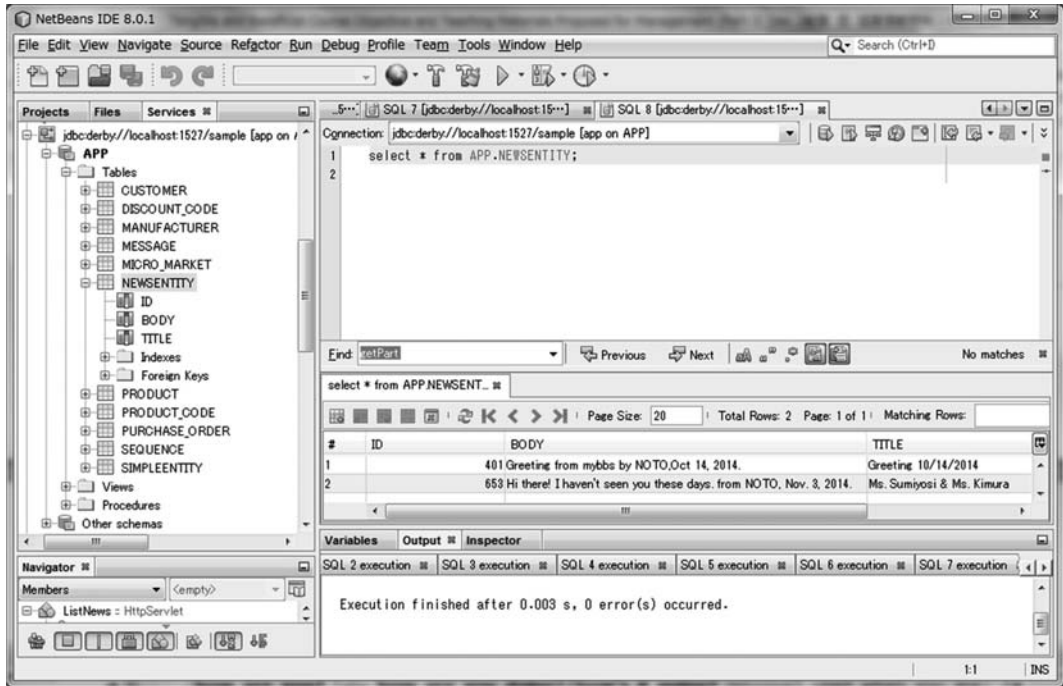


Fig. 5-28. Messages are stored in the table “NewsEntity” in the database JavaDB.

The last Lab Assignment (Lab 9) of the course is to transform the authentication function of BBS to the MVC framework with Java Beans structure. The clear separation of three artifacts, i.e. acting as data, displaying the data and manipulating the data and transferring control to the view is realized by transforming the authentication function of BBS to the MVC framework with Java Beans structure. The students are gradually to understand and get used to what the Java Web application looks like. The MVC framework is realized with a JSP page (View) and with a controller (Controller) and with a Java Beans (Model or Business Logic).

We show in Fig. 5-29 the conceptual framework of the “BulletinBoardApp” Web application with the Model-View-Controller [MVC] Pattern: Web Presentation tier, Web Control & Management tier, and Model & Business Logic tier and Database tier. The latter two tiers might well be called the Enterprise tier consisting of Enterprise Java Beans (EJB) and JPA. The Web application can perform the database-related operations supported by JPA using Session Bean as a Facade. Servlets, JSP, JSF are responsible for contents representation but do not have transaction management and/or persistent functionality. The Session Managed Bean is responsible for management in Web tier.

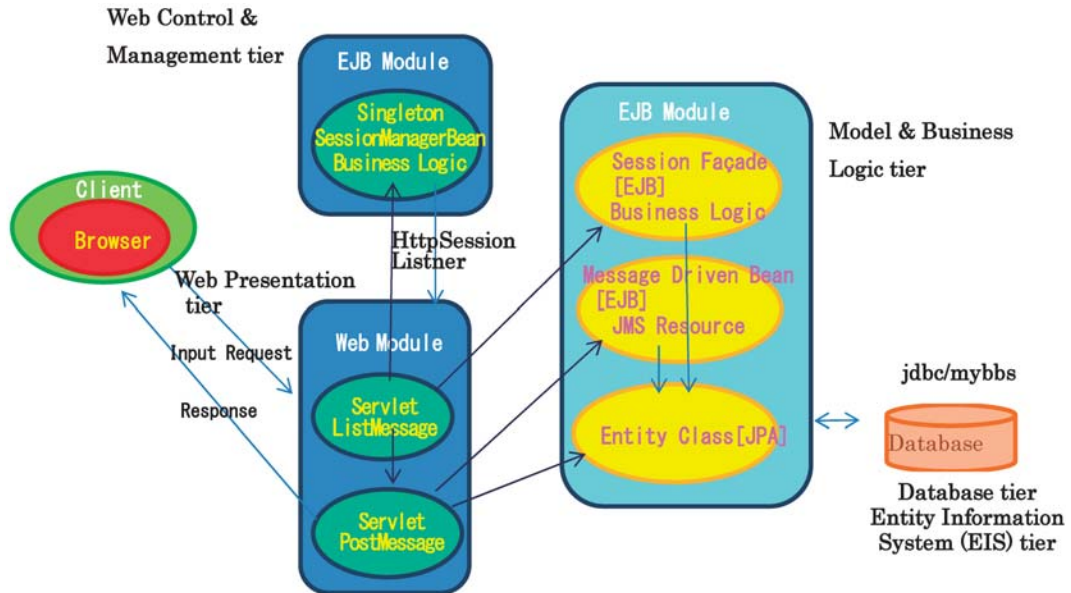


Fig. 5-29. The conceptual diagram of the “BulletinBoardApp” Web application with the Model-View-Controller [MVC] Pattern

6. Questionnaire for Students’ Feedback

While one of the authors (H. N) stayed in the Northeastern University, we planned to get feedback on the software development course in the graduate school. The questionnaire form had been prepared where the nine main categories are questioned that the student (respondent) has to answer in a set format. A distinction is made between open-ended and closed-ended questions. An open-ended question asks the respondent to write his/her own answer, whereas a closed-ended question has the respondent pick an answer from a given number of options. The response options for a closed-ended question should be exhaustive and/or mutually exclusive. Mostly five types of response scales for closed-ended questions are distinguished. Below is shown the questionnaire: I) Students’ attitude against the course, II) Contents of the course, III) Lectures, IV) Teaching Materials, V) Practices, VI) Subjects studied through the course, VII) Program Languages, VIII) NetBeans Platform and IX) General Evaluation of the Course (ISYG100, as of 2009).

I) Your attitude to the course

(Why you entered the Department of Information System, Graduate School of Engineering?)

Why you took the course ISYG100?) [selecting 2 at most out of 6]

- ① to be interested in software development
- ② to get knowledge of software development
- ③ to get Master’s Degree in Information Systems

IV) Teaching Materials

(quantity)

(5) too many (4) many (3) moderate (2) few (1) too few

(quality)

(5) excellent (4) good (3) moderate (2) fair (1) poor

V) Practices

(Lab, Assignment)

(5) too many (4) many (3) moderate (2) few (1) too few

(quality)

(5) excellent (4) good (3) moderate (2) fair (1) poor

(practicing hours at one time)

(5) too long (4) long (3) moderate (2) short (1) too short

(management of practicing session)

(5) excellent (4) good (3) moderate (2) fair (1) poor

(How many days did you spend solving one lab through Lab 1 to Lab 6?)

() days to () days

(How many weeks did you spend solving the final project?)

() weeks

(How often did you consult a TA?)

() times

(Is a TA helpful to you?)

(5) strongly agree (4) agree (3) moderate (2) disagree (1) strongly disagree

VI) Subjects

(the most interesting subject)

Select 3 items at most from [1-19: Java-related subjects] in the AC (Answer Choices) Table 6-1.

Select 2 items at most from [21-28: NetBeans-related subjects] in the AC Table 6-1.

Select 1 item from [31-36: Model-oriented subjects] in the AC Table 6-1.

Select 2 items at most from [41-50: Project-oriented subjects] in the AC Table 6-1.

(the most difficult subject)

Select 3 items at most from [1-19: Java-related subjects] in the AC (Answer Choices) Table 6-1.

Select 2 items at most from [21-28: NetBeans-related subjects] in the AC Table 6-1.

Select 1 item from [31-36: Model-oriented subjects] in the AC Table 6-1.

Select 2 items at most from [41-50: Project-oriented subjects] in the AC Table 6-1.

(the subject you think you could understand the most)

Select 2 items at most from [1-19: Java-related subjects] in the AC (Answer Choices) Table 6-1.

Select 1 item from [21-28: NetBeans-related subjects] in the AC Table 6-1.

Select 1 item from [31-36: Model-oriented subjects] in the AC Table 6-1.

Select 1 item from [41-50: Project-oriented subjects] in the AC Table 6-1.

(What subject inspired you the most)

Select 2 items at most from [1-19: Java-related subjects] in the AC (Answer Choices) Table 6-1.

Select 1 item from [21-28: NetBeans-related subjects] in the AC Table 6-1.

Select 1 item from [31-36: Model-oriented subjects] in the AC Table 6-1.

Select 1 item from [41-50: Project-oriented subjects] in the AC Table 6-1.

Ⅶ) Programming Languages

(the first time to learn Java [Check one])

yes no

(Java is easy to learn)

(5) strongly agree (4) agree (3) moderate (2) disagree (1) strongly disagree

(List up Programming Languages you have ever learned [Check all the items if any])

Java C (C++, C#) Pascal Delphi Basic VBA PHP

Ruby Perl Python Java Script COBOL FORTRAN

Assembler [others ()]

Ⅷ) NetBeans Platform

(operability)

(5) strongly agree (4) agree (3) moderate (2) disagree (1) strongly disagree

(functionality)

(5) strongly agree (4) agree (3) moderate (2) disagree (1) strongly disagree

(visual platform)

(5) strongly agree (4) agree (3) moderate (2) disagree (1) strongly disagree

(self-contained platform)

(5) strongly agree (4) agree (3) moderate (2) disagree (1) strongly disagree

(unsatisfactory points)

Description []

Ⅸ) General Evaluation of Course 'ISYG100'

(5) excellent (4) good (3) moderate (2) fair (1) poor

Thank you very much for your cooperation and patience!!!

Good luck!!

Table 6-1 Answer Choices for Query VI

1-19 : Java-related subjects

21-28 : NetBeans-related subjects

31-36 : Model-oriented subjects

41-50 : Project-oriented subjects

1	2	3	4	5	6	7	8	9	10
reference variable (pointer).	Class Object	instantiation	Event, Event listener.	Package import	Table Object Array List	directory	Combo box	Label, Button, Text Field.	enhanced for loop.
11	12	13	14	15	16	17	18	19	20
pass object (data) between class objects.	Access property of variable.	static method.	toString()	Generics	search & find in table.	Generic class.	Dynamical plug-in.	if()	
21	22	23	24	25	26	27	28	29	30
initComponents ()	JScrollPane	JFrame JPanel	Split Pane	CardLayout	JTable	Java/Swing	NetBeans		
31	32	33	34	35	36	37	38	39	40
Object-Oriented Program.	Information Model.	Business Model.	Object Model using UML.	Generic program.	Use Case				
41	42	43	44	45	46	47	48	49	50
Dynamically adjustable Dynamical plug-in.	modularity	Multi-Party relationships.	Sequence of Screens.	Multiple Product Catalogue.	Login Screen.	Update data	Categories (Roles)	Work Area	Sales Commission Rule.

7. Conclusion

In this article we have proposed preferable and beneficial curriculum programs and teaching materials relevant to such courses as management information systems, software development and application architecture for the management and information oriented students both in the graduate and undergraduate schools where the present authors have been offering several courses in charge.

First we have presented our basic way of thinking about what the software development courses should be like for the management and information oriented students. The core understanding of our students' status as their potential professional careers is that they are, mostly, somewhere in between information science specialists, software programmers and traditional clerical and managerial staffs or leaders.

One of the basic aims of this article is to figure out what kind of knowledge and skills our students are to acquire in graduate and undergraduate schools. The management oriented students should brush up their programming skills to make specification requirements ready and to construct their management information systems. The information science oriented students should learn what the business model is all about and realize their business model in their software development platform. It is just their in-between status and characteristics that facilitate the development of their capabilities in the real business world and the career pursuing that they are heading for.

We have just surveyed the current courses which have been going for several years and which the authors have been in charge of both in graduate schools and in

undergraduate schools. We have summarized the current graduate course of “Application Engineering and Development” in terms of several points of views in a tabular form. We have also reviewed five courses “Software Development I and II (2008, 2009-2013)”, “Application II (2009-2013)” and “Application II (2014)”/“Seminar II (2014)” of our undergraduate school.

Based on our basic concepts and methodology of our courses, we have set forth the curriculum contents and their teaching materials for one semester in the graduate school and for several semesters including one tangible example of laboratory works in the selected course in the undergraduate school. We take the Java development platform for granted, judging from the well accumulated teaching experience and teaching stuffs that have been working successfully in Application Engineering and Development courses in our graduate school.

Summarized below are the remarks we should make about the management of lecture classes and laboratory rooms both in graduate and undergraduate schools. All the remarks have been derived from our daily practices of giving lectures, helping students solve problems they encounter in the laboratory and communicating with them in various occasions. We ourselves should always keep in mind to work together with our students and to pursue more efficient and beneficial ways to enable the students to develop both their abilities and careers in competitive information management fields.

Information system programming requires logical thinking. And business model construction deals with the complexity pertaining to the organizational structures of the underlying business entity. Both works need a lot of abstract thinking and rigor of complex logic. The core of our innovative approaches, therefore, should facilitate abstract thinking and logical reasoning. We have relied on “visualizing logical thinking” that “designs abstract and complexity.” Design is about visualizing managing complexity. Examples of the teaching materials for abstract design have been presented: Catalog and Account, Screen design, Navigation of Screen, Business Model construction and MVC Design Patterns, and finally work on both UML and NetBeans Platforms. The work demands a lot of practices as well. It should indeed be in our software development courses that we convey the two ingredients: abstraction and practicality.

We have planned to get feedback on the software development course in the graduate school. The questionnaire forms have been prepared where nine large items are questioned to the students, mostly selecting an answer out of five answer choices: students’ attitude against the course, contents of the course, lectures, practices, teaching materials, subjects studied through the course, program languages, NetBeans platform and general evaluation of the course. In the present articles shown are only questionnaire form and items therein. The results and analyses of students’ feedback will be elaborated elsewhere.

Acknowledgement

We thank the staffs of the Information Systems Center of Hokusei Gakuen University

for constructing the computer and network environment that meets our requirement for carrying out the software development in our courses. Thanks are also due to the Visual Paradigm Company who provides the license to use the SDE EE and UML 10.3 for the UML tool. We are very much obliged to the NetBeans Community for our utilizing its free and user-friendly IDE. We would also like to appreciate the Department of Management and Information of our university for supporting us financially for getting the license of the activation the software of Visual Paradigm Company.

One of the authors (H.N.) would like to express his sincere thanks to Kal's help, generosity and hospitality while he stayed in the Northeastern University. He was greatly impressed by Kal's fine balance between knowledge, skills and practices.

References

- †) Hiroshi NOTO and Khaled BUGRARA. "Tangible and Beneficial Course Objective and Teaching Materials Proposed for Management- and Information Science-Oriented Students (Part I)", Hokusei Review, The School of Economics (Hokusei Gakuen University), Vol. 53, No. 2, March 2014.
- 1) One of the authors (H. N.) took sabbatical leave and stayed in Graduate School of Engineering, Northeastern University for half a year from 2008 through 2009, where he was allowed to join Professor Khaled BUGRARA laboratory and was able to have valuable experiences participating in a software development class in the graduate school. He could learn a lot about business modeling and visual abstraction.
 - 2) Kal Bugrara. Syllabus in Northeastern University-Graduate School of Engineering, "Information Systems Who We Are", 2008-2009.
 - 3) <http://netbeans.org/>
 - 4) Mary Campione and Kathy Walrath. The Java Tutorial Second Edition: Object-Oriented Programming for the Internet, Sun Microsystems, Addison Wesley, 1998.
Cay S. Horstmann, Gary Cornell. Core Java Volume I and II, eighth edition, Prentice Hall, 2008.
 - 5) <http://www.visual-paradigm.com/>
 - 6) Tom Pender. UML Bible, Wiley Publishing Inc., 2003.
 - 7) <http://dev.mysql.com/>
 - 8) <http://community.jaspersoft.com/project/ireport-designer>
 - 9) <http://docs.oracle.com/javaee/>
 - 10) <http://glassfish.java.net/>
 - 11) Heiko Boeck. The Definitive Guide to NetBeans Platform 7, Apress, 2012.
 - 12) <http://netbeans.org/kb/docs/web/mysql-webapp.html>
 - 13) <http://www.oracle.com/technetwork/java/index-jsp-135995.html>
 - 14) The Java EE 6 Tutorial: <http://docs.oracle.com/javaee/6/tutorial/doc/>.
 - 15) Yuli Vasiliev. Beginning Database-Driven Application Development in Java EE using GlassFish, Apress, 2008.
 - 16) David R. Heffelfinger. Java EE 6 Development with NetBeans 7, PACKT, 2011.
 - 17) NOTO Hiroshi. "A Java Application Development Platform with a Unified Modeling Language (UML) Plug-in (Part I)", Hokusei Review, The School of Economics (Hokusei Gakuen University), Vol. 52, No. 2, March 2013.
 - 18) NOTO Hiroshi. "A Java Application Development Platform with a Unified Modeling Language (UML) Plug-in (Part II)", Hokusei Review, The School of Economics (Hokusei Gakuen University), Vol. 53, No. 1, Sept. 2013.

- 19) Jonathan Wetherbee, Chirag Rathod, Raghu Kodali and Peter Zdrozny. Beginning EJB 3 Java EE 7 Edition, Apress, 2013.
- 20) We owe our basic architecture to the NetBeans tutorial “Creating an Enterprise Application with EJB3.1.”