THE DESIGN, IMPLEMENTATION, AND EVALUATION OF INCA: AN AUTOMATED SYSTEM FOR APPROVAL CODE ALLOCATION

A THESIS SUBMITTED TO THE GRADUATE DIVISION OF THE
UNIVERSITY OF HAWAI'I IN PARTIAL FULFILLMENT
OF THE REQUIREMENTS FOR THE DEGREE OF

MASTER OF SCIENCE

IN

INFORMATION AND COMPUTER SCIENCES

MAY 2002

By
Jitender Miglani

Thesis Committee:

Philip Johnson, Chairperson
Wesley Peterson
Chris Hundhausen

We certify that we have read this thesis and that, in our opinion, it is satisfactory in scope and quality as a thesis for the degree of Master of Science in Information and Computer Sciences.

THESIS COMMITTEE

____

Chairperson

____

____

To

CSDL

ACKNOWLEDGEMENTS

First of all I would like to thank Dr. Philip Johnson. This thesis would not have been possible without his guidance. He is not only my thesis advisor but also my mentor. It has been a great learning experience for me while working with him for almost two years. He enlightened me time-to-time with his views on Software Engineering. All those discussions have helped me in becoming a better software engineer.

I would also like to acknowledge past and present CSDL members. I have had great discussions with them regarding software engineering. In chronological order: William Albritton, Joy Agustin, Weifeng Miao, and Mike Staver.

Next I would like to acknowledge the entire INCA developer team. I would like to acknowledge all the ICS613 students who participated in the implementation of INCA during first two weeks. Then, the five students (other than myself) who volunteered to finish INCA. Those students are: David Liang, Weifeng Miao, Paula Nishida, Xiahong Yin, and Shenyan Zhen.

Then I would like to acknowledge Lei Wakayama, the ICS course administrator, who patiently used our system and kept encouraging us from time to time by saying that, “Its great. It is doing well.”

I would also like to thank all the ICS students who participated in INCA user survey and provided useful feedback for the improvements in INCA.

Finally I would like to thank my committee members Dr. Chris Hundhausen and Dr. Wes Peterson for taking the time to read this thesis and provide useful comments.

ABSTRACT

The ICS department of the University of Hawaii has faced problems surrounding approval code distribution as its enrollment has increased. The manual system for approval code allocation was time-consuming, ineffective and inefficient. INCA is designed to automate the task of approval code allocation, improve the quality of course approval decisions, and decrease the administrative overhead involved in those decisions.

Based upon informal feedback from department administrators, it appears that INCA reduces their overhead and makes their life easier. What are the old problems that are solved by INCA? Does INCA introduce new kinds of problems for the administrator? What about the students? Are they completely satisfied with the system? In what ways does the system benefit the department as a whole?

This thesis discusses the design, implementation and evaluation of INCA. It evaluates INCA from the viewpoint of the administrator, the students, and the department. An analysis of emails received at uhmics@hawaii.edu account indicates that INCA reduces administrative overhead. The results of the user survey show that three quarters of students believe INCA improved their course approval predictability and course requirements understandability. They prefer INCA to old method of requesting approval codes by email. INCA database analysis provided course demand information and student statistics useful for departments. This evaluation of INCA from three different perspectives provides useful insights for future improvement of INCA and for improving the student experience with academic systems in general.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

LIST OF ABBREVIATIONS

EIS	Enterprise Information Systems
EJB	Enterprise Java Beans
FCFS	First Come First Serve
ICS	Information and Computer Sciences
ITS	Information and Technology Services
INCA	I Need Course Approval
ISIS	Integrated Student Information System
J2EE	Java 2 Enterprise Edition
JSP	Java Server Pages
PA’E	Phone Assisted Enrollment
SIS	Student Information Systems
UH	University of Hawaii
US	United States
uhunix	University of Hawaii Unix systems
uhmics	University of Hawaii at Manoa Information and Computer Sciences uhunix account

Chapter 1. Introduction

1.1 The course allocation problem

Universities in United States use “student information systems” (SIS) to manage student records. These systems manage information on student admissions, student academic activities, student financials, and course enrollment.

The University of Hawaii (UH) uses ISIS (Integrated Student Information System) to manage student and course information. ISIS manages admissions information, student biographic information and student academic histories. It also manages the course catalog, schedule of classes and course enrollment. In Fall 1986, ISIS was used to carry out the registration for the first time. In Fall 1996, ISIS was interfaced with the PA’E phone registration module. PA'E (Phone-Assisted Enrollment) is a computer system that can be accessed by either a touchtone phone or a personal computer connected to the Internet. PA’E allows students to view or listen to course schedules and register for classes.

PA’E uses a single priority rule - seniority - to allocate seats in a class. According to this priority rule, seniors get to register before juniors, juniors before sophomores, and so on. However, some departments need to apply their own rules and eligibility criterions before students can be offered seat in a course. These departments need to take control away from PA’E for the purposes of deciding who will receive seats in their courses.

Various departments within the College of Arts and Sciences, the College of Business, the College of Education, the College of Engineering, and the School of Travel Industry Management offer classes with restricted enrollment. If a student wishes to enroll in courses offered by these departments, the student must obtain the approval of an authorized representative of the department or college offering the course. Once approved, the student is given a “phone approval code” to register over PA’E for the restricted classes.

1.2 Issues with approval code allocation in the ICS department

The ICS (Information and Computer Sciences) department at UH is an interesting example of a department with course allocation problems. Each semester, over 700 ICS majors and graduate students, along with hundreds of non-ICS majors and unclassified graduate students, compete for seats in the dozen or so ICS classes offered. Traditionally, approval code allocation was manual. To allocate approval codes to the students, an administrator from the department received requests for approval codes from students via email, phone or in-person. Then, the ICS administrator granted or denied the requests based upon the seats available and the student’s background. This manual approach to approval code allocation created the following problems:

1. **There is administrative overhead involved in reading and responding to thousands of emails**. Every semester, the administrator needs to respond to thousands of emails from students. The emails usually contain course requests and queries regarding courses offered and student’s qualifications. For every course request, the administrator checks student records and seats available in a course and accordingly accepts or denies a course request. If the request is denied, the administrator provides an appropriate explanation.
2. **Students miss their graduation dates if they don’t get necessary courses on time.** The biggest concern among the students is that they won’t graduate on time if they

don’t get necessary courses. They write emails to the administrator and the department to complain if they don’t get approval codes on time.

3. **There is no concrete data available to the department regarding courses in demand**. With thousands of requests to process, it is no longer clear whether the best possible decisions are being made on who should be admitted to the program. Furthermore, the requests themselves provide interesting, unexploited information about the kind of student interest in program and the kinds of classes they would like. Unfortunately, the overhead involved in this kind of analysis is too high to perform for each class and each semester using the manual system.

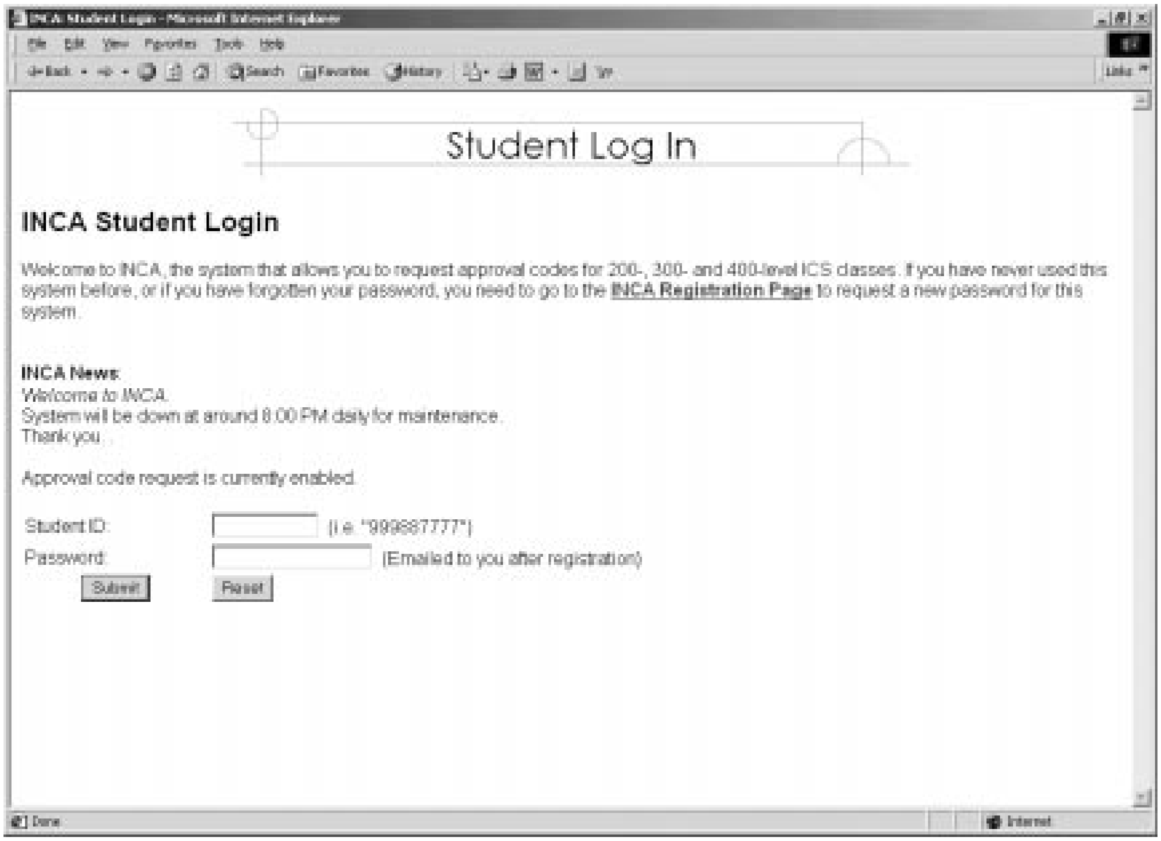
1.3 A guided tour of INCA

INCA is a web-based approval code allocation system designed to solve problems with the manual approach to approval code allocation. The basic idea behind operation of INCA is simple. The students use a web-browser and submit a list of course requests to the INCA server. The INCA server collects all the course requests over a period of time. When the administrator is ready to allocate approval codes, the system selects the top ranked students based upon a set of rules and presents administrator with a list of those students for review. After approval by the administrator, the system automatically sends emails to all of the students simultaneously, informing each of them about the decision made on their course requests and providing them with approval codes, if appropriate. This process is repeated until the semester begins.

There are three types of users of INCA: students, course administrators and site administrators. Students use INCA to add, to delete, or to modify course requests. Course administrators use INCA to allocate approval codes and edit student and course data. Site administrators perform the tasks of site initialization and site maintenance. The following sections describe the student and the administrator usage scenarios.

1.3.1 INCA student scenarios

1. Students visit the INCA website. The students visit the website http://inca.ics.hawaii.edu to request the approval codes for courses they wish to enroll in. A student login page is presented upon visiting the specified link (see figure 1.1). Students need to register with INCA in order to login into the system. The password to enter into the system is emailed to the students once they register with INCA. The INCA login page also acts like a communication point between the administrators and students as the administrators can post any news on the student login page.


INCA Student Login - Microsoft Internet Explorer
File Edit View Favorites Tools Help
Back Search Favorites History
Links
Student Log In
Done
Internet

INCA Student Login

Welcome to INCA, the system that allows you to request approval codes for 200-, 300- and 400-level ICS classes. If you have never used this system before, or if you have forgotten your password, you need to go to the **INCA Registration Page** to request a new password for this system.

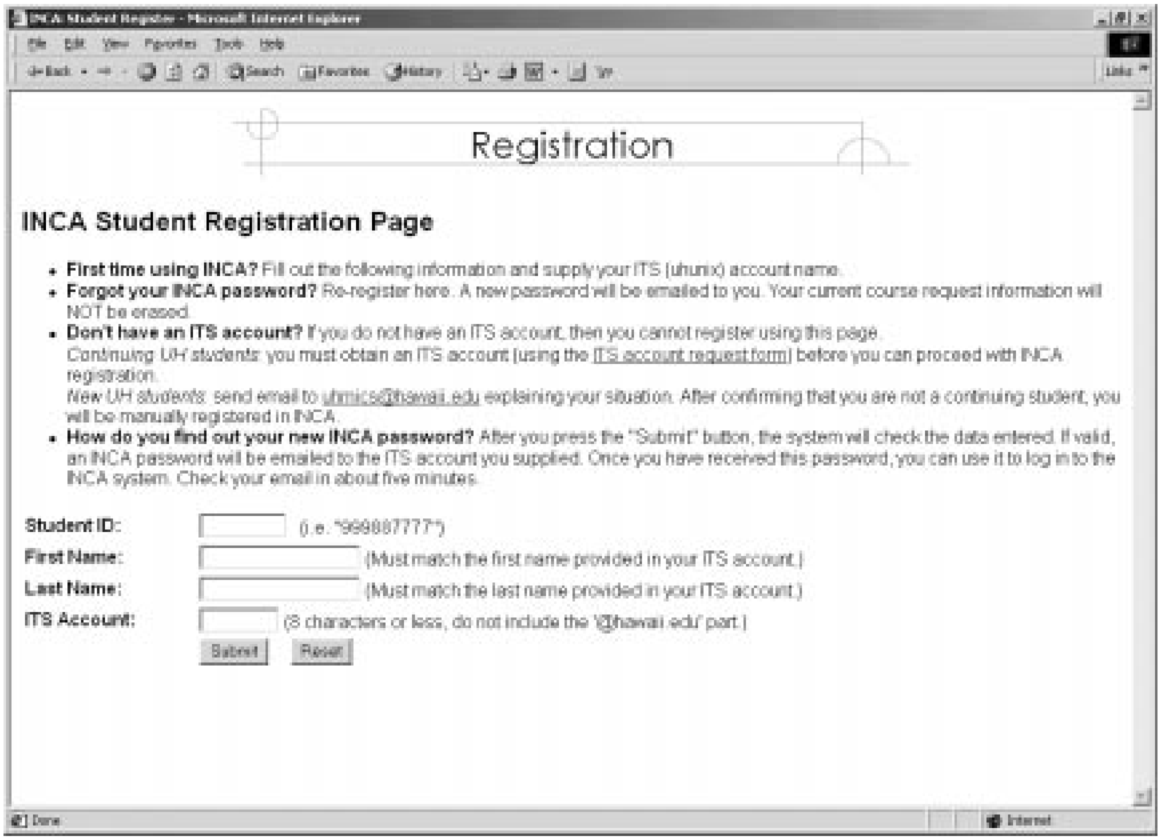
INCA News:
Welcome to INCA.
System will be down at around 8.00 PM daily for maintenance.
Thank you.

Approval code request is currently enabled.

Student ID: (i.e. "999887777")
Password: (Emailed to you after registration)

Figure 1.1 Student login page

2. Students register with INCA. The first time students visit INCA, they must register in order to get a password (see figure 1.2). The registration process requires the students to have an ITS account, which is a user account on the UH UNIX system. If a student doesn't have an ITS account, they generally need to obtain one first. However, there is one exception. New and transferring UH students can email uhmics@hawaii.edu to explain their situation and the INCA course administrator can choose to manually register them. If students forget their password, they can re-register using the INCA registration page and a new password will be emailed to them. The requirement that students have an ITS account is an important component of user security mechanisms in INCA, discussed more thoroughly in Section 3.7.


INCA: Student Register - Microsoft Internet Explorer
File Edit View Favorites Tools Help
Back Search Favorites History
Links
Registration
Done
Internet

INCA Student Registration Page

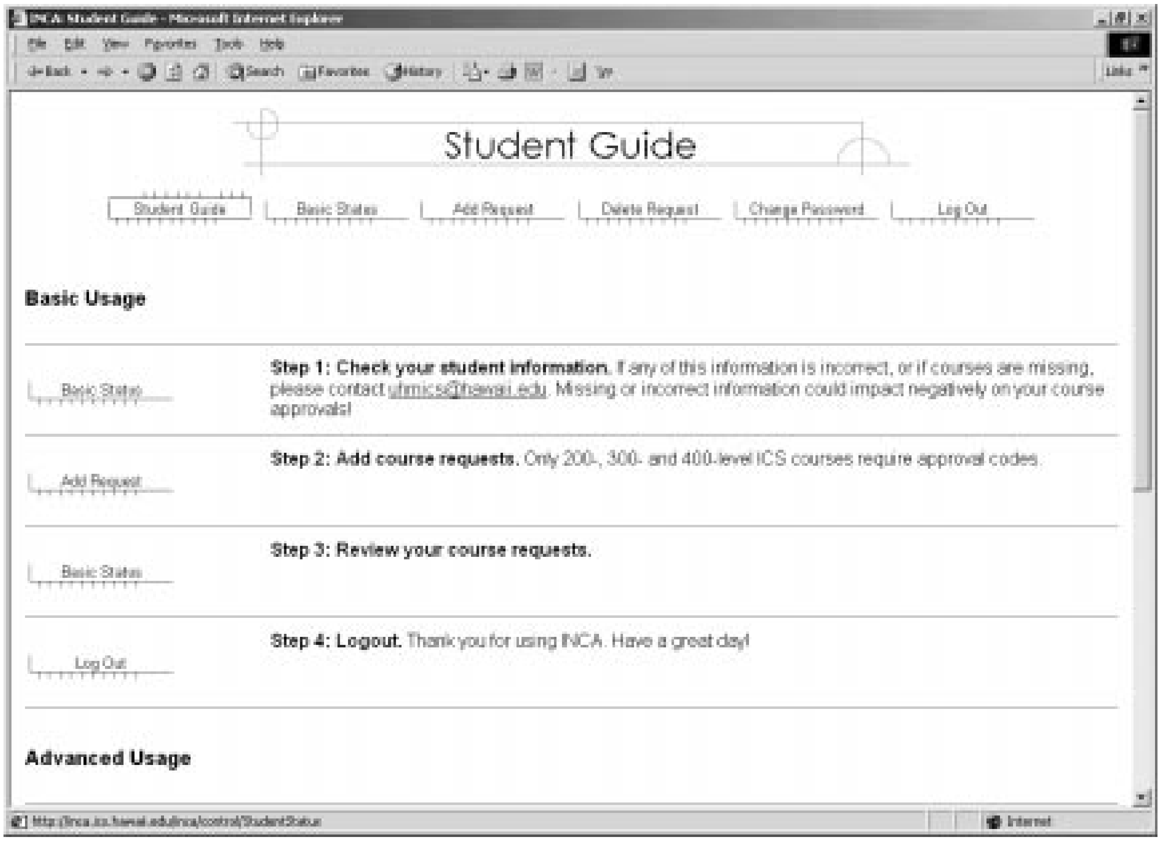
- **First time using INCA?** Fill out the following information and supply your ITS (uhunix) account name.
- **Forgot your INCA password?** Re-register here. A new password will be emailed to you. Your current course request information will NOT be erased.
- **Don't have an ITS account?** If you do not have an ITS account, then you cannot register using this page.
  Continuing UH students: you must obtain an ITS account (using the ITS account request form) before you can proceed with INCA registration.
  New UH students: send email to uhmics@hawaii.edu explaining your situation. After confirming that you are not a continuing student, you will be manually registered in INCA.
- **How do you find out your new INCA password?** After you press the "Submit" button, the system will check the data entered. If valid, an INCA password will be emailed to the ITS account you supplied. Once you have received this password, you can use it to log in to the INCA system. Check your email in about five minutes.

Student ID: (i.e. "999887777")
First Name: (Must match the first name provided in your ITS account.)
Last Name: (Must match the last name provided in your ITS account.)
ITS Account: (8 characters or less, do not include the '@hawaii.edu' part.)

Submit Reset

Figure 1.2 Student registration page

3. Students enter the system and learn how to use it. If the registration is successful, students obtain a password that allows them to enter the system. Upon successful login, they presented with the student guide page (see figure 1.3) that instructs them on how to use the system. The basic steps involve: checking their status for any missing or incorrect information; adding course requests; viewing course requests; and logging out of the system. In addition, the student guide page also explains how to do other tasks such as: modifying course requests; viewing the entire list of courses with qualification explanations; checking their rank in the approval queue and associated priority explanations; and changing their passwords.


INCA: Student Guide - Microsoft Internet Explorer
File Edit View Favorites Tools Help
Back Search Favorites History
Links
Student Guide
Student Guide
Basic Status
Add Request
Delete Request
Change Password
Log Out
http://inca.ics.hawaii.edu/inca/control/StudentStatus
Internet

Basic Usage

Basic Status — **Step 1: Check your student information.** If any of this information is incorrect, or if courses are missing, please contact uhmics@hawaii.edu. Missing or incorrect information could impact negatively on your course approvals!

Add Request — **Step 2: Add course requests.** Only 200-, 300- and 400-level ICS courses require approval codes.

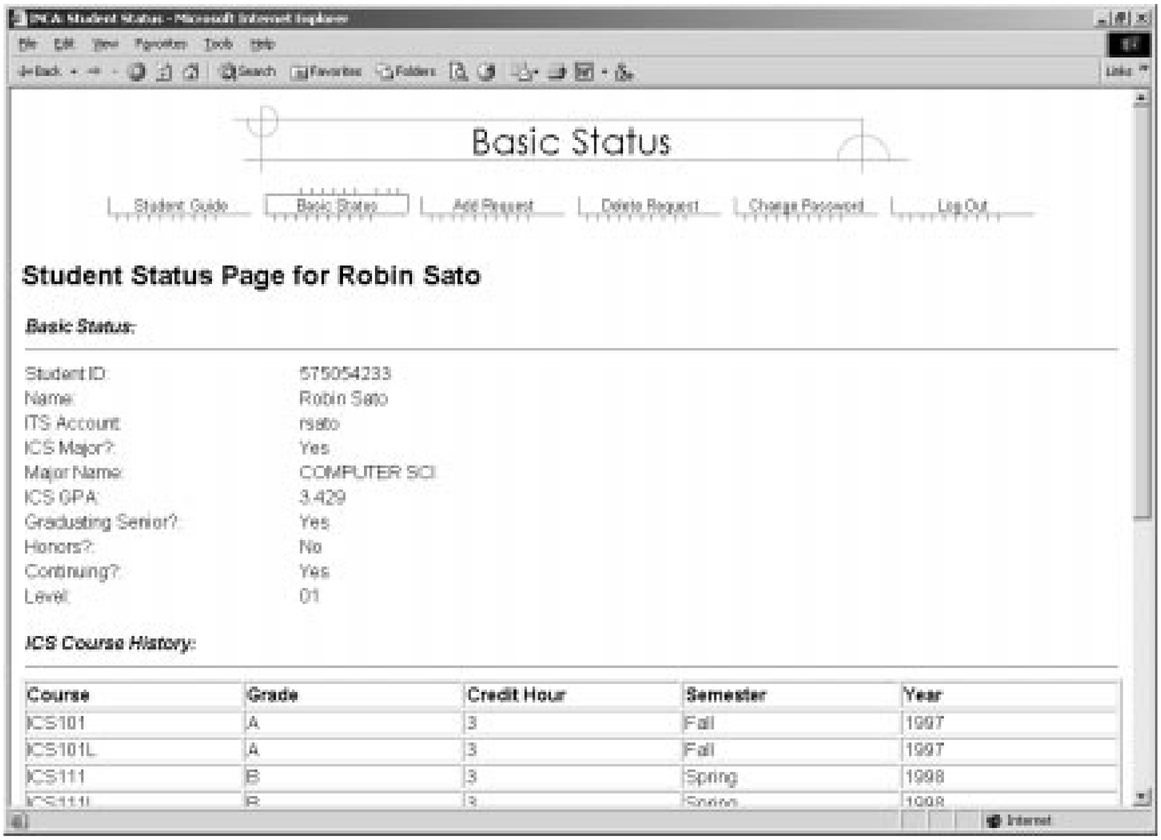
Basic Status — **Step 3: Review your course requests.**

Log Out — **Step 4: Logout.** Thank you for using INCA. Have a great day!

Advanced Usage

Figure 1.3 Student guide page

4. Students check their basic status and ICS course history. The student status page has three sections: basic status, ICS course history, and request and approvals. The basic status section displays the student's GPA, and whether they are ICS majors, graduating seniors, honors students or continuing students. It also displays their major and whether they are at graduate or undergraduate level. The ICS course history section shows the courses taken at ICS department in the past and associated grades. Students review their basic status information and ICS course history and notify the ICS administrator of any incorrect or missing information. Finally, the student status page also displays the information related to course requests made by students and the approval codes given to them. Figure 1.4 shows student status for a fake student.


INCA: Student Status - Microsoft Internet Explorer
File Edit View Favorites Tools Help
Back Search Favorites Folders
Links
Basic Status
Student Guide
Basic Status
Add Request
Delete Request
Change Password
Log Out
Internet

Student Status Page for Robin Sato

Basic Status:

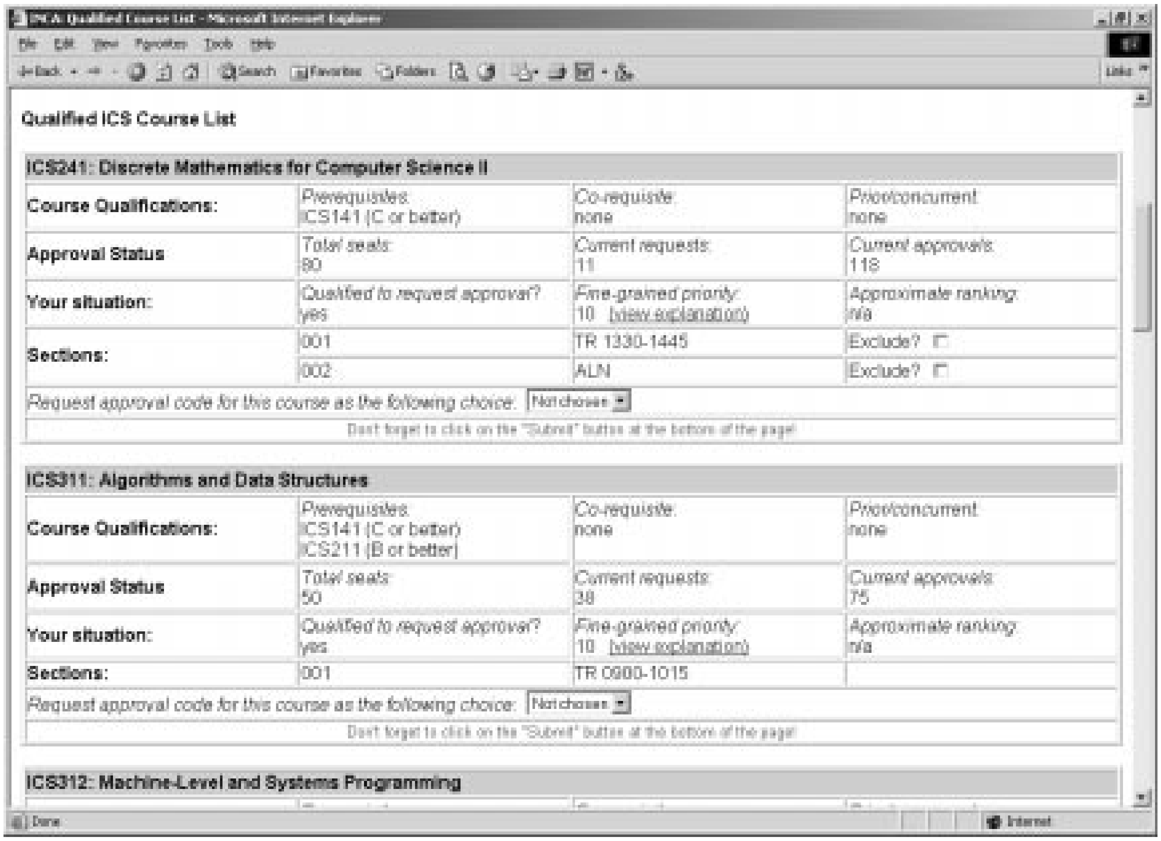
Student ID:	575054233
Name:	Robin Sato
ITS Account:	rsato
ICS Major?:	Yes
Major Name:	COMPUTER SCI
ICS GPA:	3.429
Graduating Senior?:	Yes
Honors?:	No
Continuing?:	Yes
Level:	01

ICS Course History:

Course	Grade	Credit Hour	Semester	Year
ICS101	A	3	Fall	1997
ICS101L	A	3	Fall	1997
ICS111	B	3	Spring	1998
ICS111L	B	3	Spring	1998

Figure 1.4 Student basic status page

5. Students request the courses. The add request page presents the students with the courses they are qualified for and other information related to each of those courses (see figure 1.5). The information includes course name, sections for that course, the pre-requisites, co-requisite or concurrent course requirements, the total number of seats offered in that course, the total requests that have been made, and the total number of available approval codes for that course. It also displays information on whether students are qualified for a course or not, their fine-grained priority points, and their approximate ranking based upon those priority points. While submitting course requests, the students can rank requests from most preferred to least preferred.



Qualified ICS Course List

ICS241: Discrete Mathematics for Computer Science II			
Course Qualifications:	*Prerequisites:* ICS141 (C or better)	*Co-requisite:* none	*Prior/concurrent:* none
Approval Status	*Total seats:* 80	*Current requests:* 11	*Current approvals:* 118
Your situation:	*Qualified to request approval?* yes	*Fine-grained priority:* 10 (view explanation)	*Approximate ranking:* n/a
Sections:	001	TR 1330-1445	Exclude? □
	002	ALN	Exclude? □

Request approval code for this course as the following choice: Not chosen ▾

Don't forget to click on the "Submit" button at the bottom of the page!

ICS311: Algorithms and Data Structures			
Course Qualifications:	*Prerequisites:* ICS141 (C or better) ICS211 (B or better)	*Co-requisite:* none	*Prior/concurrent:* none
Approval Status	*Total seats:* 50	*Current requests:* 38	*Current approvals:* 75
Your situation:	*Qualified to request approval?* yes	*Fine-grained priority:* 10 (view explanation)	*Approximate ranking:* n/a
Sections:	001	TR 0900-1015	

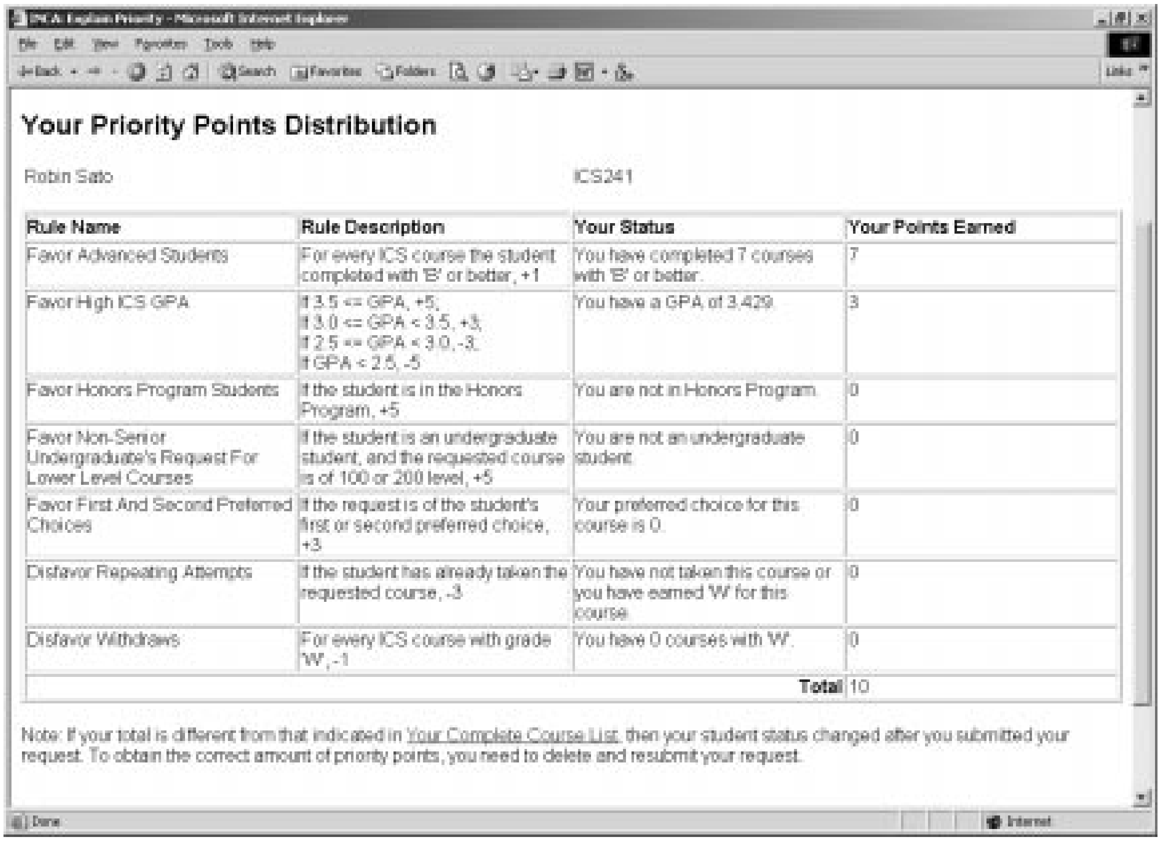
Request approval code for this course as the following choice: Not chosen ▾

Don't forget to click on the "Submit" button at the bottom of the page!

ICS312: Machine-Level and Systems Programming

Figure 1.5 Student qualified course request page

6. Students view their priority points for a course. The students can see the explanation for the fine-grained priority points on the priority point explanation page. The page lists various rules along with their description. It also displays the status of the students and priority points earned by them according to the rules. The rules are designed to favor ICS department majors and students with good academic credentials. The rules include weightings based upon the choice of requests and give more priority points to graduating students. The rules give negative points to the students repeating a particular course or taking a previously withdrawn course. Figure 1.6 shows the priority point explanation page for a fake student.


INCA: Explain Priority - Microsoft Internet Explorer
File Edit View Favorites Tools Help
Back Search Favorites Folders
Links
Done
Internet

Your Priority Points Distribution

Robin Sato ICS241

Rule Name	Rule Description	Your Status	Your Points Earned
Favor Advanced Students	For every ICS course the student completed with 'B' or better, +1	You have completed 7 courses with 'B' or better.	7

Favor High ICS GPA	If 3.5 <= GPA, +5;
	If 3.0 <= GPA < 3.5, +3;
	If 2.5 <= GPA < 3.0, -3;
	If GPA < 2.5, -5
		You have a GPA of 3.429.	3

Favor Honors Program Students	If the student is in the Honors Program, +5	You are not in Honors Program.	0
Favor Non-Senior Undergraduate's Request For Lower Level Courses	If the student is an undergraduate student, and the requested course is of 100 or 200 level, +5	You are not an undergraduate student.	0
Favor First And Second Preferred Choices	If the request is of the student's first or second preferred choice, +3	Your preferred choice for this course is 0.	0
Disfavor Repeating Attempts	If the student has already taken the requested course, -3	You have not taken this course or you have earned 'W' for this course.	0
Disfavor Withdraws	For every ICS course with grade 'W', -1	You have 0 courses with 'W'.	0
		Total	10

Note: If your total is different from that indicated in Your Complete Course List, then your student status changed after you submitted your request. To obtain the correct amount of priority points, you need to delete and resubmit your request.

Figure 1.6 Priority point explanation page

In addition to these basic tasks, students can do the following tasks:

1. Check their ranks in real time. Students can check their most current ranking as their place in approval code queue could potentially change as more and more students making requests. Students can log into INCA every few days, see their rank, and determine their current chances of getting approval codes for popular courses.

2. View list of all ICS courses requiring approval codes. Students can view the entire list of ICS courses, which explains the qualification criterions for all courses and the standing of the students with respect to those qualifications.

3. Modify course requests. Students can delete their old course requests and submit new ones, to modify their course requests.

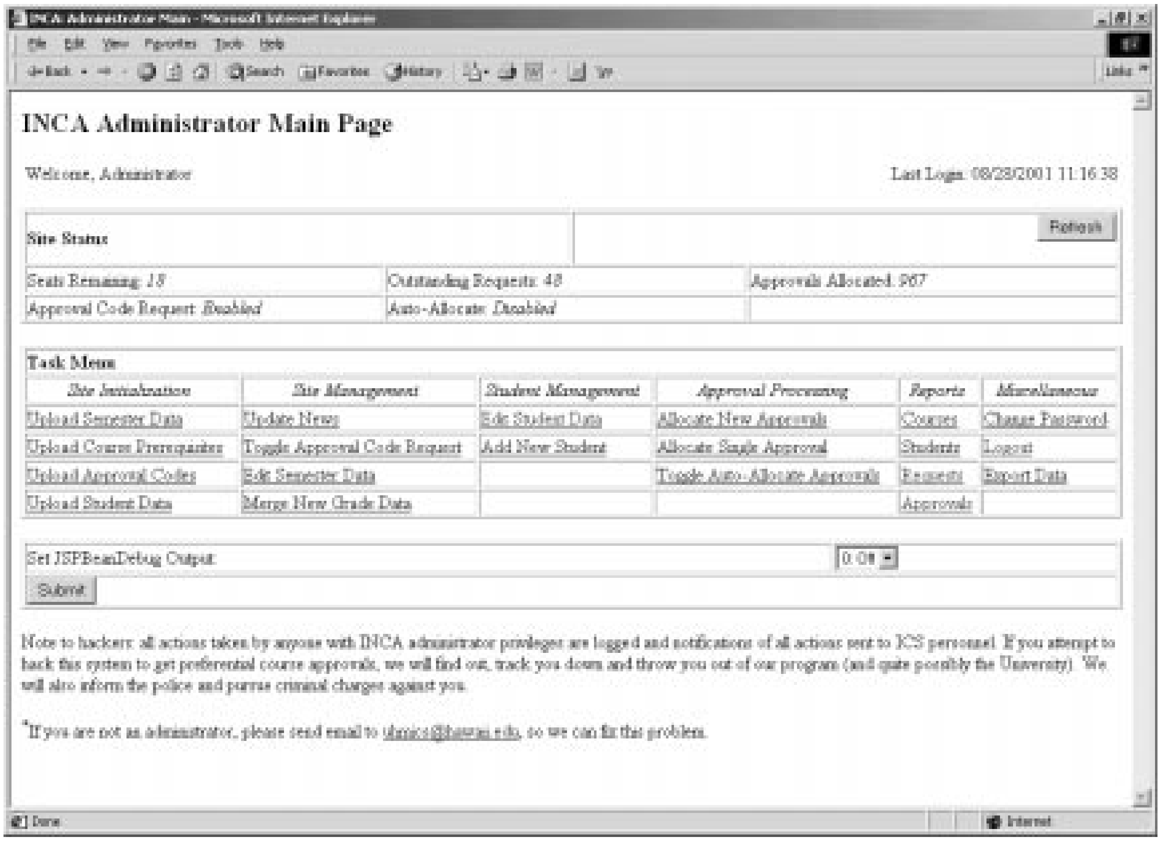
4. Change their password. Students can change their old passwords to new ones.

1.3.2 INCA administrator scenarios

The INCA course administrator and the INCA site administrator use the same administrator account to do their tasks. Both of them are presented with same main page upon successful login. The administrator main page lists all the tasks. The INCA site administrator uses some of the tasks and the course administrator uses others.

INCA site administrators perform the tasks of site initialization and site maintenance from the administrator menu. The site initialization tasks include uploading of course, student, and approval code data from XML files into the system. This is performed once a semester. The site maintenance tasks include exporting data from the database into XML files and merging new grades data, available late during registration period, with the old data. I will describe some of the course administrator scenarios in next few pages.

1. The course administrator enters into the system. The administrator main page (see figure 1.7) is presented to the course administrator upon successful login into the system. The main page allows the administrator to do tasks such as: edit the student and the semester course data; allocate the approval codes one at a time or in bulk; update the news to be displayed on student login page; enable or disable the approval code request and the auto-allocation of approval codes; and change their password. It also shows site status numbers such as: total number of outstanding requests; total number of approvals allocated; and total number of course seats remaining. The details of these numbers are presented in requests and approvals reports.


INCA Administrator Main - Microsoft Internet Explorer
File Edit View Favorites Tools Help
Back Search Favorites History
Links
Done
Internet

INCA Administrator Main Page

Welcome, Administrator

Last Login: 08/28/2001 11:16:38

Site Status		Refresh
Seats Remaining: *18*	Outstanding Requests: *48*	Approvals Allocated: *967*
Approval Code Request: *Enabled*	Auto-Allocate: *Disabled*	

Task Menu

Site Initialization	*Site Management*	*Student Management*	*Approval Processing*	*Reports*	*Miscellaneous*
Upload Semester Data	Update News	Edit Student Data	Allocate New Approvals	Courses	Change Password
Upload Course Prerequisites	Toggle Approval Code Request	Add New Student	Allocate Single Approval	Students	Logout
Upload Approval Codes	Edit Semester Data		Toggle Auto-Allocate Approvals	Requests	Export Data
Upload Student Data	Merge New Grade Data			Approvals	

Set JSPBeanDebug Output: 0:Off

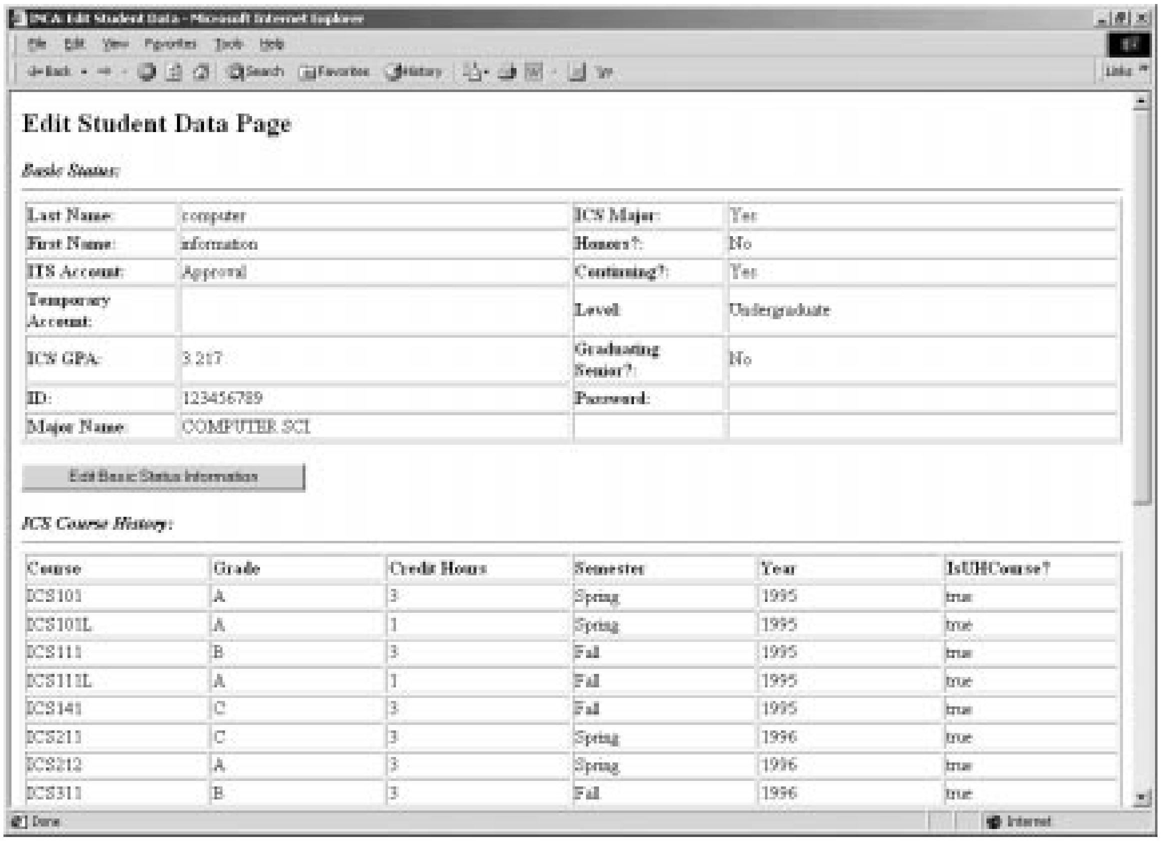
Submit

Note to hackers: all actions taken by anyone with INCA administrator privileges are logged and notifications of all actions sent to ICS personnel. If you attempt to hack this system to get preferential course approvals, we will find out, track you down and throw you out of our program (and quite possibly the University). We will also inform the police and pursue criminal charges against you.

*If you are not an administrator, please send email to uhmics@hawaii.edu, so we can fix this problem.

Figure 1.7   Administrator main page

2. The course administrator edits the student and semester records. The edit student data page (see figure 1.8) is divided into four sections: edit the basic status information of student; edit the ICS course history of student; edit the requests made by the student; and edit the consents for courses given to the student. The entire information related to the student is presented on edit student data page and a new edit page is presented for each section. In addition to editing the student records, course administrator can also add new students into the system. INCA allows the course administrator to add, delete, and modify the course sections information as some sections are added and removed late during the registration period.


INCA: Edit Student Data - Microsoft Internet Explorer
File Edit View Favorites Tools Help
Back Search Favorites History
Links
Done
Internet

Edit Student Data Page

Basic Status:

Last Name:	computer	**ICS Major:**	Yes
First Name:	information	**Honors?:**	No
ITS Account:	Approval	**Continuing?:**	Yes
Temporary Account:		**Level:**	Undergraduate
ICS GPA:	3.217	**Graduating Senior?:**	No
ID:	123456789	**Password:**	
Major Name:	COMPUTER SCI		

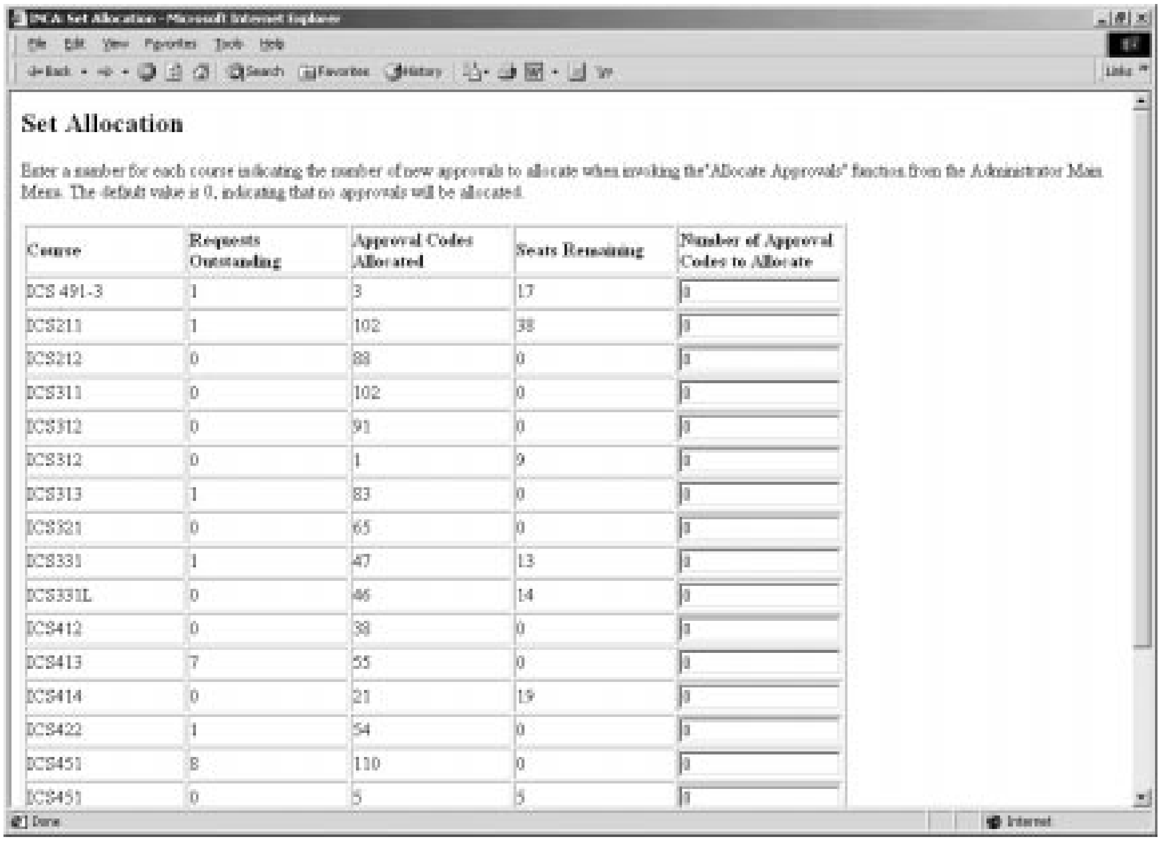
Edit Basic Status Information

ICS Course History:

Course	Grade	Credit Hours	Semester	Year	IsUHCourse?
ICS101	A	3	Spring	1995	true
ICS101L	A	1	Spring	1995	true
ICS111	B	3	Fall	1995	true
ICS111L	A	1	Fall	1995	true
ICS141	C	3	Fall	1995	true
ICS211	C	3	Spring	1996	true
ICS212	A	3	Spring	1996	true
ICS311	B	3	Fall	1996	true

Figure 1.8 Edit student data page

3. The course administrator allocates the approval codes. The course administrator can allocate the approval codes one at a time or in bulk. The course administrator needs to set an allocation policy in order to allocate the approval codes in bulk (see figure 1.9). The allocation policy lets the administrator distribute a certain number of approval codes at a time. Once the administrator sets the policy, INCA presents the administrator with list of all the students that have been given the codes, that haven't been given codes, and that the system proposes to give codes. The administrator reviews the list and makes final changes, if any. Then, INCA sends out automatic emails to all the students containing the approval codes.


INCA: Set Allocation - Microsoft Internet Explorer
File Edit View Favorites Tools Help
Back Search Favorites History
Links
Done
Internet

Set Allocation

Enter a number for each course indicating the number of new approvals to allocate when invoking the"Allocate Approvals" function from the Administrator Main Menu. The default value is 0, indicating that no approvals will be allocated.

Course	Requests Outstanding	Approval Codes Allocated	Seats Remaining	Number of Approval Codes to Allocate
ICS 491-3	1	3	17	0
ICS211	1	102	38	0
ICS212	0	88	0	0
ICS311	0	102	0	0
ICS312	0	91	0	0
ICS312	0	1	9	0
ICS313	1	83	0	0
ICS321	0	65	0	0
ICS331	1	47	13	0
ICS331L	0	46	14	0
ICS412	0	38	0	0
ICS413	7	55	0	0
ICS414	0	21	19	0
ICS422	1	54	0	0
ICS451	8	110	0	0
ICS451	0	5	5	0

Figure 1.9 Set allocation policy page

1.4 Research issues

INCA is a new system. As with every new system, many issues must be successfully addressed in order to make the system widely acceptable. Table 1.1 lists the most important research issues from six perspectives: administrator, student, department, designing allocation systems, commercial, and technological.

Table 1.1 The research issues from different perspectives

Perspective	Research issues
Administrator	• Does INCA reduce the administrative overhead or does it introduce new kinds of problems?
Student	• Does INCA makes the approval code allocation process visible, predictable and understandable to the students?
Department	• What kinds of useful data INCA can provide to the departments? • In what ways is INCA helpful to the faculty? • Does INCA really improve the quality of the ICS program?
Designing allocation systems	• How should we design the rules? • What weights should be assigned to each rule? • Should the ranking systems be transparent or opaque?
Commercial	• What are the other areas in which we can apply INCA allocation technology? • Can other departments across University of Hawaii use INCA? • Can other universities across US use INCA?
Technological	• What are the different technologies available for building multi-tier systems and how does the technology used to build INCA compare with those technologies?

1.5 Thesis statement

INCA is a web-based approval code allocation system that automates the approval code allocation process during registration sessions. It benefits:

1. *Administrators, by decreasing administrative overhead.*
2. *Students, by making the allocation process predictable and understandable to them.*
3. *Departments, by providing them with valuable data that can be used for feedback and planning on their curriculum.*

Table 1.2 provides the operational definitions of these claims:

Table 1.2 The operational definitions of thesis claims

Term	Operational definition
Administrative overhead	Reduction in administrative overhead means reduction in total time spent on reading and responding to student emails.
Predictability	A predictable allocation process allows students to predict their chances of getting into a course, based upon their ranking in the approval queue.
Understandability	An understandable allocation process makes the students understand the decisions behind the approval code allocation. Student's grades, major, advancement into ICS program, and prerequisite and co-requisite course requirements are the major factors that govern the approval code allocation decisions.
Valuable data	INCA provides valuable data to the department in terms of the demand of the different courses that supports future curriculum planning.

In order to evaluate first claim, I analyzed the emails sent to "uhmics" account during Spring 2001 semester (when INCA was not used) and Spring 2002 semester (when INCA was used). During Spring 2001 semester, students used uhmics account to submit their course requests by email. An ICS administrator used to check all the emails and respond to them in a timely manner. During Spring 2002 semester, students used uhmics account to report problems with INCA. I looked into the nature of problems that occurred before and after INCA was in use. To evaluate second claim, I conducted a user survey of students who used INCA. To evaluate third claim, I investigated the data stored in INCA.

1.6 Structure of the thesis

The second chapter discusses the work related to INCA. I will discuss different types of information systems at universities and how INCA fits into the big picture. Then, I will discuss different technologies used to build multi-tier systems and how technology used to build INCA compares with them. Finally, I will discuss different type of allocation systems and the relevance of procedural justice in designing allocation systems.

The third chapter discusses the architecture of INCA, the design of the three tiers and the communication between them.

The fourth discusses the implementation of INCA.

The fifth shows how I designed an evaluation of INCA from the perspectives of students, administrator, and the department. I will also describe the results and how they relate to my hypotheses in chapter five.

Finally, in the last chapter, I discuss conclusions and future directions.

Chapter 2. Related work

This chapter discusses related work from three different perspectives. First, I will describe information systems used at UH and how INCA fits into the big picture. I will also discuss the information systems similar to INCA used at other universities. Then, I will discuss INCA from a technological point of view. I will describe different technologies used to build multi-tier systems and how technology used to build INCA compares with those technologies. Finally, I will discuss different types of allocation systems. I will describe the concepts of procedural justice and productivity in education and how they are relevant to INCA design.

2.1 Information systems

Universities need to manage information to do their operations in a systematic manner. The information systems used at universities can be roughly classified into three categories shown in Table 2.1.

Table 2.1 Information systems used at universities

System	Tasks
HRMS (Human Resource Management System)	Human resources, Payroll, Stock administration, Payroll interface, Pension administration
Financials	Asset management, Payables, Receivables, General ledger, Projects, Budgets, Purchasing, Inventory, Billing, Contracts
Student Information Systems	Recruiting and admissions, Student academic records, Academic advising, Student financials, Course enrollment

Universities use information systems to do their administrative tasks. These information systems differ from university to university because of factors including funds availability, administration, faculty, and staff. Some universities keep all their data on old mainframe systems, whereas others have adopted modern systems. The development language and hardware platforms differ. Some systems are centralized and some are web-based. Some universities buy solutions from vendors, whereas others build in-house solutions. The needs and situations differ from university to university and so are the types of information systems. In the next section, I will describe the information systems being used at UH and how INCA fits with them.

2.1.1 Information systems at UH

University of Hawaii uses 10-15 different kinds of information system to manage the administrative and student information. Table 2.2 describes the different kinds of information systems being used at UH. At the end of the table, I describe INCA with regard to other systems at UH. It is to give an overview of how INCA fits into big picture.

Table 2.2   Information systems at UH

System	Description
CAPIS	Coordinated Admissions Program Information System. Produces reports related to admission activities.
DARS	Degree Audit Reporting System. Academic advising. Compares a student's academic work with the requirements of an institution's academic program and prepares a comprehensive report detailing student progress toward meeting those requirements.

FMIS	Financial Management Information System. Maintains Financial information for entire University of Hawaii. Sub-systems include General Ledger, Accounts Payable, Contracts and Grants, Budget Level Summary, Payroll Inquiry, Purchasing for OPRPRM and fiscal officer authorized purchase orders, Accounts Receivable, Departmental Checking and Fixed Assets.
ISIS	Integrated Student Information System. Maintains course catalog, schedule of classes, admissions, student biographic data, registration, and academic history.
HRMS	Human Resource Management System
PFIS	Physical Facilities Information System. Provides information on space utilities
SAIS	Student Aid Information System. Maintains and tracks financial aid information
SECE	Student Employment and Cooperative Education. Allows students to search jobs and employers to post jobs
SIMS	Student Information Management System. Provides student and course data for planning, policy making and decision support
SIS	Student Information System. Provides system-wide management information on student enrollment and activities.
INCA	INternet Course Allocation. Used for approval code allocation with the departments.

The following table shows the development languages, vendor and hardware platforms for the systems mentioned above. Some of the systems are developed in-house, whereas others are bought from higher education system vendors. Most of the systems are developed in COBOL and run on university mainframe system. The detailed information about these systems can be obtained from MIS ITS website [11]. Again, at the end of table 2.3, I describe INCA with regard to the information shown in the table.

Table 2.3 Development and deployment details of UH information systems

System	Language	Vendor	Platform
CAPIS	Natural and Cobol	In-house	Runs of University's IBM Mainframe platform. It is a batch system run once for fall and spring semesters to produce managerial and operational reports relating to admission activities for decision making
DARS	COBOL	Purchased from Miami University	Purchased by College of Arts and Sciences in 1994. Runs on University's IBM mainframe platform. DARS reports are also accessible via a web browser
FMIS	Developed from Software AG's Financial Records System (FRS) software	In-house	Installed on University's IBM Mainframe
ISIS	COBOL, Software AG's Natural	Purchased from System and Computer Technology Corporation (SCT)	Runs on UHM's IBM mainframe platform
HRMS	PeopleSoft tools	PeopleSoft	Version 5.12 on IBM RS6000 Version 7.5 on Sun Solaris
PFIS	COBOL	Developed by	Runs on University's IBM mainframe

		ITS MIS	platform
SAIS	Natural	In-house	Runs on University mainframe
SECE	Java	Joint project between UH Manoa SECE and MIS	Java based web application. Utilizes LDAP for user authentication
SIMS	Natural	In-house	Runs on University’s IBM Mainframe
SIS	COBOL and Natural	In-house	Runs on University’s IBM mainframe platform
INCA	Java, EJB, JSP, servlets, XML	Developed in CSDL	It is an n-tier system. Requires a web server, an application server and a database server. The web server should be capable of supporting servlets and JSP. The application server should be capable of supporting EJBs. Faster processor and more RAM will result in better INCA performance.

2.1.3 Course allocation at other universities

Almost every university may have different list of information systems from that of above. INCA is closely related to ISIS and PA’E. For the purposes of this research, I will look at the registration systems being used at other universities, how they handle the problem of “course allocation”, and what systems they have that are similar to INCA.

I looked into registration systems used at more than 20 different universities. I selected these universities at random. Most of the universities use phone-based or web-based registration systems. Most of them also have concept of restricted classes.

However, they adopt different approaches for handling that issue. Most require the student to consult their academic advisor and fill out some kind of form to get into restricted classes. Only one of the 20 universities, University of Indiana at Bloomington, has the concept similar to approval codes, which they call “registration access codes”. But, they don’t have a system similar to INCA and they use a manual method of allocating the registration access codes. Table 2.4 lists the registration systems used at different universities and whether or not they have a concept similar to approval codes or not.

Table 2.4 Registration systems used at other universities

University	Phone registration	Web registration	Approval codes	Registration page
Alabama	Yes	TideWeb	-	registrar.ua.edu
Arizona	RSVP	Webreg	-	www.registrar.arizona.edu
California	URSA	URSA	-	ursa.ucla.edu
Georgia	-	OASIS	-	www.reg.uga.edu
Idaho	-	-	-	www.uidaho.edu/registrar
Illinois	Yes	Yes	-	www.online.uillinois.edu
Indiana	-	Regweb	Registration access codes	registrar.Indiana.edu
Kentucky	UK VIP	-	-	www.uky.edu/registrar
Michigan	-	Wolverinacess	-	www.umich.edu/~reg
New Jersey	RTTRS	Webreg	-	registrar.Rutgers.edu
North Carolina	-	-	-	regweb.uit.unc.edu
Ohio-state	-	-	-	www.ureg.ohio-state.edu
Oregon	Duck Call	Duck Web	-	registrar.uoregon.edu

South Carolina	-	VIP	-	argo.regs.sc.edu
Tennessee	Stripes	Tigerweb	-	enrollment.Memphis.edu
Texas	TEX	ROSE	-	www.texas.edu/student/registrar
Utah	-	-	-	www.saff.utah.edu/regist
West Virginia	STAR	-	-	www.arc.wvu.edu/star
Wisconsin	-	Yes	-	www.wisc.edu
Wyoming	STAR	Hole in the wall	-	siswww.uwyo.edu/reg

2.1.4 INCA

INCA is a web-based system that can be used in conjunction with the PA’E to register for class. PA’E is university-level system, i.e., it allows all the students inside a university to register for the classes in whatever department they want. INCA is a department-level system, i.e., it allows the students in a department to request for approval codes from the department so that they can register for the classes over PA’E.

2.2 n-tier database systems

INCA is an example of an n-tier system. The client, usually a web browser, makes the requests to the web server in the presentation tier. If the request requires the processing not possible at the web tier, then, it is passed to the application server in the business tier. If the processing of request needs access/changes to data in the database, the application server contacts the database server in database tier. The results of processing are communicated back from application server back to web server and further back to the client.

There are a number of technologies that can be used to build n-tier systems. J2EE, Microsoft .NET and CORBA/C++ are three major ones used to build n-tier systems.

2.2.1 Overview of multi-tier systems

Figure 2.1 illustrates the concept of multi-tier systems. It shows how to construct a 2-tier, 3-tier and n-tier systems via 3 scenarios.

1. Client directly accesses the database. This happens in 2-tier systems.

2. Client or web browser makes request to presentation tier and presentation tier contacts database. Or Client directly accesses the application server and application server contacts the database. This happens in 3-tier systems.

3. Client or web browser makes request to presentation tier. It passes on the request to business tier that further passes request to database tier. This happens in an n-tier system.

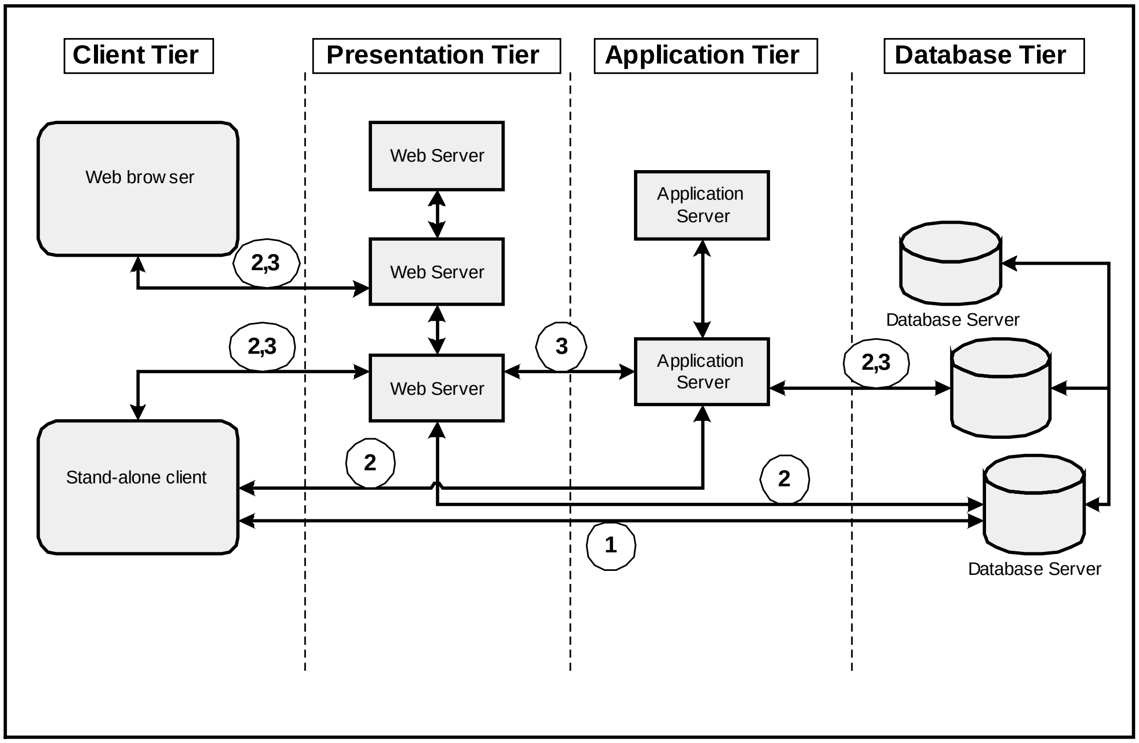


Figure 2.1 Different scenarios in n-tier systems

2.2.2 J2EE

The Java 2 Platform Enterprise Edition (J2EE) was designed to simplify the development, deployment, and management of multi-tier enterprise solutions. J2EE is a set of specifications released by Sun Microsystems. It is not a product. Vendors like BEA, IBM and Oracle implement the J2EE specifications to build the enterprise products. In addition to the specification, Sun has also released a reference implementation of J2EE known as J2EE-RI.

The J2EE architecture is based on Java programming language. All J2EE components are written using Java programming language. These components are transformed into Java byte codes and executed by a JRE at runtime. A J2EE application is hosted within a container, which provides qualities of service necessary for enterprise application, such as transaction, security and persistence services.

Java servlets and Java server pages are used for presentation in the presentation tier. They communicate with Enterprise Java Beans (EJB) in the business tier using RMI/IIOP. The EJBs communicate with the database in the database tier using JDBC.

Some of the popular products implementing J2EE specifications include Weblogic from BEA systems, Websphere from IBM, JRun from Allaire, and Jboss, an open-source application server. Currently, There are over 30 products that implement J2EE specification.

2.2.3 Microsoft .NET

Microsoft .NET is product suite that enables organizations to build enterprise-class web services. Microsoft .NET is largely a rewrite of Windows DNA, which was Microsoft's previous platform for developing enterprise applications. Windows DNA includes many proven technologies that are in production today, including Microsoft Transaction Server (MTS) and COM+, Microsoft Message Queue (MSMQ), and the Microsoft SQL Server database. The new .NET Framework replaces these technologies, and includes a web services layer as well as improved language support.

Microsoft .NET architecture is based on Common Language Runtime (CLR), analogous to the JRE. .NET components can be written in any language like C++, CB, C# and even COBOL. All these get translated into Microsoft Intermediate Language (MSIL), which is analogous to Java byte codes. The IL code is interpreted and translated into a native executable using CLR.

ASP.NET is used for presentation in the presentation tier. The ASP pages communicate with COM+ components in the business tier using DCOM or SOAP. The COM+ components communicate with the database in the database tier using ADO.NET.

.NET platform includes following .NET enterprise servers: SQL server 2000, Exchange server 2000, Commerce server 2000, Application center server 2000, Host Integration server 2000, Internet Security and Acceleration Server 2000 and BizTalk Server 2000.

2.2.4 A comparison of technologies

There are different factors that can be used to compare technologies used for building multi-tier systems. These include time-to-market, vendor support, legacy systems support, platform maturity, languages and web services support, and portability. Here, I compare J2EE and .NET in terms of n-tier technology infrastructure support. Figure 2.2 draws the comparison.

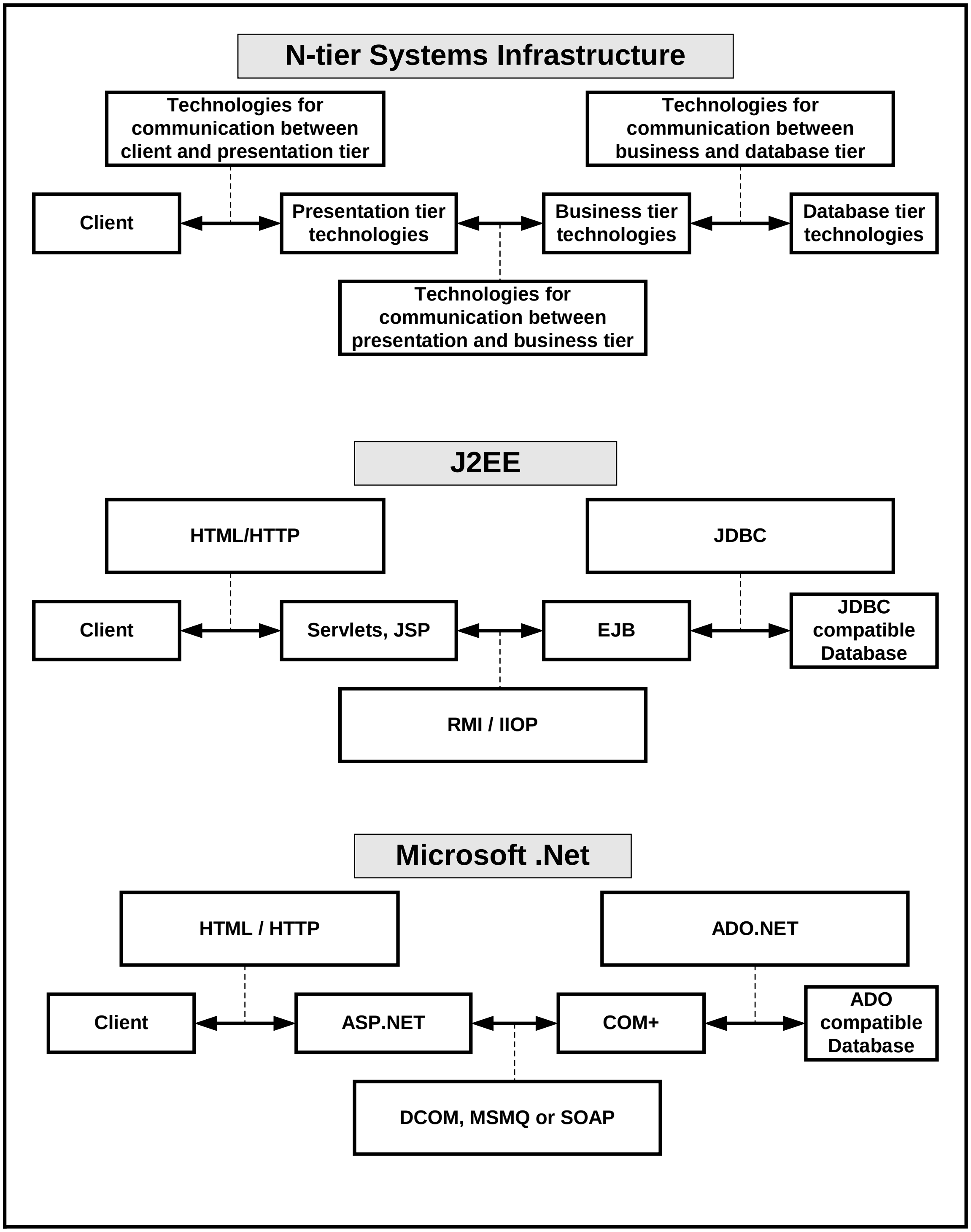

N-tier Systems Infrastructure
Technologies for communication between client and presentation tier
Technologies for communication between business and database tier
Client
Presentation tier technologies
Business tier technologies
Database tier technologies
Technologies for communication between presentation and business tier
J2EE
HTML/HTTP
JDBC
Client
Servlets, JSP
EJB
JDBC compatible Database
RMI / IIOP
Microsoft .Net
HTML / HTTP
ADO.NET
Client
ASP.NET
COM+
ADO compatible Database
DCOM, MSMQ or SOAP

Figure 2.2 Comparing J2EE and Microsoft .NET

2.2.5 INCA

INCA is implemented using J2EE technology. INCA was run on J2EE-RI (Java 2 Enterprise Edition, Reference Implementation) from Sun Microsystems, for Fall 2001 semester. J2EE-RI packages web server, application server and database server into a single product. The web server is Jakarta Tomcat and the application server is from Sun Microsystems. The database server is Cloudscape. INCA was also successfully deployed on other application servers. This includes BEA WebLogic that has a web server and application server integrated into a single product. INCA was also deployed on the combination of “resin” web server, “Jboss” application server and “cloudscape” database server. Weifeng Miao’s master’s thesis [9] discusses the deployment details.

2.3 Rule-based prioritization systems

INCA is a rule-based prioritization system that applies certain rules to prioritize the students and fulfills the requests of top-ranked students. The approval code distribution was based only on seniority prior to INCA. Adopting a new technology in an organization has its own social implications, which we discuss below.

2.3.1 Allocation tasks and their classification

Jon Elster [4] has done work on allocation systems. He examined a large variety of non-market institutions where scarce goods are allocated. Some examples of the allocation tasks considered are:

- Who gets a kidney for transplantation?
- Who is admitted to selective colleges?
- Who is selected for layoffs?

- Who is chosen for military service in time of war?
- Who is allowed to adopt children?
- Who is allowed to immigrate?

Elster classifies allocation tasks in terms of selection, admission, and placements as follows:

- Selection involves ranking individuals and allocating the scarce good from the top to bottom, until it is exhausted.
- Admission involves comparing individuals against some threshold, and allocating the good to all who pass.
- Placement involves matching individuals to heterogeneous units of good.

Some of the criterions that play a role in decisions are as follows:

- Merit (as in selection to receive honors)
- Need (as in selection for space in intensive care units)
- Seniority (as in layoffs in unionized firms)
- Notions of equity.

Other examples of allocation systems include credit-card request processing systems. The systems apply certain rules before the customer's request for credit card is approved.

INCA is a selective allocation system. Approval codes used to be given to the students on a first come first serve basis, after doing an eligibility check, prior to INCA. With INCA, a ranking list is produced by giving points to the students based on various rules and then selecting the students from the list. Some students do complain about the selection policy implemented by INCA. So, INCA needs to be evaluated to determine whether it really improves the quality of ICS program.

2.3.2 Procedural justice

“Procedural justice” refers to a research literature that examines how perceptions of the means (or procedures) through which decisions are made in organizations affect people’s satisfaction with or attitude towards these organizations. This is a relatively large body of literature [5]. Here, I will present a summary of it.

“The fairness of a firm’s procedures may have a greater impact on organizational commitment than the fairness of personal outcomes that workers receive, perhaps because procedures define the organization’s capacity to treat employees fairly. Thus, if they see procedures as fair, employees may view the organization positively, even if they are currently dissatisfied with such personal outcomes.”

The implication is that if there is a visible (i.e. clearly articulated) procedure which is viewed as fair, then, even if someone is unhappy with his/her individual outcome under the procedure, he/she will probably still view the organization positively.

INCA also uses priority rules to rank the students before allocating the approval code. Setting priorities is something that cannot make everyone happy. Some of the rules may heavily favor somebody and heavily disfavor the others. However, if the rules are visible and fair, then, even if somebody is dissatisfied with the rules, they will view it positively and accept them.

Chapter 3. INCA Design

INCA is an n-tier system consisting of a client tier and three server-side tiers. There are a number of technologies used for building n-tier systems. Sun Microsystems J2EE and Microsoft .NET are the leading technologies among them. INCA is developed using J2EE technology. Sun Microsystems has released J2EE blueprints in an application, called Java Pet Store (JPS). INCA design closely follows the JPS design.

In this chapter, I will discuss the design of INCA. I will describe the history and motivation behind INCA first. Then, I will discuss the modular design of INCA and how the modular design is implemented using J2EE technology components. Next, I will discuss the design of individual tiers and the communication between them. Finally, I will discuss the security issues and the design patterns in INCA.

3.1 INCA history

It was Summer 2000 when Professor Philip Johnson from ICS department started thinking about an automated system to solve problems with manual approval code distribution. He started a project named “Courseapp” to build the automated system while teaching his class ICS613, Advanced Software Engineering, during Fall 2000 semester. He was teaching students software engineering along with Java 2 Enterprise Edition (J2EE) technology. While teaching he designed the initial version of Courseapp with the following design goals in mind:

1. The system should reduce the administrative overhead involved in checking student emails, checking their records over ISIS, and appropriately responding to those emails.

2. The system should improve the quality of ICS program by promoting the students with good academic credentials.
3. The system should capture useful course demand information to help departments respond to the student needs.
4. The system should make the process of approval code allocation understandable to students.

Near the end of the semester, 20 students started coding the system and they coded it for around two weeks. When the semester was over, the system was not even close to functional. Six students under the guidance of Professor Philip Johnson, proceeded to complete the system so that it could be used for Fall2001 semester, and the system was renamed as “INCA”. After one month, two members left. The remaining four students, along with Professor Philip Johnson worked day and night for around four months and the system went live on April 30, 2001 for Fall 2001 approval code distribution. Along with David Liang, Paula Nishida, and Weifeng Miao, I worked on the system. At the time of this writing, only two students, Weifeng Miao and I, are maintaining this system, which now consists of more than 50,000 lines of Java code and 400 files.

3.2 High level architecture

The high-level architecture of INCA will be discussed in two steps. I will describe the major modules in INCA, their responsibilities, and the communication between them first. Then, I will discuss how these modules are implemented using J2EE technology.

3.2.1 Modular Design

INCA consists of six major modules: Course, Student, Request, Approval, Control, and User Interface. The description of modules along with their responsibilities is as follows:

Course. This module maintains the course schedule information for the current semester. It is responsible for maintaining and providing following information:

- Course information such as the course number, the course name, and the course credit hours.
- Section information such as the section number, the section timings, the number of seats in the section, and the name of instructor teaching the section.
- Course requisite requirements for different courses. Courses may have prerequisite, corequisite, and concurrent course requirements. Some courses may require consents. Some courses may require certain grade in order to qualify.

Student. This module maintains all the information related to a student. It maintains and provides following information:

- Student basic information such as the student name, the student ID, the email account, the major name, and the departmental GPA.
- Student status information such as the student level (indicates whether student is an undergraduate or graduate), the graduating status (indicates when student will graduate), and the program status (indicates whether student is a departmental major or is in the honors program).
- Grade information such as the courses student took in the past, and the grades student got.

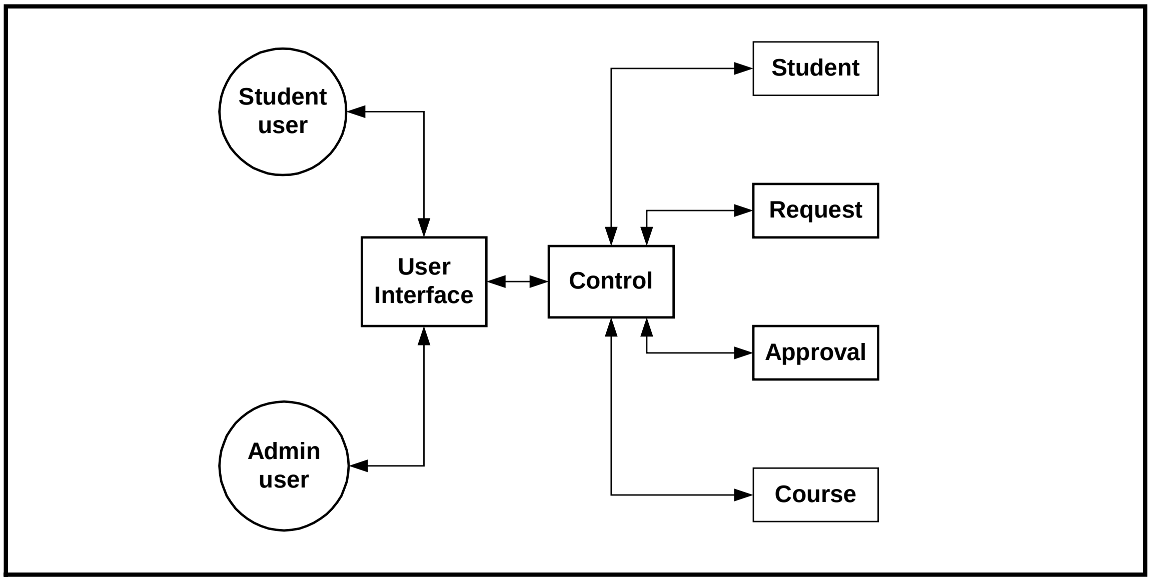


Figure 3.1 INCA modules.

Request. This module maintains the information related to student course requests. It allows the addition, deletion, and modification of course requests.

Approval. This module maintains the information related to course approval codes. It allocates approval codes to the students.

Control. This module does not maintain any information. It controls the data flow with in the application. It is responsible for deciding the interface to be presented to the user, and the processing of client requests.

User Interface. This module presents the information to the users. It accepts different user input, and displays appropriate output.

The users interact with the user interface module. The user interface module communicates the user input to control module. The control module communicates with the student, course, request, and approval modules. The control module acts at the intermediary between user interface module and rest of the modules. It passes the information back and forth. Student, Course, Request, and approval module also interact with each other. Figure 3.1 shows the six major modules of INCA and communication between them.

INCA modules define the package structure for the application. INCA consists of following Java packages:

- edu.hawaii.courseapp.course
- edu.hawaii.courseapp.student
- edu.hawaii.courseapp.request
- edu.hawaii.courseapp.approval
- edu.hawaii.courseapp.control
- All user interface module related files are kept in the directory named docroot.

3.2.2 n-tier architecture

INCA modules are implemented as an n-tier system using J2EE technology (see figure 3.2). INCA has three server-side tiers, namely, web tier, application tier (a.k.a. EJB tier) and database tier (a.k.a. EIS tier). The client requests are received at the web tier. The web tier serves pages to the client by combining together formatting information in the JSP page with the dynamic content provided from the EJB tier. The EJB tier consists of EJBs that implement the business logic of the application. The EJB tier receives the requests from the web tier, applies the business logics, interacts with the persistent storage in the EIS tier, and finally communicates the results back to the web tier. The EIS tier consists of a relational database system that contains a set of tables that implement the persistent storage for the information in INCA.

J2EE technology defines entity and session beans to manage the data and JSP pages to do the presentation. INCA Course, Request, Approval, and Student modules are implemented using entity beans and session beans. The control module is split into EJB tier and Web tier. The user interface module is implemented using JSP pages.

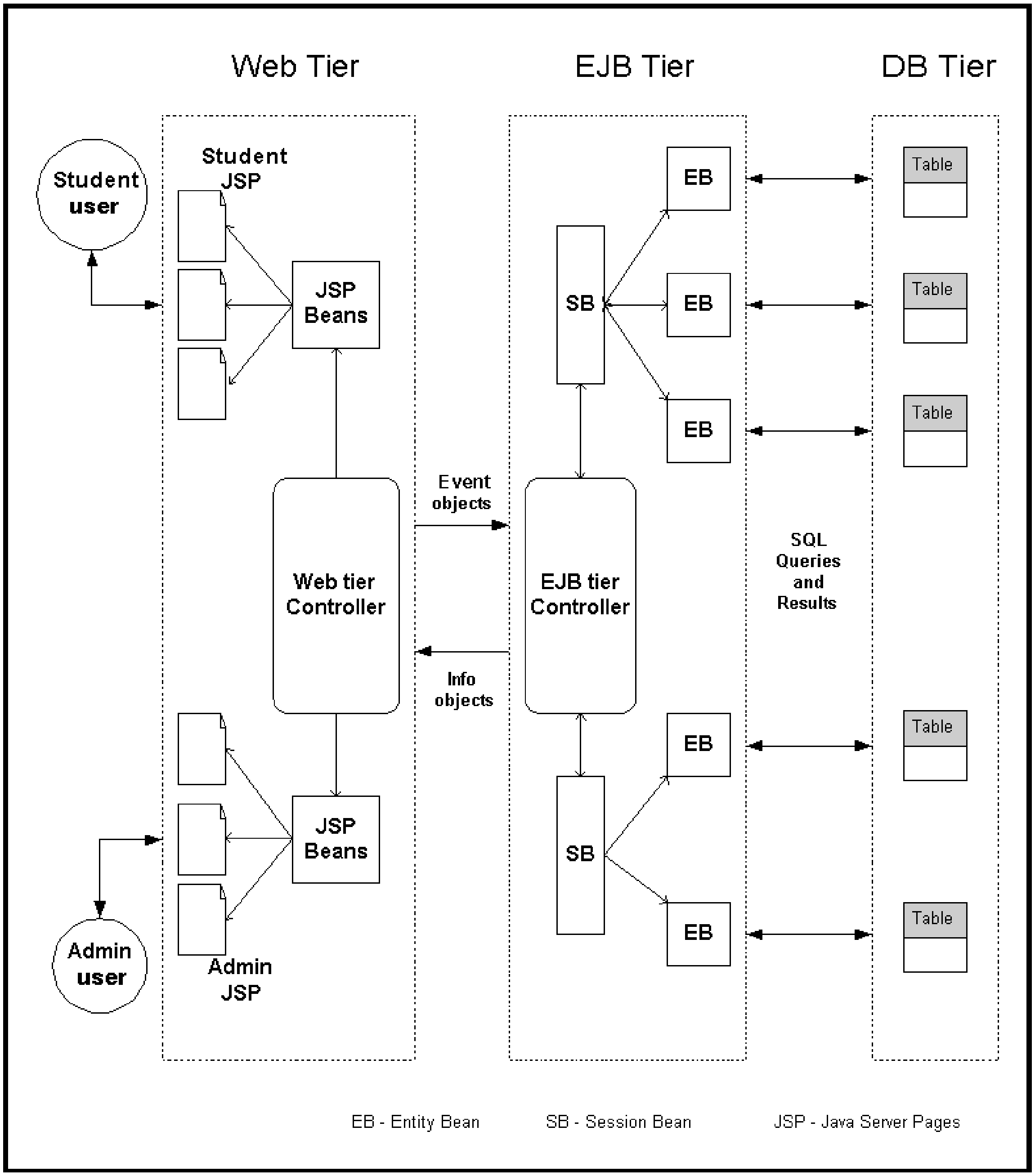


Figure 3.2 INCA tiers and communication between them

INCA modules to n-tier architecture mapping is responsible for further sub-structuring of packages. Each module has an “ejb” and “model” sub-package. The ejb sub-package contains the entity and session bean classes. The model sub-package consists of model and info object classes. Utility classes for each module are contained in util sub-package and Unit test classes for each module are contained in test sub-package. The control module further has a web sub-package that contains the controller classes in Web tier. Appendix A shows the detailed package structure.

3.3 The web tier design

The responsibility of web tier is to provide dynamic and interactive content to browser-based clients. All pages to be presented to the users are packaged inside a web application. A web application is a collection of HTML/XML documents, web components (servlets, JSP pages), and other resources like images, utility classes in archived format known as Web Archive (WAR) file. All web applications run inside a web container, of a web server, which provides runtime environment for the web application. Sun J2EE-RI uses Apache Jakarta Tomcat as a web server. A web server works with an EJB server that may or may not be located on the same machine.

JSP pages

JSP pages define the look-and-feel of the INCA to the user. Java Server Pages (JSP) is a kind of server-side scripting language used for dynamic content generation. JSP pages contains traditional HTML along with embedded Java code and JSP markup tags that allows a web-page designer to access data from Java code running of server. When the page is requested by a user and processed by HTTP server, the HTML portion of the page is passed straight through. The code portion of the page, however, is executed at the time request is received, and the dynamic content generated by the code is spliced into the page before it is sent to the user. JSPBeans provide the dynamic content to the JSP pages. Appendix A shows the navigation structure for student, course administrator, and system administrators respectively.

JSP beans

All of the information residing in database is kept in model objects in EJB tier and JSP bean objects in the web tier. Whenever a user action results in changes to database, those changes need to be reflected back to models in EJB tier and JSPBean objects in the Web tier.

The JSPBean objects can be classified into two categories depending upon accessibility and life span of the beans: application-level JSPBeans and session-level JSPBeans. The accessibility of a JSPbean determines which pages or parts of INCA can access the bean and its properties. The JSPBean's life span determines how long a particular bean exists before it is no longer accessible to any webpage. Session-level JSPbeans can be accessed by the current request and any subsequent request from the same browser window and have life span limited to life to the user's session. Application-level JSPbeans can be accessed by the current and any future request that is part of INCA and they can exist as long the INCA server is up. INCA SRS [13] discusses different JSPbeans with their scope information.

Web tier Controller

The web-tier controller consists of three main components: main servlet, request processor, and screen flow manager. All HTTP requests are received by the main servlet. The servlet handles data-related operations and navigation among different JSP pages.

The servlet gives the requests to request processor. The request processor transforms it into an event and passes the event to EJB tier for further processing. The results of processing in EJB tier are communicated back to the servlet. Then, servlet passes on the results to screen flow manager. The screen flow manager decides which web page needs to be presented to the user and tells about that page to the servlet. The servlet retrieves the JSP page told by screen flow manager and displays it to the user.

MainServlet

The servlet acts as a front component in INCA. It doesn't do any presentation. It handles HTTP requests and converts the requests into a form that the application understands. The JSP pages act like presentation components. Front components are useful because they provide a single entry point to the application, thus making security, application state, and presentation uniform and easier to maintain. Using servlets as the front component is often known as servlet-centric design. Another possible design approach would be JSP-centric design in which flow is maintained by JSP pages only.

3.4 The EJB tier design

In an n-tier J2EE application, the Enterprise JavaBean (EJB) tier hosts application specific business logic and system level services such as transaction management, concurrency control, and security. In the J2EE programming model, EJB components are a fundamental link between presentation components hosted by web tier and business critical data maintained in database tier. The EJB components consist of entity beans, session beans, and utility classes packaged together in JAR (JAva Archive) file format.

Entity Beans

Entities represent data in database, so changes to an entity bean result in changes to the database. Entity beans provide an object-oriented interface that makes it easier for developers to create, modify, and delete data from the database. Representing data as entity beans makes development easier and more cost effective. However, the changes in the state of entity beans must be synchronized with the data in the database.

The process of coordinating the data represented by a bean instance with the database is called persistence. There are two types of entity beans, distinguished by how they manage persistence. Container-managed beans have their persistence automatically managed by the EJB container. The container knows how a bean instance's fields map to the database and automatically takes care of inserting, updating, and deleting the data associated with entities in the database. Beans using bean managed persistence do all this work explicitly; the bean developer must write the code to manipulate the database. INCA used container-managed entity beans, as they are the simplest to develop and they allowed the developers to focus on business logics, delegating the responsibility of persistence to the EJB container.

Session Beans

A session bean is not persistent like an entity bean; nothing in a session beans maps directly into a database or is stored in sessions. Session beans are useful for describing interactions between other beans or for implementing particular tasks.

Session beans can be either stateful or stateless. Stateful session beans maintain conversational state when used by a client. Conversational state is not written to a database; it's state that is kept in memory while a client uses a session. On the other hand, Stateless session beans do not maintain any conversational state. Each method is completely independent and uses only data passed in its parameters. INCA SRS [13] lists different sessions beans, their type (stateful or stateless), and the entity beans managed by them.

Models

Models hold the same data in the EJB tier as tables hold in database tier. The models are managed by a set of session beans. The session beans communicate with the lower level entity beans, which encapsulate the access to the underlying persistent storage. INCA SRS [13] lists different models objects and the session beans managing them.

EJB tier Controller

There are separate controller classes for handling student and administrator pages. Both controller classes are implemented as session EJBs. They create the remaining session beans and initialize them. The EJB tier controller classes handle all the events and communicate the processing results back to their peer web tier controller classes.

3.5 The database tier design

INCA uses relational databases in the database tier. The data is represented in the form of tables with rows and columns. Each table has a primary key and joins with other tables using foreign keys. SQL queries are used to create tables, insert, delete, and update data in tables, and perform the queries on the data. The tables are normalized and there is no data redundancy. INCA SRS [13] lists different database tables and their internal format.

3.6 Communication among tiers

INCA has a client tier and three server-side tiers. Clients communicate with the web tier using HTTP request objects. The web tier controller takes these request objects, converts them into event objects, and passes these event objects to the EJB tier. The EJB tier retrieves the desired information from the event objects and invokes different methods in session beans. The session beans pass on processing information to entity beans. The entity beans use SQL queries to communicate with database tier. The results of processing are communicated back by the EJB tier to web tier using info objects. INCA SRS [13] lists the event and info objects used by INCA.

3.7 INCA security

INCA addresses security by providing proper authentication and access control measures. The authentication problem deals with ensuring that users are who they assert themselves to be. This is to avoid:

1. Non-UH students from getting into the system just to play around.
2. ICS students spoofing other ICS students.
3. Students hacking in as administrators.

INCA authentication procedure is automated and is listed as follows:

1. When a student registers with the system by providing their first name, last name, SSN, and uhunix id, the system does a remote telnet to uhunix and fingers the ITS account name. This validates the ITS account exists.
2. The system then does a lookup in the internal student database (originally obtained from ISIS) by SSN. If a match is found on the SSN, and if the first and

last names also match, then the registration data is bound to that record automatically.

3. Administrative SSN and initial passwords are supplied as command line options when the system is started. They are not stored in any data file or in the source code.

In order to address the access control issues, INCA uses `SecurityAdapter` class to filter out unauthorized access. So, every time a student requests a page, the system looks at the URL requested and checks whether the requesting user has privilege to access that URL or not.

3.8 INCA design patterns

INCA uses the following design patterns (See [10] for details):

1. **Front Controller**. Provides a centralized controller for managing the handling of a request. It simplifies the implementation and maintenance of user interface presentation and workflow.

2. **Session Façade**. Use of a session façade to encapsulate the entity beans. Entity beans from the business tier are not exposed to the clients in another tier.

3. **Model-View-Controller**. Present different views based on the same data model. Facilitate maintenance, extensibility, flexibility, and encapsulation by decoupling software layers from one another.

Chapter 4. INCA Implementation

INCA is implemented using the Java programming language and J2EE technology. The J2EE technology bundles together lots of Java technologies such as Servlets, EJB, JSP, JNDI, JavaMail, and JDBC. INCA uses all of these technologies along with many other tools and technologies. INCA uses Sun Microsystems J2EE-RI (J2EE Reference Implementation) for the packaging and deployment of INCA. Apache Jakarta's Ant is used for compiling and building the system. JUnit and HTTPUnit open-source frameworks are used for writing unit tests. JDOM API is used for XML manipulation. Log4J API is used for logging. CVS is used for multiple developer coordination and source code version control.

This chapter describes the implementation of INCA. I will discuss the implementation of the three tiers, namely, web tier, EJB tier and database tier first. Then, I will describe how all the source code is packaged and deployed on to the J2EE-RI server. Finally, I will show other implementation details such as the build and test mechanism, the logging mechanism, and the data import and export mechanisms.

The implementation of three tiers will be described with code examples. The code examples are taken from important classes of the system. The code examples along with description illustrate the control flow through the system. Comments and Variable declarations are omitted in code examples to keep them short. Only important statements in the important methods are written and remaining code is indicated with notation "…". The log statements from the actual code are omitted too. Knowledge of Java and its APIs is assumed.

4.1 Web tier implementation

Entering into the system. The web tier consists of JSP pages, servlets, JSP beans, controller classes, and some utility classes. The control begins at `MainServlet.` It acts as an entry point into the system. It receives all user requests and responds to those requests. It creates `ScreenFlowManager` and `RequestProcessor` class objects when its `init()` method is called and initializes them by calling their `init()` methods. Every time user requests a page, the servlet forwards the request to `ScreenFlowManager` to get the next screen to be presented to the user. It also forwards the request to `RequestProcessor` for processing the request.

```
public class MainServlet extends HttpServlet {
  public void doPost(HttpServletRequest request, HttpServletResponse
      response) throws IOException, ServletException {
    ...
    getScreenFlowManager().getNextScreen(request);
    getRequestProcessor().processRequest(request);
    ...
  }
  ...
}
```

Deciding about right JSP page. When `ScreenFlowManager` is initialized, it loads the `requestMappings`.xml file into an `urlMappings` hashmap with the help of `ScreenFlowXmlDAO` class. Each URL mapping entry in the XML file maps a URL to a JSP page and optional request and flow handlers. Each URL mapping entry in stored in an `URLMapping` object. All these objects are stored in `urlMappings` hashmap keyed by the URL. For every incoming HTTP request, `ScreenFlowManager` extracts the requested URL from the HTTP request object and gets the corresponding `URLMapping` object from `urlMappings` HashMap. Then, `ScreenFlowManager` extracts the flow handler class name from the `URLMapping` object and invokes the `processFlow()` method of the handler class. Most URLs have an associated flow handler, which selects the page to be presented depending upon the input. All flow handlers implement `FlowHandler` interface and are named as `XXXFlowHandler`. For example, `StudentRegisterFlowHandler` decides whether successful login or an error page to be presented to the user.

```
public class ScreenFlowManager {
  public void init(ServletContext context) {
    ...
    requestMappingsURL = context.getRealPath(
                        “/WEB-INF/xml/requestMappings.xml”).toString();
    urlMappings =
            ScreenFlowXmlDAO.loadRequestMappings(requestMappingsURL);
  }
  public void getNextScreen(HttpServletRequest request) {
    String selectedURL = request.getPathInfo();
    URLMapping  urlMapping = getURLMapping(selectedURL);
    ...
    FlowHandler handler = null;
    String flowHandlerString = urlMapping.getFlowHandler();
    handler = (FlowHandler)getClass().getClassLoader().loadClass
                   (flowHandlerString). newInstance();
    handler.doStart(request);
    String flowResult = handler.processFlow(request);
    handler.doEnd(request);
  }
  ...
}
```

Processing request at web tier and passing it to EJB tier. When `RequestProcessor` is initialized, it creates the `RequestToEventTranslator` object. Whenever a HTTP request is received, `RequestProcessor` asks

`RequestToEventTranslator` to convert the request to an event. Before converting it to an event, `RequestToEventTranslator` looks into `URLMapping` object and extracts the request handler, if any. If there is a request handler class, it passes the request to the handler class to get an event object.

All request handlers are named as `XXXRequestHandler`. They extend the `RequestHandlerSupport` abstract class, which implements the `RequestHandler` interface. The interface defines a `processRequest()` method that returns the event object having the `CourseappEvent` interface.

Once `RequestProcessor` gets the event, it handles over the event object to `AdminControllerWebImpl` or `StudentControllerWebImpl` depending upon the type of event. `XXXContollerWebImpl` are web-tier controller classes. These classes pass on the event object to their corresponding `XXXController` classes in the EJB tier.

```
public class RequestProcessor {
  public void init(ServletContext context) {
    ...
    eventTranslator = new RequestToEventTranslator(context,
                                                   screenManager);
  }
  public void processRequest(HttpServletRequest request) {
    ...
    event = eventTranslator.processRequest(request);
  }
}
```

All event objects are named as `XXXEvent` and they extend `CourseAppEventAdapter` abstract class, which implements `CourseAppEvent` interface. `CourseAppEvent` extends `Java.io.Serializable` as these objects are passed among the tiers.

Updating web tier components. Suppose the event results in some changes to the database. The same changes are reflected back in the model objects in the EJB tier. The list of models that have been changed is communicated back to Web tier controller classes by the EJB tier controller classes. The web tier controller classes pass information about the updated model list to `RequestProcessor`. The `RequestProcessor` then calls the `notifyListeners()` method of `JSPBeanManager` class.

`JSPBeanManager` is responsible for managing all JSPBeans and synchronizing their state with their corresponding model objects in the EJB tier. The `JSPBeanManager` extends the `JSPBeanUpdateNotifier` class, which is used for adding listeners to the model objects. `JSPBeanUpdateNotifier` also accepts the updated model list and notifies the JSPBean classes to update themselves using the `performUpdate()` method.

All JSPBean classes are named `XXXJSPBean; the` `JSPBeanManager` class initializes them. All of the JSPBeans implement `JSPBeanUpdateListener` interface. The interface defines a method called `performUpdate()`, which allows the JSPBeans to update themselves according changes that took place in corresponding Model objects. All `XXXJSPBean` objects store data in info objects. The info objects are of the form `XXXInfo` and provide two important functionalities. They are used for passing information between EJB and Web tier. They are also used as a storage mechanism for JSPBeans.

Each JSPBean class uses `JSPBeanManager` inside the `performUpdate()` method to access model objects. The update sequence is as follows. JSPBeans contacts `JSPBeanManager`. `JSPBeanManager` calls the controller classes in webtier (`AdminControllerWebImpl`, or `StudentControllerWebImpl`). The controller classes in Web tier call controller classes in EJB tier (`AdminController` or `StudentController`). The contoller classes in EJB tier contact session beans (`XXXMgr` classes), which further contacts model objects (`XXXModel`) to get the updates. The results are passed back to JSPBeans in a collections data structure.

Presenting the right JSP page. Once the request has been processed and data structures at Web tier have been updated, `RequestProcessor` transfers the control back to `MainServlet`. `MainServlet` looks at processing results, gets the right JSP page name from `ScreenFlowManager`, and retrives that JSP page. The servlet container removes all the JSP tags and inserts the data from Web tier data structures. Once JSP page is converted to a HTML page, it is passed to the client browser.

4.2 EJB tier implementation

Entering into the EJB tier. EJB tier comprises of session beans, entity beans, model objects, controller, and utility classes. Control enters into EJB tier whenever an event is transferred from the Web tier for handling. The `AdminContollerWebImpl` finds the home interface of the `AdminControllerEJB` using JNDI. The following code illustrates that.

```
public static AdminControllerHome getAdminControllerEJBHome() throws
NamingException {
   Context  initial  = JndiUtil.getInitalContext();
```

```
    Object objref = initial.lookup(JNDINames.ADMINCONTROLLER_EJBHOME);
    return (AdminControllerHome)PortableRemoteObject.narrow(objref,
              AdminControllerHome.class)
}
```

Then, it creates the remote interface `AdminController` using the `create` method on Home interface. It hands over the event to the remote interface that is implemented by `AdminControllerEJB`. `AdminControllerEJB` handles all the events in the `handleEvent` method. `handleEvent` looks at the type of event and processes it accordingly. The result of processing is stored in `processResult` variable. The `handleEvent` method returns the list of updated models to the Webtier using the `JSPBeanUpdateManager getUpdateModels(event)` method.

The controllerEJBs are session beans. So, They have a remote interface, a home interface, and an EJB implementation. Entity beans also have a primary key class component associated. A list of entity bean and session bean classes is shown in Appendix A.

It is useful to look at the handling of one event in order to illustrate the components of EJB tier other than controller. Let us take a look at how the student registration event is handled.

`StudentRegister.jsp` page allows the students to register with INCA. The page consists of an HTML form that accepts four fields: Student first name, Student last name, Student ID, and student ITS account. When student submits the form, the request is received by the servlet. Then, Webtier processing is done as described in the previous section. The `StudentRegisterEvent` class extracts all four fields from the request object and places it in the event object. The event object is passed to the EJBtier. At the

EJB tier, AdminControllerEJB handles the student register event. The class extracts all the fields from the event object and then authenticates the user using AuthenticationMgr class. Once authenticated, the controller EJB checks that whether student exists in database.

Handling tasks with session beans. This is done using StudentsMgr class. StudentsMgr provides access to the data of all the students and StudentMgr provides access to the data of a single student. If student exists, StudentMgr is used to get the StudentStatusModel and then StudentInfo object for the student. The Info object typically represents a row of a table in the database tier. They are used to transfer values from models to JSPBeans. Info objects also carry an operation flag that indicates the kind of operation performed on model objects (add, delete, or update) and JSP bean objects in the web tier must perform same operation. Whenever a model object is created, or updated, a corresponding info object is created and stored in a updatedInfos collection. This collection is sent to JSPBean when it requests the updates from the model objects. All entity and session beans define call back methods.

4.3 Database tier implementation

There are two ways to create and insert the data into the tables. The first way is to use SQL scripts and stand-alone clients. The second way is to use the J2EE-RI deployment tool. INCA used the later approach for actual data insertion. The J2EE-RI deployment tool creates the tables when the application is deployed. In addition, it automatically generates the necessary SQL scripts by looking at the entity bean specifications. Once the application is deployed and tables are created, INCA uses a web-based interface to read XML data files and write the data to database using entity beans. We used the first approach during the development of INCA as the data needs to be quickly inserted and tables need to be frequently created and destroyed.

The following tables were made: Course, Consent, ApprovalCode, ExcludedSection, Grade, Qualification, Request, Requisite, SectionTable, and Student. INCA SRS [13] lists the tables and their field structure in detail.

4.4 Initialization and communication among tiers

The data in the web tier, the EJB tier, and the database tier must be kept consistent. If the actions of a user result in changes to the data, the changes are propagated from the web tier to the EJB tier and then to the database tier. The database tier maintains the data in tables. The EJB tier maintains the data in model objects and the web tier keeps all the data in JSP bean objects. The data is updated in two phases. In the first phase, web->ejb->database->ejb->web, only models in ejb tier are updated. In the next phase, web->ejb->web, JSP bean objects are updated.

INCA initialization

After INCA is deployed for the first time, when the first request is received,

1. The `MainServlet` creates `RequestProcessor` and `ScreenFlowManager` objects. `ScreenFlowManager` loads all the `urlMappings` from `requestMappings.xml` into the `urlMappings` HashMap. `RequestProcessor` creates the `RequestToEventTranslator` object.

2. `MainServlet` loads the `JSPBeanManager` class. `JSPBeanManager` further creates `AdminControllerWebImpl` and `StudentControllerWebImpl`. These classes invoke the create methods of `AdminController` and `StudentController` EJBs respectively. For create methods in the home interface, ejbcreate methods in

corresponding EJB classes are invoked. The AdminControllerEJB ejbcreate method creates JSPBeanUpdateManager. Then, it creates all the session beans. All the session beans also have ejbcreate methods and they create models and entity beans in those. Models hold the information in data structures and these data structures get created when models are created. Now, The control is transferred back to JSPBeanManager that creates all application level and session level JSPbeans now. JSPBeans add themselves as the listeners to their corresponding model objects. Then, JSPBeanManager invokes the session bean classes corresponding to every JSPBean and invoke the updateAll method of those session beans. The method loads everything in database tables into model objects using the entity bean finder functions. Then, JSPBeanManager invokes the performUpdate methods on JSPBeans. These methods retrieve the updated models as a collection and then JSPBeans update themselves from that collection.

3. MainServlet creates the SecurityAdapter class.

4. MainServlet processes the request by giving it to RequestProcessor. RequestProcessor hands over the request to RequestToEventTranslator for processing. RequestToEventTranslator looks at the request object and extracts the requested URL. Then, it extracts the URLMapping object corresponding to that URL from the hashmap. From that, it gets the request handler and from request handler it gets the event object. As a result of processing the request, the processResult variable gets set. ScreenFlowManager uses this processResult in the next step.

5. `MainServlet` handles over the request to `ScreenFlowManager` to get the next screen. `ScreenFlowManager` looks at the requested URL and gets the page to be presented from the corresponding `URLMapping` object.

6. `MainServlet` forwards a new page to the user.

4.5 Packaging and deployment.

J2EE defines an .ear (Enterprise archive) file, which is a JAR file for packaging Enterprise JavaBeans JAR files and web component JAR files (.war files) together into one complete deployment called a J2EE application. A J2EE application has an application descriptor that points to the EJB and web component JAR modules. It also has a runtime deployment descriptor that provides information such as JNDI names. Runtime deployment descriptors are specific to an EJB container. All the EJB Jar modules also have their deployment descriptors. These deployment descriptors provide declarative information about enterprise beans, such as transaction requirements, security requirements, persistent fields, and so on.

As Figure 4.1 shows, `inca.ear` consists of ejbjar files and war files. These files are specified in `application.xml` file. `sun-j2ee-ri.xml` is Sun Microsystems J2EE-RI specific deployment descriptor. All the ejbjar files bundle the entity and session beans along with their corresponding model and info objects. All the ejbjar files have their deployment descriptors `ejb-jar.xml`. The war component has a file `web.xml` that acts as a deployment descriptor for the web components. The `requestmappings.xml` file is also packaged in the web component. The component includes JSPbean classes, webtier controller classes, HTML and JSP pages, and the image files.

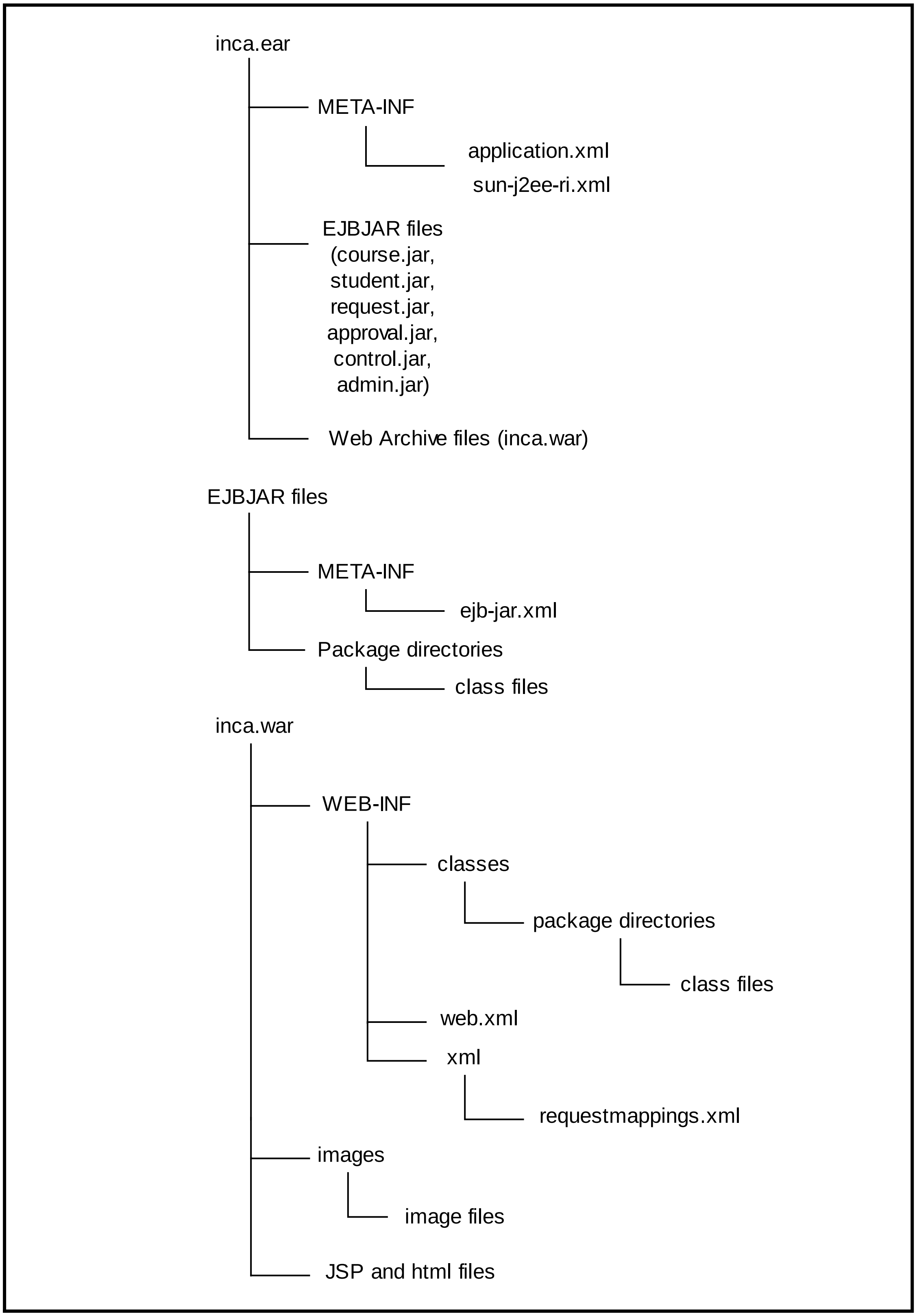


Figure 4.1 Contents of INCA ear file

4.6 Build and test mechanism

INCA used two different approaches to building and deploying the application: The first approach involves Jakarta Ant, and the second involves the J2EE-RI deployment tool.

Jakarta Ant is similar to make utility of UNIX. Jakarta Ant uses build.xml file, which is analogous to Makefile in the make system. One can specify different targets in the build.xml file that are executed by Ant. The INCA build.xml file allowed compiling of source files, building of Javadocs, building of ejbjars and war components, building of ear file, verifying of ear file, and deployment and undeployment of the application on J2EE server. A hierarchical build structure was used with one build.xml for every single component and a top-level build.xml that invokes the targets in component-level build.xml files.

The second approach used Kawa IDE for compiling files and Sun J2EE-RI deployment tool for building ejbjar, war, and ear components, verifying, deploying and undeploying the application from J2EE server.

INCA used Junit for the testing of EJB components and HttpUnit for the testing of webpages. The Junit targets were built into the Ant build files. All the test classes of the form TestXXX were used to implement unit test using Junit framework.

The builds and tests were performed daily. One developer was responsible for daily build and smoke tests. All the developers committed their code onto CVS server by a certain time and the build manager checked out all files at that time. Then, the build manager made a fresh build daily and tested it.

4.7 Logging mechanism

INCA used log4J for logging purposes. Three levels of logging were implemented: minimum, brief, and detailed. Also, two modes of logging were implemented: developer and production. The message entity bean was responsible for logging purposes. Log class used message entity bean.

4.8 Data import and export mechanisms

Data for INCA comes from various sources. Student data comes from ISIS in a plain text file with fields formatted in columns. Approval data comes from PA’E in a plain text file with fields formatted in columns. Course offered in the semester are specified in an Excel spreadsheet. Course requisite information is available over UH website. All this data is converted into XML format first. Utility programs convert the student and approval data into XML format. Course and Requisite XML files are need to be manually constructed. Then, JDOM API is used fro reading data from XML files into the database using entity beans. Data can also be exported from the database to XML files.

Chapter 5. Evaluation and Results

This chapter describes the evaluation and results of INCA. I evaluated INCA from three different perspectives: the course administrator, the student, and the department. I did an email analysis to investigate the hypothesis that INCA reduces the administrative overhead involved in the approval code allocation process. I conducted a student user survey to investigate the hypothesis that INCA improves the predictability and the understandability of students. The operational definitions of predictability and understandability are stated in Table 1.2. To investigate the hypothesis that INCA provides valuable data to the department that can be used for feedback and planning on their curriculum, I analyzed the INCA database and came up with examples of information that could be useful to department. In the following sections, I will describe the administrator, student, and department evaluation and results. Then, I will relate the results to the hypothesis. The term administrator refers to course administrator, as opposed to the INCA system administrator, in the following sections.

5.1 Administrator evaluation and results

INCA was used for the first time during Fall 2001 registration and for a second time during Spring 2002 registration. The manual method of approval code allocation was used during Spring 2001 registration. The same ICS course administrator, who distributed approval codes manually for Spring 2001 registration, used INCA to allocate approval codes for Fall 2001 and Spring 2002 registrations. Anecdotally, the course administrator seemed happy and satisfied with the system and commented that INCA is very fast, easy-to-use, and performing very well.

Students used the uhmics@hawaii.edu account to inform the administrator of their course requests and queries during Spring 2001 semester and to inform the administrator of their questions and problems with INCA during Fall 2001 and Spring 2002 semester. Students also used this account to ask the administrator to correct their records in case the student information listed in INCA was incorrect. In order to perform administrator evaluation, I did an analysis of the emails sent to uhmics@hawaii.edu during Spring 2001 and Spring 2002 semesters.

5.1.1 Email analysis

The email analysis helped me investigate the hypothesis regarding administrators. According to the hypothesis, INCA should reduce the administrative overhead involved in reading and responding to the emails. Another goal of email analysis was to look into student problems to uncover ways in which INCA could be improved.

First, let us look at the typical registration period for Spring and Fall semesters. Approval code requests start two weeks prior to the registration begins over PA’E. The registration period for Spring semesters over PA’E lasts around 8 weeks. The registration period for Fall semester over PA’E lasts around 20 weeks and it overlaps with summer registration. The following paragraphs describe the Spring and Fall registration periods in detail.

For Spring registration, continuing students can register during their scheduled call-in times for the first two weeks. Registration occurs on the basis of seniority during this period. Following this, non-scheduled registration occurs during which anybody can register. Registration gets cancelled in case students fail to provide their payments by the payment deadline at the end of the third week. After the payment deadline, registration closes and reopens after three weeks. The first two days are scheduled for call-in times for new students. Following this, there are four days of non-scheduled call time for all students. After that, late registration period begins for two weeks and the final payment deadline marks the end of Spring registration.

For Fall registration, continuing students can register during their scheduled call-in times for first two weeks. Registration occurs on the basis of seniority during this period. Then, non-scheduled registration occurs during the next three months during which any continuing student can register. The registration ends at the payment deadline and registration closes for two weeks. Once the registration reopens, new students get to register for one week during their scheduled call-in times. This is followed by one week of non-scheduled call-in times for all students. After that, late registration period lasts for two weeks and final payment deadline marks the end of Fall registration.

I cannot compare Spring 2001 (when INCA was not used) with Fall 2001 (when INCA was used for first time) as the registration periods differ by 12 weeks. Also, Fall 2001 overlapped with Summer 2001 registration. I compared Spring 2001 with Spring 2002 (when INCA was used for second time). Out of a total of 1517 emails, 924 emails were received during initial registration period of Spring 2001. A total of 312 emails were received during initial registration period of Spring 2002. I used these emails to perform the comparative analysis as a majority of registration occurs during that period and also due to the time constraints this of thesis. I classified the emails into different categories based on their content. I read all the emails to not only to test the hypothesis that INCA reduces administrative overheads but also to discover the student problems before and after and INCA.

5.1.2 Email analysis methodology

All of the emails sent to uhmics account since 14 November 2000 resided on the UH UNIX servers. The course administrator downloaded emails to her personal computer, but kept a copy of all the emails on UH UNIX systems. These emails are the source of the email data analysis. I also downloaded the emails from UHUNIX system to my personal computer using Netscape Messenger email client. Netscape Messenger stores the emails in a proprietary file format and lets the users to do email operations via a client window.

I analyzed that Netscape file format and wrote a Netscape email file parser. For each message, I extracted the following information from the Netscape file: message number, date and time at which message was sent, and sender information (I assigned an ID to each sender). I imported the information to a Microsoft Excel spreadsheet. I started reading the emails from Netscape Messenger one-by-one. I grouped and regrouped the emails into different categories as I went along. I categorized every email and then recorded the information in the Excel spreadsheet. At the end, I summed up each of the categories. This kind of email analysis helped me in answering the following kinds of questions:

- What are the different types of emails?
- What are the most common emails?
- Which problems existed before INCA was used?
- Which problems got vanished with INCA?
- What new kinds of problems got introduced?
- What is the time period during which most of the emails were sent?

5.1.3 Email analysis results

Table 5.1 lists the email categories and typical email contents during Spring 2001 (when INCA was not used) and Spring 2002 (when INCA was used). The table also provides the information on total number of emails received for a particular category and the percentage constituted by that number. A total of 921 emails were received during initial registration period of Spring 2001 and 312 emails were received during initial registration period of Spring 2002. This amounts to a reduction of almost 67% of emails when INCA was used.

Table 5.1 Typical email contents during Spring 2001 and Spring 2002

Email Category	Typical Email contents	Spring 2001		Spring 2002	
		No.	%	No.	%
Course Requests	Addition, Deletion, Modification, or resending of course approval code requests.	756	82	13	4
Queries	Queries on courses, prerequisites, late grades, and approval codes allocation.	71	8	48	15
Consents	Emails related to consent from the instructor.	12	1	11	4
Concerns	Concern regarding approval code allocation, timely registration and graduation.	13	1	6	2
INCA info correction	Basic information (major, graduating senior, or level) or grades data need correction.	0	0	32	10
INCA problems	Problems, complaints, bug reports, and queries regarding INCA and its usage.	0	0	167	54
Irrelevant	Thanks notes, Duplicate emails, and other emails not relevant for analysis.	68	8	35	11
Total		**921**	**100**	**312**	**100**

Table 5.2 shows timelines for registration for continuing classified students during Spring 2001, Fall 2001, and Spring 2002 registrations. This is the initial registration period and accounts for 70-80% of the registrations.

Table 5.2 Registration timelines

Semester	Approval code request begins	Registration period on PA'E	Payment Deadline
Spring 2001	November 14	November 27 to December 15	December 15
Fall 2001	April 30	April 23 to August 3	August 3
Spring 2002	November 19	December 3 to December 21	December 21

As we can see from the Table 5.2, the approval code requests usually begin two weeks prior to the registration. However, they were delayed by three weeks for Fall registration when INCA was used for first time as INCA was not ready yet.

Figure 5.1 shows the emails received during 5-week period for Spring 2001 registration. The average number of emails received per day was approximately 9. Figure 5.2 shows the emails received during 5-week period for Spring 2002 registration. The average number of emails received per day was approximately 28. So, The number of emails was reduced by almost 67% when INCA was used.

Figure 5.3 and 5.4 show the email distributions of different type of emails when INCA used (Spring 2002) and when INCA was not used (Spring 2001). As we can see from the figures, the big pie of course requests during Spring 2001 nearly vanished in Spring 2002 when INCA was used.

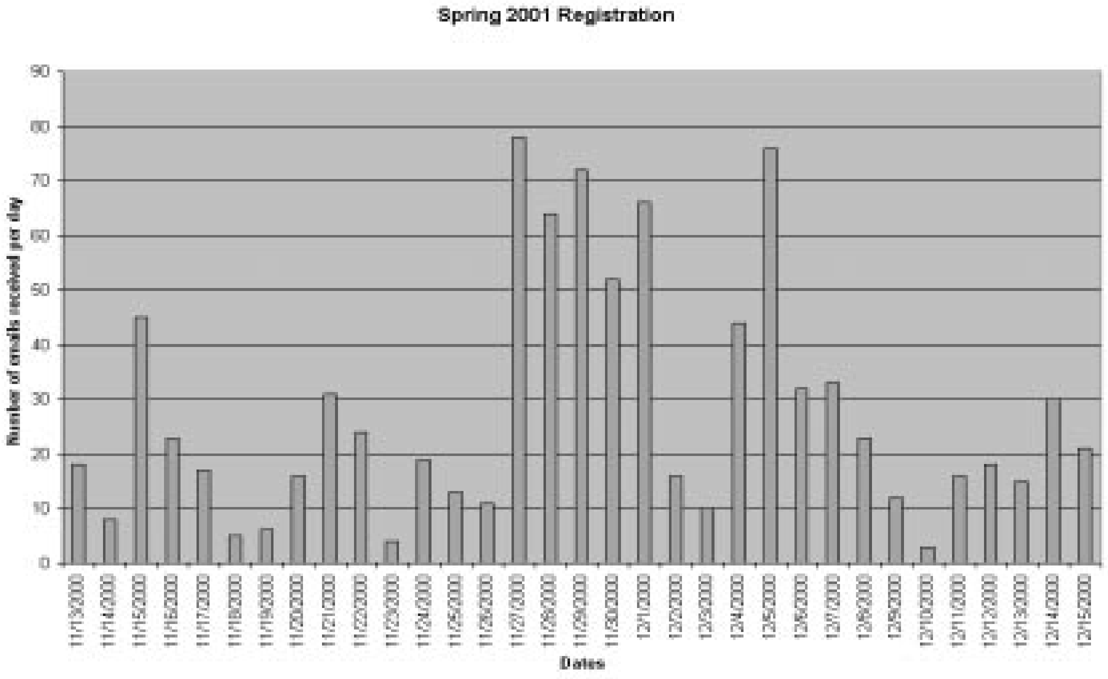


Figure 5.1 Number of emails received per day during Spring 2001 registration

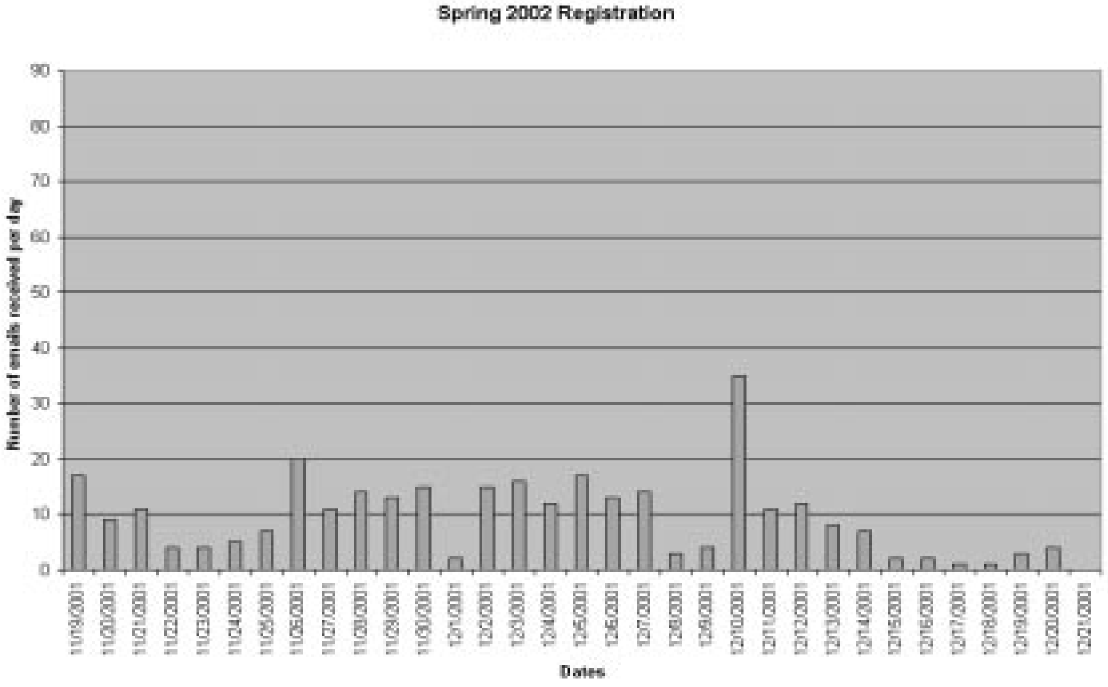


Figure 5.2 Number of emails received per day during Spring 2002 registration

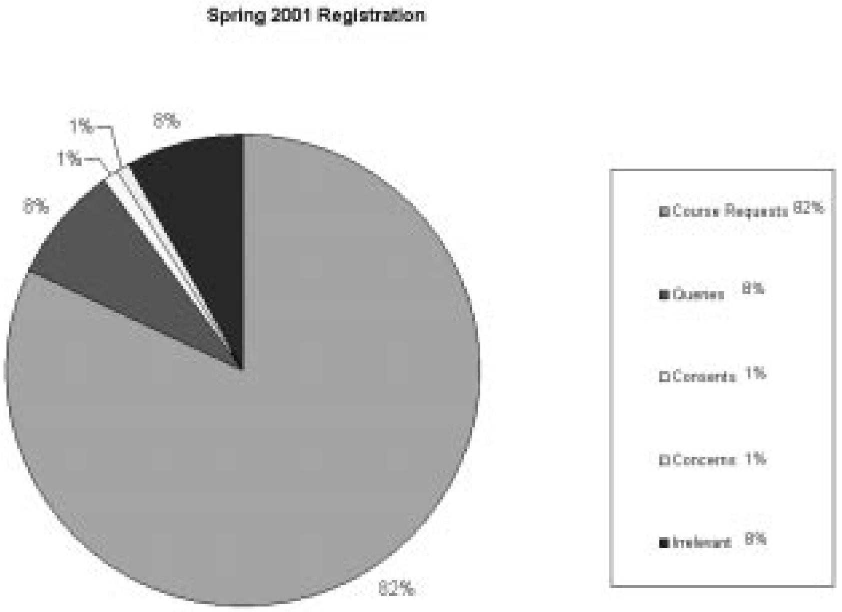


Figure 5.3 Email distributions when INCA was not used

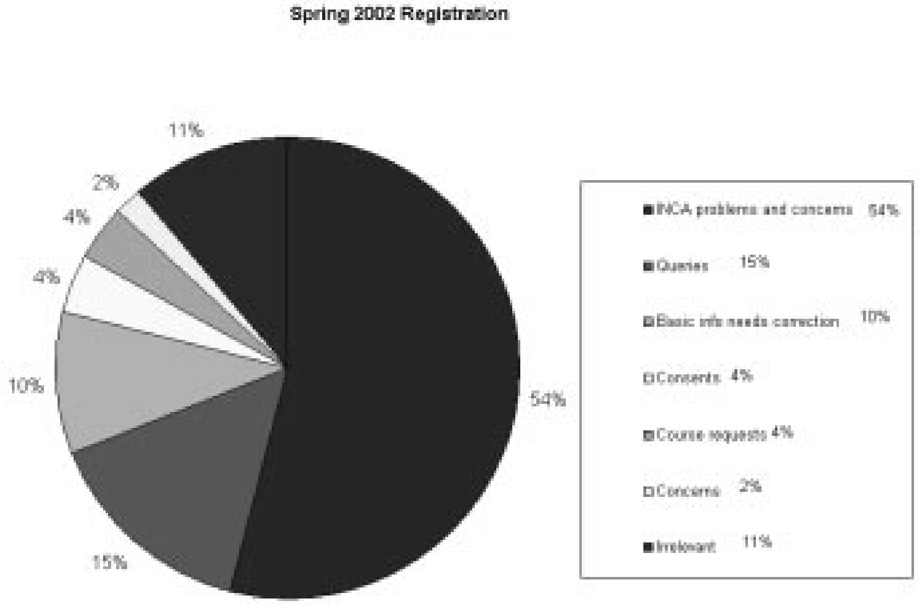


Figure 5.4 Email distributions when INCA was used

5.1.4 Emails analysis results and hypothesis

As the results indicate, there was almost 67% of reduction in the number of emails when INCA was used. The course requests emails (which formed 82% of the total emails) nearly vanished. These were the ones that used to consume a great deal of time for the administrator. For every course request email, administrator needed to log onto ISIS, verify student records, check whether student is meeting all course requisite requirements and then responding appropriately. Moreover, if a student sends an email for deletion or modification of course approval code requests, the administrator needed to manually keep track of those.

With INCA, all those emails vanished. But, with INCA came new responsibilities. In the case of new students, the administrator needs to add the student to the system. In the case of transferring students, the administrator needs to add the transfer grades. Also, the administrator needs to modify basic status information, including graduating senior status, ICS major status, and continuing status for students. However, these emails formed only 10% of the total number of emails and since INCA is very fast and user friendly as compared to ISIS, this job doesn't take much time.

Another job that came with INCA is to respond to queries regarding INCA and its usage and communicating INCA problems and bugs with INCA system administrators. The administrator also needed to add consent of the instructor into the system. Even though this seems like a lot of work, it isn't. This is because the number of emails with INCA is reduced and INCA is much faster than ISIS. Moreover, the administrator doesn't need to check requisite information manually, maintain approval code paper-sheets, and send approval codes by email.

As Figures 5.1 and 5.2 illustrate, the number of emails received per day decreased drastically with INCA. The average number of emails received per day (including week-ends) before INCA was 28. The maximum number of emails approached 78 during certain days. The average number of emails per day (including week-ends) with INCA is 9 with a maximum number of daily emails as 35 and the rest less than 20. Thus, average number of emails received per day is reduced by almost a factor of three. All these numbers indicate that INCA significantly reduces the administrative overhead for approval code allocation. The INCA course administrator confirmed these results.

5.2 Student evaluation and results

The goal of student evaluation was to investigate my hypothesis regarding students. I conducted a user survey in order to do student evaluation. INCA provides the students with real-time information on total seats, total requests that have been made, and their ranking in the queue to help them predict their chances for getting a course based upon their ranking. There was no such concept as *predictability* before INCA. Secondly, INCA helps students understand why or why not they are qualified for a certain course. INCA is designed to improve the *understandability* of course requirements. Without INCA, manual explanations were needed.

5.2.1 User survey

The INCA user survey consisted of six questions. The first question assessed usability. The second and third questions were designed to assess if INCA improves course approval predictability and course requirements understandability. The fourth question assessed student views about the priority point mechanism. Students could report their problems or suggestions in the fifth question. Finally, the sixth question assessed whether students prefer INCA to the old method. The INCA user survey is listed in Appendix B.

5.2.2 User survey methodology

I conducted the user survey on the web, using INCA itself. The survey was anonymous and participation in the survey was voluntary. Students were not given any credit for participating in the survey.

I posted the survey over the INCA website. Students were given an option to participate in the survey before logging out of the system. Each time a student filled out the survey, an email was automatically sent to me containing the responses. There was no time frame on the survey. The survey was up as long as INCA was up. Students could fill out the survey whenever they wanted, could also fill out the survey multiple times. I was able to detect the multiple responses from same student. I used the last response while reporting the results.

As soon as I received an email containing a survey response, I recorded the results of questions 1, 2, 3, 4.1, and 6 in an MS Excel spreadsheet. I used the codebook shown in Table 5.3 to record the results in the spreadsheet. I recorded the comments to questions 4.2 and 5 in a MS Word document. Table 5.4 shows a portion of the Excel spreadsheet used to record the

Table 5.3 Survey codebook

Q1. Usability
Excellent – 5
Very Good – 4
Good – 3
Average – 2
Poor – 1
Q2. Predictability
Extremely useful – 3
Somewhat useful – 2
Not useful at all – 1
No opinion – 0
Q3. Understandability
Extremely effective – 3
Somewhat effective – 2
Not effective – 1
No opinion – 0
Q4.1 Priority point approach
Strongly support it – 3
Prefer it – 2
not support it – 1
No opinion – 0
Q6. Old method or INCA
INCA – 2
Old method – 1
No opinion – 0
Q4.2 Opinion on priority points
got answer – 1 (see Q4.doc)
Q5. Comments or suggestions
got answer – 1 (see Q5.doc)
Other Conventions
En – nth Extra response
F – Final response
N – Null response
EN – Extra null response

survey results. The top row in the spreadsheet is the SUM row. It is used to display the sum of responses for a particular option. Excel updates the sum automatically whenever a new row is inserted. The third row shows the different options for a question. The option details are given in Table 5.3.

Table 5.4 User survey results sheet

SUM	149	17	49	41	22	20	149	58	62	25	4	147	46	67	31	3	147	29	54	58	6	145	86	34	25	54	55
SNO	**Q1**						**Q2**					**Q3**					**Q4.1**					**Q6**				**Q4.2**	**Q5**
		5	**4**	**3**	**2**	**1**		**3**	**2**	**1**	**0**		**3**	**2**	**1**	**0**		**3**	**2**	**1**	**0**		**2**	**1**	**0**		
1		1						1						1				1					1				
2			1						1				1						1				1				
3			1						1					1					1						1		
4			1					1						1				1					1				
5				1					1					1					1						1		
6					1				1					1					1					1			
7	E1				0				0						0					0					0	1	1
8			1						1				1						1						1	1	
9				1					1					1						1					1	1	
10				1						1				1				1							1		
11			1						1					1					1				1			1	1
12				1				1					1						1				1				
13						1				1					1					1				1			
14			1					1					1							1					1		
15				1				1					1					1					1				
16			1					1					1					1					1				
17				1					1						1				1				1			1	1
18				1					1					1						1			1			1	
19			1								1					1		1					1			1	
20	E1				0					0						0				0				0			
21			1						1					1				1					1				
22			1						1					1						1			1			1	
23	F			1				1						1					1				1				
24			1					1						1					1				1				
25				1					1					1				1					1				1
26	NE						N					N					N					N					
27	F					1				1						1				1					1		

…
…
…

5.2.3 User survey results

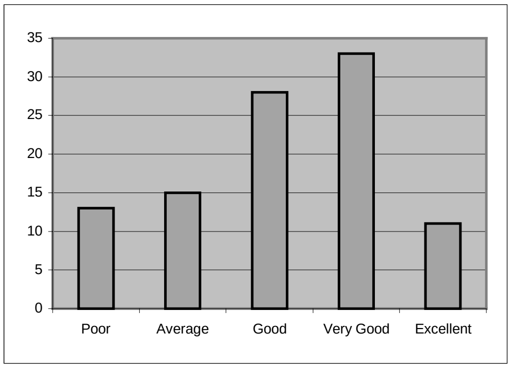
Out of 343 students who logged into the system during one-month period when the survey was up on the INCA website, 149 filled out the survey. This amounts to a response rate of 43%. The results of the user survey are summarized below.

Usability. Please rank the overall quality of INCA in terms of ease-of-use.

- Excellent
- Very Good
- Good
- Average
- Poor

Responses.

Option	Responses	Percentage
Excellent	17	11 %
Very Good	49	33 %
Good	41	28 %
Average	22	15 %
Poor	20	13 %



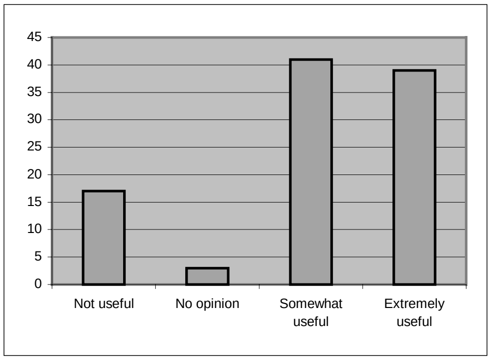
Predictability. INCA provides real-time information on total seats, total requests that have been made, and your ranking in the queue. This is to help you in predicting the chances of getting the approval codes. To what extent was this useful feature?

- It was extremely useful. Now I can predict my chances of getting the approval codes.
- It was somewhat useful. It is nice to know your rank in different courses, but I think I can get the courses I want without it.

- It was not useful at all.
- No opinion. It was not applicable to me

Responses.

Option	Responses	Percentage
Extremely useful	58	39 %
Somewhat useful	62	41 %
Not useful at all	25	17 %
No opinion	4	3 %

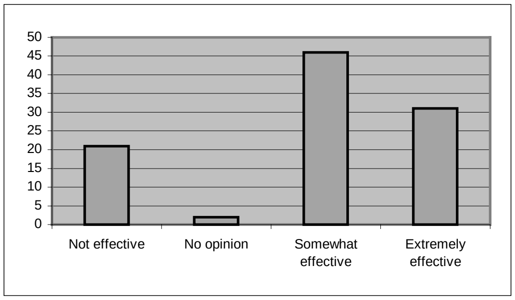


Understandability. Many ICS courses have prerequisites, co-requisites, and concurrent course requirements. INCA tells you why you are qualified or disqualified for a certain course. To what extent was this effective in improving your understanding of different course qualifications?

- It was extremely effective. Now I know why I am qualified or disqualified for certain courses.
- It was somewhat effective. I already knew course qualifications, but it is nice to know them through INCA.
- It was not effective at all.
- No opinion. I don't care about understanding course qualifications.

Responses.

Option Details	Responses	Percentage
Extremely effective	46	31 %
Somewhat effective	67	46 %
Not effective at all	31	21 %
No opinion	3	2 %

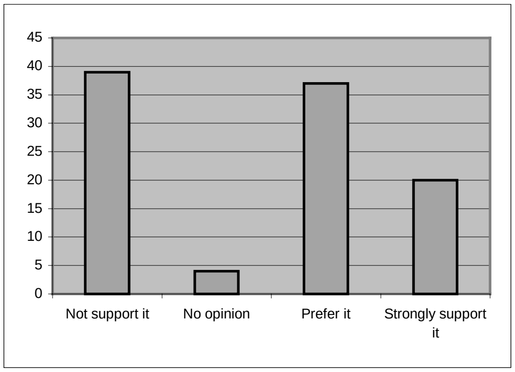


Priority points approach. INCA provides the "priority points" mechanism that enables students with a strong academic background to have higher priority for their future courses. To what extent do you support this approach to course allocation?

- I strongly support it. It should definitely be used next semester.
- I prefer it. But it doesn't matter if it is not used for next semester.
- I don't think it should be used.
- No opinion. I don't care.

Responses.

Option details	Responses	Percentage
Strongly support it	29	20 %
Prefer it	54	37 %
Not support it	58	39 %
No opinion	6	4 %



Opinion of students on priority points approach. INCA prioritizes the students using different rules such as: favor advanced students; favor students with high ICS GPA; favor first and second choices of students; and do not favor repeat attempts. Do you have suggestions for new priority rules or changes to the current priority rules? Should point values be changed?

Responses. I received 54 responses for this question. A list of all the responses is presented in Appendix C. Almost half of the responses are regarding complaints about GPA as a priority rule. Here are some of the student quotes:

- *"I don't agree with the points system. It really segregates the students in an unfair manner. Just because someone gets a better grade than me in class does* *not make it right for them to get first dibs on a class. Why would this even be an option? If this was something like a scholarship or something other than a University derived system it might be okay. This actually angers me that it was even put in."*

- *"Having points for a high GPA is good but I think the penalty for low GPA should be less or none at all."*
- *"What kind of message are we sending out here? You're too stupid to take these courses"*
- *"Prioritizing the students may, in effect, help students who do not need help and hurt students who do need help."*
- *"Not being able to get classes to graduate because of your grades is a discrimination."*

Students also complained about the repeated attempts and withdrawals as priority rules. Almost 10 students recommended that class standing should be used as a priority rule with graduating seniors being given the highest priority, followed by seniors, juniors, sophomores, and freshman.

In a nutshell, students comment that they should not be penalized for something. The rules like GPA, repeated attempts to take a course, and withdrawals from a course penalize the students and decrease their chances for getting the courses. Extremely outstanding students may be rewarded, but students should not be penalized for their GPA. GPA depends upon a lot on different factors. Some students have families. Some work full-time. Some do not learn as quickly. One student commented:

- *“It's very disheartening when I sign up for an approval code before anyone else, and somebody comes in at the last minute, and since they got an A in the prerequisite class, and I got a B, they get in ahead of me, even though I signed up two weeks before they did. Of course, this would be a moot point if the university could pay our professors more, so that we could have more instructors (and more QUALIFIED instructors), and then we could have enough sections of each class to fulfill everyone needs. I think the ics department has less than 20 instructors, and that is a travesty.”*

One student did very nice analysis of all the rules to prove that system heavily favors the students with good academic credentials. A few others also commented that all students should be given equal chances for getting a course.

Comments and suggestions for improvements. If you would like, please tell us about the problems you faced or give some suggestions for improvements in INCA.

Responses. I received a total of 55 responses for this question. A list of all the responses is presented in Appendix C. The responses are mixed in nature. Whereas some students complained about the system and expressed their problems, others appreciated the system.

Some students suggested a help page that describes the INCA process to new students while some others suggested improvements in user interface as instructions does not seem very clear to them. Almost half of the comments were from students who were not able to request a course because they are taking a prerequisite for that course. INCA didn’t have the grades for prerequisites and was not allowing them to register for the classes. They think that there should be some procedure for those students to request the approval codes. One student gave the following comment on timeline for INCA:

- *"I would prefer if all students were able to put in requests a week before registration and INCA released all approval codes the day before registration starts, and from the second day of registration on, it should be first-come first-serve."*

A few students suggested that brief course descriptions for courses like ICS491 should be provided. Some students were concerned about INCA security. Some students reported registration problems.

One student gave the following comment about the approval codes and approval code distribution logic:

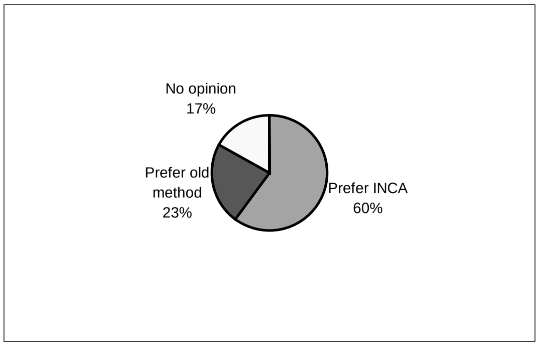
- *"Approval codes only exist to show proof that you have the proper prerequisites to take a course, not to show priority to certain students. Registration through pa'e handles student priority in attaining classes. The current system uses the approval codes in a way they were not meant to be, and it's also kind of redundant to have 2 methods (the current approval code system, and registration through pa'e), that do the same thing (handle student priority in attaining courses). All in all, I prefer INCA to having to obtain approval codes in person, or by email. INCA can show us on one page, in a compact format, what classes we currently qualify for, and it makes it easier for us to see a graphical representation of our past courses, and what courses we may take next semester. I only have a problem with the logic used with the assignment of approval codes. Everything else about INCA is very good, and whoever (whether it was one* *person, or a group of people) created INCA in its current form, deserves my gratitude, and a pat on the back for a job well done!"*

INCA or old method. Do you prefer INCA or the old method of sending emails to uhmics should be used for distribution of approval codes?

- ○ I prefer INCA
- ○ I would prefer sending emails to uhmics@hawaii.edu to request the courses
- ○ No opinion, I don't care

Responses.

Option	Responses	Percentage
Prefer INCA	86	60 %
Prefer old method	34	23 %
No opinion	25	17 %



5.2.4 User survey results and hypothesis

The results of the user survey indicate that INCA significantly improves the students predictability of course requests and understandability of departmental policies for course approval.

39% of students believe INCA is extremely useful for them for predicting their chances of getting the approval codes. 41% of the students believe INCA to be somewhat useful in predicting the chances. This amounts to a total of 80% of the responses in favor of INCA in terms of improving the predictability.

31% of students believe INCA is extremely effective for them for understanding the course qualifications. 46% of the students believe INCA to be somewhat effective in improving the understanding of students. This amounts to a total of 77% of responses in favor of INCA in terms of improving the understandability.

In summary, more than three quarters of the students believe that INCA improves their predictability and understandability and this confirms my hypothesis that INCA improves the predictability and understandability of students.

5.3 Department evaluation

The goal of the department evaluation was to come up with the data valuable to the departments for curriculum planning. I looked into the data residing on INCA server and extracted out the information useful to the departments such as the total requests received for a course and number of seats for that course. I also gathered student statistics such as the number of ICS majors, the number of graduating seniors, the number of continuing students, and so on.

5.3.1 Database analysis

The INCA database analysis helped me in validating the hypothesis that INCA provides valuable data to the departments that can be used on feedback and planning on their curriculum. The INCA database stores the data regarding students, their grades, their requests and the approval codes granted to them. It also maintains the course scheduling information along with prerequisites and co-requisite requirements for different courses. I examined the INCA database and wrote SQL scripts to extract different kinds of information from the INCA database.

5.3.2 Database analysis results

I analyzed the INCA database and came up with some information useful to the departments. INCA provides useful course demand information and student statistics. The course demand information is summarized in Table 5.5

Table 5.5 Course demand information

Course	No. of sections	Total seats	Total approval codes	Total requests received	Total approval codes allocated	Total seats remaining
ICS 211	5	182	265	41	41	141
ICS 212	3	105	148	17	16	89
ICS 241	2	80	118	11	11	69
ICS 311	1	50	75	39	39	11
ICS 312	2	80	96	91	91	0
ICS 313	1	50	71	59	58	0
ICS 321	1	50	82	26	26	2
ICS 331	1	40	57	14	14	26
ICS 331L	2	40	57	13	13	27
ICS 413	2	60	90	35	33	27
ICS 414	2	60	90	3	3	57
ICS 415	1	40	58	67	58	0
ICS 421	1	40	60	39	36	4
ICS 424	1	20	30	31	29	0
ICS 451	1	40	59	33	33	7
ICS 491-1	1	20	37	55	31	0
ICS 491-2	1	20	37	16	16	4
ICS 491-3	1	20	37	47	37	0
ICS 491-4	1	20	37	14	13	7
Total	**30**	**1017**	**1504**	**651**	**598**	**471**

Course demand information. As we can see from the table 5.5, INCA provides the following information.

- It tells how many requests were received for a course. How many of the students were granted approval codes and how many of them were denied?
- Which courses got their seats filled up and more students wanted that course? These courses are the popular ones. For example, ICS 415, ICS 424, ICS 491-1.
- Which courses got more seats than the number of students requesting the course? These courses are less popular ones. For example, ICS 414.

Student statistics. INCA provided the following student statistics for Spring 2002:

- Total students in database: 549
- Number of undergraduate and graduate students: 471 undergrad, 71 grad, 7 other
- Number of ICS majors: 382
- Number of graduating seniors: 71
- Number of continuing students: 535
- Number of students in honors program: 0
- Number of student email addresses: 480
- Total number of students who logged into INCA for Spring 2002: 343
- Students from different majors: 82. Unclassified (16), General arts and sciences (15), Gen business (7), Chemistry (6), Biology (4), Electrical Engineering (4), LIS (4), Economics (3), Mathematics (3), Mgt Info Sys (3), Communication (2), Pre-medical (2), Psychology (2), Accounting (1), Animal sciences (1), Architecture (1), Art (1), English as a second language (1), Geography (1), International Mgt (1), Linguistics (1), Meteorology (1), Physics (1), Political Science (1)

5.3.3 Database analysis results and hypothesis

As Table 5.5 indicates, INCA provides information regarding total number of course requests received for a course. INCA also provides real-time information on the remaining course seats and the number of approval codes that have been allocated.

The popular courses are the ones for which there are more requests than the number of seats. During Spring 2002, all the seats for these courses got filled up during initial registration period that ended 21 December 2001. Table 5.6 lists the popular courses for Spring 2002 semester. This is useful course demand information that INCA provides on a real-time basis. With this information, the department can take the decision whether to open new sections or increase the number of seats for a course. For example, the department can open new sections for ICS 312, ICS 313, and ICS415 in the next semester. Also, the department can increase the number of seats in ICS 424, ICS 491-1, and ICS491-3. It also seems from the table that courses involving web programming seem to be very popular. So, the department can try to offer more courses involving web programming in the future.

Table 5.6 List of popular courses for Spring 2002

Course	Course Name	Sections	Seats	Requests
ICS 312	Machine Language and Systems Programming	2	80	91
ICS 313	Programming Language Theory	1	50	59
ICS 415	Introduction to programming for the web	1	40	67
ICS 424	Application Frameworks	1	20	31
ICS 491-1	E-Commerce	1	20	55
ICS 491-3	Design for the Mobile Internet	1	20	47

INCA also provides demand information for courses that are less popular. For example, ICS 414 has 60 seats in 2 sections and only 3 requests were received. This implies ICS 414 is not a very popular course among students. So, the department may offer only one section for that course in future.

Without INCA, all this information was very difficult to obtain. This supports the hypothesis that INCA provides valuable data to the departments that can be used in future curriculum planning.

Chapter 6. Conclusions

This chapter describes my research conclusions. First I will summarize my research. This will be followed by the contributions of the research, limitations of the research, and the possible future directions.

6.1 Research summary

The ICS department was facing problems with manual approval code distribution. INCA was designed to automate the approval code distribution process. This thesis looked into the problems that existed before and after INCA. I discussed how INCA solved the previous problems and what newer kinds of problems came with INCA. I analyzed INCA from the perspectives of the ICS course administrator, the ICS students, and the ICS department. I found that INCA reduces the administrative overhead, but it introduces new kind of work for the administrator. I found that most of the students are satisfied with INCA and prefer INCA to be used for the next semester to the old email based approach for approval code allocation. INCA also helped in improving the predictability and understandability of students of approval code allocation process. Finally, INCA provides very useful data to the departments for their curriculum planning.

INCA was not only designed to automate the approval code distribution but also to improve the quality of the ICS program by promoting good students. INCA does that by means of a prioritization system. The prioritization system defines certain rules and ranks the students by giving them points according to those rules. Even though most students prefer INCA to be used for next semester, they don’t want the current priority rules. They strongly disagree with the use of GPA in priority rules.

6.2 Research contributions

Evaluating student information systems. This thesis discussed the evaluation of INCA from different perspectives. Student information systems should be evaluated not only from the administrator perspective but also from students and university or departments perspectives. The student information systems should be further tested for performance and scalability.

Designing priority rules for a course allocation system. Designing rules to prioritize students is not easy. INCA was designed to improve the quality of ICS program by promoting students with good academic credentials. So, a merit-based allocation system was implemented. On the other hand, PA’E implements seniority-based and first come first serve allocation schemes. Students are accustomed to PA’E style allocation methods. They opposed INCA selection scheme because it penalizes students by giving them negative points. The priority rules should be designed in such a manner that they favor students with good academic credentials but don’t penalize other students.

6.3 Research limitations

Email analysis limitations. The emails received at uhmics@hawaii.edu don’t describe all of the problems. If a student never emailed a problem or directly approached the department or course administrator to discuss the problem, then, that is not covered.

User survey limitations. Three questions in the user survey didn’t have as many negative responses as the number of positive responses. Questions 2, 3, and 4 have two positive responses, one neutral response, and one negative response. There should have been two negative response options. However, after looking at the numbers, I feel that it wouldn’t have made much a difference in the results even if two negative response options had been there.

No exact measure of administrator time. Ideally, One would compare the exact time administrator spends in approval code allocation to prove that INCA reduces the administrative overheads. However, it is extremely difficult to obtain the time spent on approval code allocation, as it is hard to manually collect the time spent in reading and responding to the emails. So, I used a qualitative approach instead of a quantitative approach to prove that INCA reduces the administrative overheads.

6.4 Future directions

Further research on INCA can take several paths from here. This section summarizes a few of the directions.

- **INCA across the UH.** Presently, INCA is only being used in the ICS department. Other departments at UH are still using the manual approach to approval code distribution. In order for INCA to be used in other departments, there are two approaches: First, customize the current INCA source code to the department needs and let each department maintain the INCA server. Second, implement a department module in INCA and let one INCA server handle all the departments. The INCA business plan [12] discusses the departmental and university versions of INCA as well as the commercialization issues of INCA technology.
- **Integration with existing systems.** Currently INCA gets the student data from ISIS and the approvals data from PA’E in plain text files. INCA conversion routines convert data in plain text file format to XML format and then data is entered into INCA database. Over the registration period, changes happen to the database as new

students are entered, transfer grades of transfer students are entered, and basic status information of students in updates. So, we are maintaining student data at two places: ISIS and INCA. A better solution would be to integrate INCA with the existing systems so that we have just one copy of database. Another possible approach would be to implement the course prerequisite processing module and ranking module inside the PA'E.

- **Multiple roles in INCA.** The department course administrator and the system administrator functionality should be separated. The faculty role should be added to allow faculty to see the requests and approval information and to provide consents.
- **Academic advising.** INCA not only tells about the different course requirements, but, also about chances of getting into the courses. It can be configured to tell students what courses they should take in order to graduate and what different options they have.
- **INCA over phone.** INCA should be made available via touchtone telephone systems so that students that are not familiar with INCA can use the system.
- **Priority points mechanism.** The priority point mechanism needs more research. New rules should be devised and existing point values should be researched.
- **Usability.** The enhancements in user interface should be done to improve the user experience. Usability testing should be done to make interface look and feel better.
- **New features addition.** Help pages should be added. New reports should be added for the administrator and faculty. Security should be improved by encrypting the data to and from INCA.

Appendix A. INCA design details

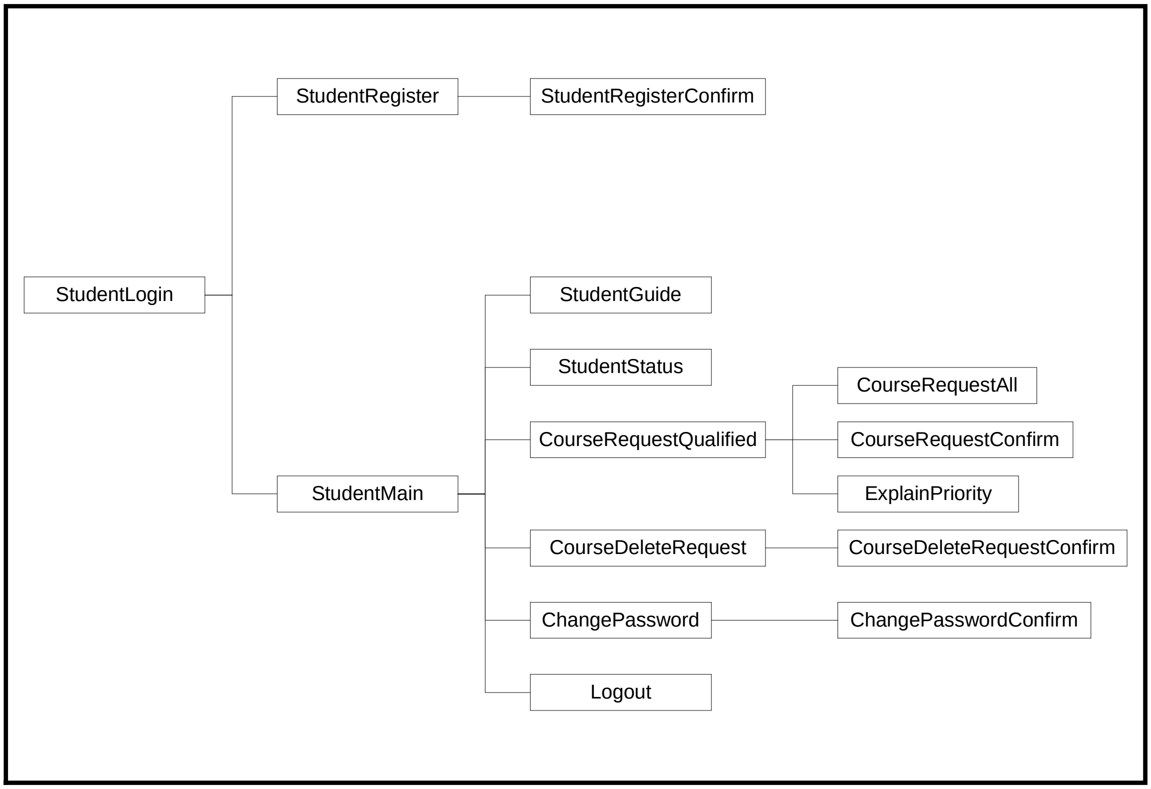


Figure A.1 Student JSP navigation diagram

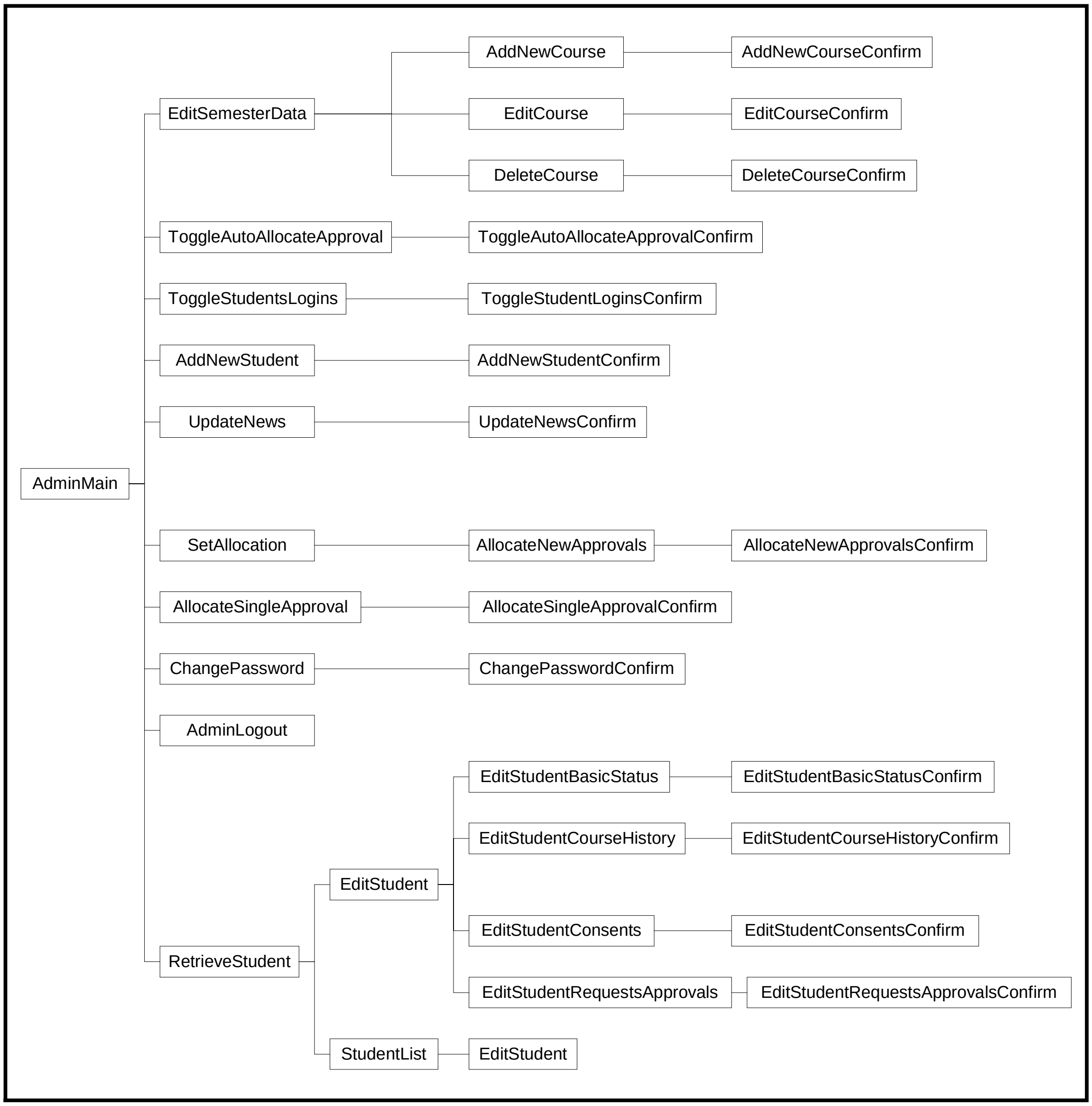


Figure A.2 Course administrator JSP navigation diagram

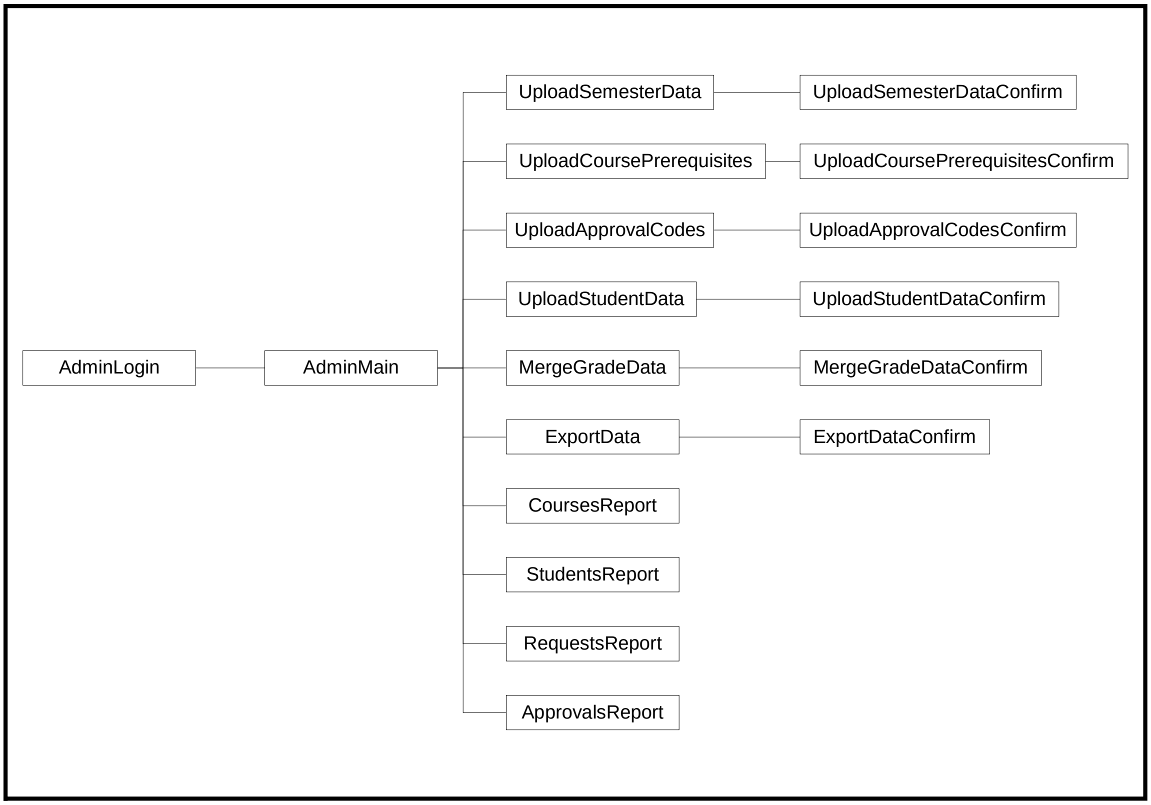


Figure A.3 System Administrator JSP navigation diagram

Table A.1 INCA package structure and important files

edu.hawaii.courseapp	course	ejb		Course|EJB|Home|PK, Section|EJB|Home|PK, Requisite|EJB|Home|PK CourseMgr|EJB|Home
		model		CourseTableModel, CourseInfo, RequisiteInfo, SectionInfo
		test		Test*
		util		*ComparatorID
	student	ejb		Student|EJB|Home|PK, Grade|EJB|Home|PK, Consent|EJB|Home|PK, StudentMgr|EJB|Home
		model		StudentStatusModel, StudentInfo, GradeInfo, ConsentInfo
		test		Test*
		util		*ComparatorID
	request	ejb		Request|EJB|Home|PK, ExcludedSection|EJB|Home|PK, RequestMgr|EJB|Home
		model		RequestTableModel, RequestInfo, ExcludedSectionInfo
		rule		Rule, *rules classes*
		test		Test*
		util		*ComparatorID
	approval	ejb		ApprovalCode|EJB|Home|PK, ApprovalMgr|EJB|Home
		model		ApprovalTableModel, ApprovalInfo
		test		Test*
		util		*ComparatorID
	control	ejb		StudentController|EJB|Home, AdminController|EJB|Home, JSPBeanUpdateManager
		event		*Event
		data		*Data
		util		AuthenticationMgr, WebKeys
		web		MainServlet, RequestProcessor, ScreenFlowManager, JSPBeanManager, AdminControllerWebImpl, StudentControllerWebImpl
			handlers	*FlowHandler, *RequestHandler
			jspbeans	*JSPBean
	util			EJBUtil, JndiUtil, Log, JNDINames

Appendix B. INCA user survey

As part of my M.S. thesis research, I am performing an evaluation of the INCA system in order to support future improvements. Please support this effort by taking a few minutes to answer the following six questions. Participation in this research is voluntary, no information about your identity will be used in the results, and you will receive no course credit for participation. For information about the results of this research, please contact me, Jitender Miglani, at jitender@hawaii.edu.

1. Please rank the overall quality of INCA in terms of ease-of-use.

- Excellent
- Very Good
- Good
- Average
- Poor

2. INCA provides real-time information on total seats, total requests that have been made, and your ranking in the queue. This is to help you in predicting the chances of getting the approval codes. To what extent was this useful feature?

- It was extremely useful. Now I can predict my chances of getting the approval codes.
- It was somewhat useful. It is nice to know your rank in different courses, but I think I can get the courses I want without it.
- It was not useful at all.
- No opinion. It was not applicable to me.

3. Many ICS courses have prerequisites, co-requisites, and concurrent course requirements. INCA tells you why you are qualified or disqualified for a certain course. To what extent was this effective in improving your understanding of different course qualifications?

○ It was extremely effective. Now I know why I am qualified or disqualified for certain courses.

○ It was somewhat effective. I already knew course qualifications, but it is nice to know them through INCA.

○ It was not effective at all.

○ No opinion. I don't care about understanding course qualifications.

4. INCA provides the "priority points" mechanism that enables students with a strong academic background to have higher priority for their future courses. To what extent do you support this approach to course allocation?

○ I strongly support it. It should definitely be used next semester.

○ I prefer it. But it doesn't matter if it is not used for next semester.

○ I don't think it should be used.

○ No opinion. I don't care.

INCA prioritizes the students using different rules such as: favor advanced students; favor students with high ICS GPA; favor first and second choices of students; and do not favor repeat attempts. Do you have suggestions for new priority rules or changes to the current priority rules? Should point values be changed?

5. If you would like, please tell us about the problems you faced or give some suggestions for improvements in INCA.

6. Do you prefer INCA or the old method of sending emails to uhmics should be used for distribution of approval codes?

- I prefer INCA
- I would prefer sending emails to uhmics@hawaii.edu to request the courses
- No opinion, I don't care

Appendix C. User survey results

This appendix lists all the responses to questions 4.2 and 5 of the user survey. Responses to questions 1, 2, 3, 4.1, and 6 are listed in Section 6.2

Question. INCA prioritizes the students using different rules such as: favor advanced students; favor students with high ICS GPA; favor first and second choices of students; and do not favor repeat attempts. Do you have suggestions for new priority rules or changes to the current priority rules? Should point values be changed?

Responses.

1. Favor students with high ICS GPA is a good thing, but INCA should not totally based GPA and nothing else.

2. if you always give the repeat attempts students low priority, they will have a harder time getting the classes they need to graduate.. you're holding them back a lot.. you should change that...

3. For those students that are seniors, and if they aren't doing so good, they may have a harder time getting into classes, which means a harder chance to grad.

4. The point system is harsh if someone is going to repeat a course. From my viewpoint, someone with a higher class standing should get the class over a underclassman who has more chances to take the course.
In summary, I think the priority should go as follows:
If the class is an upper division class, prioritize by class standing, then within that, by first timers then repeaters.

5. The GPA spectrum for giving priority points is too broad. It should be something like < 3.0, 3.0 to 3.33, 3.34 to 3.6, and 3.7 >

6. The points given to a person with a 3.0 ICS GPA or higher is drastically different than a person with less than 3.0 GPA. Point difference should be a little closer. Someone with a 2.9 GPA has 6 points less than someone with 3.0 GPA.

7. Withdrawals should not carry great negative weighting. It's almost enough that the student is out the cash, in terms of a penalty.

8. Students with higher class standing should be given the top priority in registering for classes. Thats how it is with the rest of the university. Just because a student has a low gpa, or not considered an advanced student delays the students abilities to do better in school by not being able to take classes he or she needs at the time as opposed to taking it another semester because he or she didnt have priority points to take the class. The ICS department should reconsider this priority system. If this continues, it only delays students from graduating from the ICS program. No matter what their Major GPA is or how many times the class has been repeated, they should be given the chance to take the class based on their class standing. It is only fair to keep the priority system the way it was before INCA came about. The priority system should be kept the same as how the rest of the university allows student to register for classes. Whats the sense of delaying students from graduating?????

9. don't like the idea of dis-favoring repeated attempts.

10. Maybe it should be by class standing such as Junior, Senior etc

11. I think the priority rules are fine for most students, but it might disallow certain students from continuing in the program until they "beat-out" other students for a seat. If they have a low priority, they will get bypassed often and might become disgruntled.

12. What about some kind of seniority points? I do not want to wait another semester for a required WI course since those courses are so limited. I am also unsure if there's negative scoring for grades D or F but if there is, I think those who recieve those kind of grades already lose alot of priority points for retaking classes. Will the priority points change if the schoolwide grading system converts to +/- system?

13. I don't think we should use this system. People with lower GPA's might feel inferior because of this. We should just make it on a first come first serve basis.

14. definitely yes. it should be changed. students who tried to continue their class knowing that they are at the borderline of a "b" and a "c" get penalize twice. once for lower gpa -3 and for repeat -3. whereas, if they had just withdrew they would have a -1. the situation could be reverse. a student is borderline and decided it was best to withdrew than continue the course with the possibility of not getting a "b". he/she could have pass, but wouldn't know. what if half the students that withdrew could have pass? wouldn't it be better for the ics department and students if they had continue and struggle it out. i personally think it is better than giving up a semester. suggesting -1 for gpa lower than 3.0. -1 for repeat. that way the difference between withdrawing and continuing a class would be 1.

although come to think about this. it really isn't fair. student who have 4.0 and +5 are probably honor students and get an addition +5.

anyway, hope you don't increase the withdrawal point just to be even. students with good grades already have the advantage. give the rest of us a break. some of us can program, but just can't remember everything in the book for the exams.

15. I think that all students should have equal chance at getting a course.
Hopefully juniors and seniors can get the courses they need to graduate by having the advisors keep a record of what courses they want to take, and when so that the faculty can adjust their teaching schedules.
Besides that it should be on a fist come first server basis, but with added notices about the exact date when it will first be possible to get courses and/or approval codes.

16. I don't agree with the points system. It really segregates the students in an unfair manner. Just because someone gets a better grade than me in class does not make it right for them to get first dibs on a class. Why would this even be an option? If this was something like a scholarship or something other than a University derived system it might be okay. This actually angers me that it was even put in. I get a worse grade and i spend as much time or more on a project and homework and that dems someone else to get a better ranking than me.

17. I don't think it is fair to favor students with high ICS GPA. I think the approval code system should be based on class standing because the higher your class standing, the more you need certain classes. The rule i think is great, but might need to be modified is the one about repeat attempts. I think those students who have dropped the course or withdrawn should be given a low priority for the same class. But if you took the course and just didn't get a high enough grade to pass, you should have a higher priority for that class the following semester.

18. I have stated my opinions about this last semester. I feel that in this system "the richer get richer, the poorer get poorer", so to speak. I think what has hurt me a lot is the way my major GPA is distributed pointwise. Having points for a high GPA is good but I think the penalty for low GPA should be less or none at all. Either that or remove my Fs and D from the grading scheme. I have taken the class again and I think the passing grade should be used to determine GPA (granted this probably wont help my GPA very much, but it may help others).

19. I think it should not favor repeat attempts only if the student didn't do well or dropped the course at the last minute. But if the student just chose to withdraw early in the semester because of schedule conflicts or things like that, then he/she shouldn't be penalized.

20. i think it should favor advanced students. People close to graduating should be given priority so they can get their classes and graduate on time

21. I don't think that gpa should be a factor when determining priority for approcal codes. Many things can affect your grades in school. Some of us, who come from disadvantaged backgrounds must pay our own way through school, by working either full or part time the whole year round. This makes it difficult to maintain a high gpa, since the majority of our time is spent working. By having gpa as a factor in the priority of us getting classes, those of us that are disadvantaged financially will take longer to graduate, and in fact will have to spend more money (since we are in school longer) on our education, than

someone who comes from a rich family, and who doesn't have to work for a living while attending school. It's very disheartening when I sign up for an approval code before anyone else, and somebody comes in at the last minute, and since they got an A in the prerequisite class, and I got a B, they get in ahead of me, even though I signed up two weeks before they did. Of course, this would be a moot point if the university could pay our professors more, so that we could have more instructors (and more QUALIFIED instructors), and then we could have enough sections of each class to fulfill everyones needs. I think the ics department has less than 20 instructors, and that is a travesty.

22. I don't see any priority given to graduating seniors. I thought it was in last semesters inca system. I also think that this should be assigned a relatively high value on the point scale.

23. I don't think GPA and should be used as priority. Anyone wanting to enroll in a class should be given the same opportunity to get in a class on equal footing as anyone else. The fact that the class may be difficult and students may not pass already serves as to deter a non-serious student from taking the class. Also, repeat attempts should not be penalized. Taking into consideration the advanced students and the first/second choice should be retained.

24. Points should be taken away from students who are repeating a course, however those points should be put back if they see a counselor and find out why the student is taking the course again. For instance, perhaps the student more classes than they could handle, ICS may not be the major for them, they had problems with other parts of their life, or some other reason. In any case, students should not be disfavored if they are honestly trying their hardest to understand the course material.

25. If some kind of rules are already used, such as prerequisites, why should make more rules? I think as long as they pass, we'd better give them chances to learn. Thank you.

26. Well since i don't support the priority rules, i don't really have any suggestions to improve them.

27. I think that seniority should be the priority rather then GPA just as Pa'e is set up. This way everyone will get a chance to get into the classes that they really want at some point because everyone will eventually gain their seniority.

28. why doesn't class standing have any priority? seniors need to graduate!!

29. I had to repeat and I did much better the second time around. I don't think that repeating should hinder your chances of getting a class. Also I don't think GPA should be a factor because for the same class, one Prof. will be totally different from another for the same class, giving some students an advantage. Advanced students? isn't registering first say that you are an advanced student? I just don't like the idea of priorty, I find it a bit unfair.

30. Graduate students should be given priority, since many of them need to make up undergraduate course deficiencies in order to graduate.

31. Yes, point values should be changed. I don't think a student's GPA should effect their chances of getting the classes that they need/want. Some students have children and must work full time and don't have as high GPA as other students whose parents do everything for them.

32. Do what every other departments do: giving out approval codes in person!!!!! We go to the office, filled out a form, then get the approval codes. If the ICS office doesn't know how to do that, go ask the math department.

33. I think that repeat attempts should not be looked down upon, in fact keeping them from registering will deter them from completing the ICS degree.

34. I feel that it's a burden on the server it is installed on and should be deleted

35. I don't care...I have no suggestions

36. nope, i think the priority points are unfair.

37. dont do that, it sucks

38. i'd prefer no changes

39. no i do not favor it

40. i don't think repeat attempts should be a factor.

41. I think people near graduation should get a higher priority

42. dont favor students like that

43. The point system hinders the chances for students who have done bad in the past but have turned a new leaf to get forward in reaching their goal of graduation. Not being able to get classes to graduate because of your grades is a descrimination.

44. dont use it

45. Seniors should have highest priority. Juniors should have second highest priority. Sophomores should have third highest priority. Freshman should have the lowest priority. There should be no other factors in determining priority.

46. It doesn't matter so much. Well, it maybe work to promote the study.

47. Change the Darn thing, TOO MANY PROBLEMS..., Information is not accurate, TOO MANY HUMAN ERRORS. PEOPLE BEING DELETED FROM THE SYSTEM, CRITICAL INFORMATION NOT UPDATED... IN SHORT WAY TOO MANY ERRORS! GET RID OF IT PERIOD!!!!!!!!!!!!!!!!!!!!

48. it sucks, i dont think it should favor students

49. I think the points for the first choice, second choice... shouldn't be used. It's difficult because sometimes we would just like a course that fits in our schedule, and ranking them like that means that we are less likely to get course codes for classes that come after our second choice.
Also, I think there should be a way to enter in the consent of a teacher if you haven't taken the required pre-requisites.

50. It feels wrong to penalize students who have a lower GPA or withdrawals. Given the number of students in the ICS program, such students may never get the chance to take the courses they want or need to graduate. What kind of message are we sending out here? You're too stupid to take these courses? I don't think any other undergraduate programs do this; all the other programs go on a first-come-first-served basis.

51. its junk

52. Yes, the students are doubly penalized for having dropped classes and having below a 3.0 gpa. The ones who drop are likely to have low grades. The system should be revamped to penalize drops, but instead of deducting points for low gpas, give points for having a 2.75 or higher. The system is set up right now to only give points for honors, getting Bs, and making a class 1st or 2nd pick. This forces those who are not the best students to become stuck in a rut they have no way of making it out of.

53. The priority system may have flaws. Just because one student has better grades does not make him or her a more advanced or favored student, especially in ICS. There are a lot of smart students (by certain rules) that cannot be in the honors program. Prioritizing the students may, in effect, help students who do not need help and hurt students who do need help. The first and second choice thing I like.

54. Point values seem to be fine, however, I'd like it if Withdrawals expire over time. Since I've personally withdrawn before, before INCA was used, I'll be penalized for it until I graduate. Having known this before hand, I might have just taken the D or C grade I was heading toward.

Question. If you would like, please tell us about the problems you faced or give some suggestions for improvements in INCA.

Responses.

1. The main problems I have is the courses I am taking right now does not take any consideration in INCA. This is bad because students always stay one semester ahead of the prerequistes. This system will require students to have two semester ahead to be fully qualify for the course request. Example, I am taking ICS313 now. I am not qualify for ICS424.

2. I don't like using my SSN as my login name. This is similar to my gripe with PAE. Maybe a better way to do this is to use your UH email address since the database captures this information anyways.
I do favor this system rather than the old method. Under the old method, you would have to email Lei and would not get immediate feedback. The priority points and number of requests do help in easing the process of distributing approval codes.

3. It's pretty hassles to wait until the end of the semester to get approval code for a course that has a prerequisite that I am currently taking. Usually I don't get the approval code until after tuition is due.

4. Wasn't in the database. Had to go to the ICS department to get my profile inputted into INCA.

5. I don't think the instructions are clear.

6. As a graduate student with a BS in other field, I need to make up 10 300+ undergraduate classes. But I feel very comfortable since whenever I want to take a class I have to send email explain I have enough background although I didn't take... It is kinds of stupid. I think I am quite clear what kinds of stage I am on and whether I can handle it. It is also true for other graduate students who have similar situations. As graduate students, we know better about ourselves than average undergraduate and so I never know any of us drop a class because he/she can't handle it.

7. INCA's delays last summer (about entering ICS 211 final grades for the Spring semester) did not let me get an approval code for this Fall's classes until about two weeks before the semester started. This meant that I had to arrange my current schedule based on the assumption that I would be getting into the ICS classes I needed. This, combined with Pa'e giving priority that was time-sensitive, meant that many of the classes I should have had priority to register for over freshman and sophomores, actually placed me at about the same priority level as incoming freshman. Also, because I couldn't officially register for the ICS classes, it caused a lot of problems with the financial aid people thinking I was a part-time student and not wanting to give me financial aid. I was forced

to add an introductory class before the payment deadline so I could get myself to full-time status, then dropped it once I had my ICS classes.
INCA is also still listing me as a non-ICS major although I have been one for the past two years. This implies that I will have to take ICS 241, although I am an ICS major and have passed ICS 141 before the class was added to the degree requirements.
I know for a fact that a fellow undergrad was also able to get an approval code for ICS 312 (and he is currently passing it) without having completed ICS 141 with a 'C' or better. I believe he took ICS 141 concurrently with 211 and only passed 211, yet received approval to get into 312.

8. I am a graduate student with an undergraduate degree in mathematics. I completed some of the entry requirements as an undergraduate at a different school, and because this I cannot signup for courses I am qualified to take at times.

9. My grade is not updated since July. And it has some errors. I wonder how can I report those errors and ask for update.
Is INCA serve as my record in ICS department?

10. too confusing, layout very cluttered , bad design.
layout too confusing, only INCA programmers can understand how INCA works.

11. It sucks that you can't request for a course until your grades from the current semester are put into the system. It means I have to wait until right before the deadline because it takes so long and I can't register when it is my registration appointment time.

12. I would like to see a more flexible process. Currently I'm taking a course which is a prerequisite for another course. However, I can't request the more advanced course because INCA apparently doesn't take in to account my current courses.

13. The only problem i had with INCA was in the Summer 2001. I was trying to register for the fall semester, but INCA didn't know what classes i was currently registered for, so i couldn't register for any ICS courses at that time. I know it depends on whether i pass my current classes ro not, but you should at least be able to get some sort of approval code or reservation.

14. I am taking ICS 141 and ICS 312 this semester. But those classes are prerequisite for registering ICS 331 or ICS 313. I cannot add any of these class. If I can't add them right now, at least, I expect some information about this situation. Thanks.

15. Last semester when I used INCA, it didn't even give me my approval code. I had to wait and wait, and eventually I just decided to email uhmics@hawaii.edu for an answer because I didn't want to keep waiting or all the courses might have closed by then and I wouldn't be able to take any of the courses at all. I would prefer if all students were able to put in requests a week before registration and INCA released all approval codes the day before registration starts, and from the second day of registration on, it should be first-come first-serve.

16. Last semester I was given a code, and when i tried to use it, it said that it had already been used.

17. As I understand it, once we get an approval code for a class, it is impossible for us to somehow "give" that code back, per se, and attempt to get a different approval code in its place. Since all of the +200 level courses require a C or B in a prerequisite class, we often cannot finalize our schedule until after grades come out, and this leaves many ics students scrambling at the last minute to make all their classes fit. Students may have to sign up for a different section, or maybe none of the sections will fit in a students schedule. So then the student is "stuck" with an approval code for a class that he/she cannot take (because of scheduling constraints). I believe that the approval codes should be given on a first come, first served basis. Also, there should be no limit on how many approval codes are given out per course, or how many approval codes a student may receive. Approval codes only exist to show proof that you have the proper prerequisites to take a course, not to show priority to certain students. Registration through pa'e handles student priority in attaining classes. The current system uses the approval codes in a way they were not meant to be, and it's also kind of redundant to have 2 methods (the current approval code system, and registration through pa'e), that do the same thing (handle student priority in attaining courses). All in all, I prefer INCA to having to obtain approval codes in person, or by email. INCA can show us on one page, in a compact format, what classes we currently qualify for, and it makes it easier for us to see a graphical representation of our past courses, and what courses we may take next semester. I only have a problem with the logic used with the assignment of approval codes. Everything else about INCA is very good, and whoever (whether it was one person, or a group of people) created INCA in its current form, deserves my gratitude, and a pat on the back for a job well done!

18. I had some problems with the initial logging on and incorrect info on my record but it seems to be working okay now. I assume this was a one time problem with the initial data loading and login id.
One improvement would be to get approval codes faster like by the next day or at least more often than 2 times a week.

19. When I first started using this system, I did not know the web address to the site. It would be helpful if a link to INCA was placed on the ICS homepage.

20. I think 331 and 331L can have the same priority. 331 and the lab 331L are co-requisits, but the same choice can only be picked up once. I chose 331 as my 1 choice, and 331L 2nd choice, but I do hope I can take 331 and 331L at the same time .
Thank you.

21. With two sections named "basic status", things can get somewhat confusing. Also, it's not very easy to figure out how to use the system the first time, although it's very simple once you've learned how. Perhaps first-time users could be greeted with a quick introduction to the system.

22. I haven't had any problems so far. Its great!

23. The only real problem i had was INCA's startup, everyone who took an ICS class that needed an approval code registered late.

24. For students who took lower level courses, i.e. 111 at other campuses,
the prerequisite pages gives erroneous information. But this is a minor detail and the fault of record management at the U.H.

25. I got a strange error and INCA was not letting me submit my choices.

26. it doesn't work most of the time, courses cannot be requested because grades cannot be entered fast enough from the previous semester..
ie; if taking ics141 now and expecting to pass, student should be allowed to apply for courses, otherwise he waits a semester for nothing!!!

27. Still showing a class that got cancelled. When I deleted a request, it did not register in the DB right away, even though it showed on the status page. I know because when I went to add another request it would not let me, kept telling me I have to delete a request even though I did earlier. When a class was cancelled, I could not delete it. That also screwed me over since I could not request more classes. Also it started emailing me registration codes 2 weeks after the semester started for classes that it did not give me codes initially.
I hate INCA.

28. It would be nice if it were possible to delete a request even after receiving an approval code for it.

29. There probably should be a separate help page that explains the INCA process in general. For example, I have just registered but I did not
get any approval codes. So, I am assuming that after the registration
period has expired, approval codes will be mailed out. This should be
made clear on INCA somewhere.

30. for the 491 courses, there should be a little description about the course.

31. I always have to change my password.

32. why does inca display a higher number of approvals than it does requests for some courses...this is very confusing since it isn't explained why this may happen

33. I had a big, big problem that I couldn't register on INCA last semester since it first started. It didn't recognize my name and ITS account. This semester, right now, I could not get into the account again. For the past week, it said my student ID is invalid, which is totally impossible. Today it finally accepted my SID, but my password is not accepted

now, and I'm 100% sure that my password is the one I used before. So far, INCA system worked like hell to me. I'm using somebody else's account to write this protest.

34. My priority points were off, and it says that I am not an undergraduate when i wanted to take a class in the 200 level.

35. the layout of the priority tables were quite confusing to understand...

36. My student I.D. (user ID) was deleted from the database from last
semester. I had to e-mail inorder for it to be reinstated. This was a
little inconvinant, because I could have requested the class I desired
a day earlier.

37. I have requested approval code for ICS415 and ICS421. My ranking is below 10, but I still do not receive the approval code yet. On the other hand, my friend has a negative point and have the ranking around the 30's and get the approval code. The INCA program has some serious bugs in distributing the approval code needed to fix. I think this is not fair at all

38. Let me in to system. Better dispersal of codes. Allow consent for codes

39. i dont know what the ranking means

40. INCA does not automatically reflect course list with transcript submission. there probably should be some sort of centralized, automatic entry of transcript records.

41. Compared to other departments, which give approval code even before final grade comes out, the INCA system is inefficient and ineffective as well.

42. Earlier When I tried to login. It would display an error message saying that my ID was not in the system, so I then I registered again and It also displayed the same error message. Today I registered again and everything is fine now. But it was a hassle. Also last semister for me it would send me an approval code every week for a while even thou I had already register for the class. And so of codes did not work.

43. i requested an approval code for ics421, but i still haven't receive it yet, my ranking is 25, but i knew some of my friends have ranking in 30 something, but got the approval code already. What is the purpose of the ranking for? i think there is a bug in the inca system.

44. its unfair, i have to wait forever to get my codes because im not smart enough

45. Why does approval codes take so long. Even with the class rankings and chances of getting the classes I want very high, waiting for the approval codes is combersome. Most times I am at a terminal at odd hours of the day. With time release approval codes registering for my classes can be a problem if I miss a chance to register for a class I

needed. Since Im not at a terminal 24hrs a day I cant reley on this system to get codes on a timely basis. There has to be a better way of distributing the codes.

46. Also, students who are taking the prerequisites for a class that they want to take in the next semester are not allowed to register for them until after grades are out. This is very inconvenient. By the time that students can get approval codes, the classes are all filled up. This system is nothing but another obstacle, if not a barrier, for students who are already struggling to finish college. Being a full time student with a 25 hour/week job, I find that the INCA system is a hindrance to my education, stretching out the time that I need to be at UH from 4 years to 5 years.

47. i had requested an approval code for ics421, but i still haven't received yet, there are peoples ranking lower than mine, they got their approval code already, i think there is a bug in the inca, it can't keep track of who get the priority to get the approval code first. you guys should consider that.

48. Should have someone you can call with questions, not just email. Process to get approval codes should be a step-by-step process with a seperate page for each step and no way to get lost.

49. Since it is machine system, just like human being, it will do sth wrong if there is any bugs. It is recommended to keep both system for the PAC.

50. CRITICAL INFORMATION NOT UPDATED! TO MANY ERROR AND MISTAKES. IF IT CANT BE RIGHT THE FIRST TIME WHEN IT COUNTS GET RID OF THE SYSTEM OR MAKE IT BETTER. AT TIMES LIKE THESE, THIS TYPES OF ERROR SHOULD NOT OCCUR. THE INCONVENICE THIS SYSTEM HAS CREATED IS HIGHLY UNEXCEPTABLE! HIGHLY UNEXCEPTABLE!

51. ditch it

52. It takes a really long time to get my course codes because I am not an ICS major. I am however trying for a minor. I'm beginning to feel like the ICS department doesn't even think about their minors since it is so hard to get approval codes unless you have permission for the classes.
For question #6, I do care but either way for me is the same hassles. I have to go through a lot just to get my approval codes.

53. It's frustrating to request approval codes, then have to wait and wait and wait for the codes to be issued. Then, at the last minute, get bumped out by other students who had a higher priority rating but requested approval codes way later than you did.
There are several ICS491 courses but no apparent description of what the classes are about (aside from the course title.) How about posting this information on the ICS website?

54. I feel #2 on the survey may convey mis-information. While it is nice to know your priority in the queue, it gives no information as to the priority of students ahead of you. Therefore a student who knows that he has high priority can register with INCA at anytime before codes are given out and effectively bump you from the class. So while option 1 indicates that it is extremely useful, this is not necesarily the case, it is a good tool to have, but doesn't really give you enough information to establish the probability of your getting the class. Option 2 is also oddly worded, this feature is useful in that it tells you what rank you are for that class, however, its fairly obvious that this feature alone won't get you the classes you want.

55. information across pages is sometimes in accurate. On the requests page and the basic info page, the number of appovals differs.

Bibliography

[1] Chad Vawter and Ed Roman. *J2EE vs. Microsoft .NET, A comparison of building XML-based web services.* The Middleware company. June 2001

[2] Charles M. Tudd, Eliot R. Smith, and Louise H. Kidder. *Research methods in social relations*. Harcourt Brace Jovanovich College Publishers. Sixth Edition

[3] Duane K. Fields, Mark A. Kolb. *Web development with Java Server Pages.* Manning publications, 2000

[4] Jon Elster. *Local Justice, How institutions allocate scarce goods and necessary burdens.* Russell Sage Foundation. 1992.

[5] Lind E. and Taylor T. *The social psychology of procedural justice*. 1988

[6] Nicholas Kassem and the Enterprise team. *Designing Enterprise Application with the Java 2 Platform, Enterprise Edition.* Addison Wesley. 2000

[7] Richard Monson-Haefel. *Enterprise JavaBeans*. O'Reilly. 2000

[8] Roger Sessions. *Java 2 Enterprise Edition (J2EE) versus The .NET Platform. Two Visions for eBusiness*. ObjectWatch Inc. March 2001

[9] Weifeng Miao. *J2EEval: A Method for Performance Analysis of Enterprise JavaBean Applications.* MS Thesis. 2001

[10] J2EE design patterns <http://java.sun.com/blueprints/patterns/j2ee_patterns/>

[11] MIS ITS website <http://mis.its.hawaii.edu/projects.html >

[12] INCA business plan <http://csdl.ics.hawaii.edu/techreports/01-06/01-06.pdf>

[13] INCA Software requirement specifications <http://csdl.ics.hawaii.edu/techreports/01-09/01-09.html>