COURSE PROPOSAL AND APPROVAL SYSTEM

A Paper
Submitted to the Graduate Faculty
of the
North Dakota State University
of Agriculture and Applied Science

By
Elangovan Murugaiyan

In Partial Fulfillment
for the Degree of
MASTER OF SCIENCE

Major Department:
Computer Science

November 2012

Fargo, North Dakota
North Dakota State University
Graduate School

Title
COURSE PROPOSAL AND APPROVAL SYSTEM

By
ELANGOVAN MURUGAIYAN

The Supervisory Committee certifies that this disquisition complies with North Dakota State University’s regulations and meets the accepted standards for the degree of

MASTER OF SCIENCE

SUPERVISORY COMMITTEE:

Dr. Kenneth Magel  
Chair
Dr. Rui Dai
Dr. Hyunsook Do
Dr. Kenton Rodgers

Approved by Department Chair:

11/20/2012  
Date  
Dr. Kendall E. Nygard  
Signature
ABSTRACT

North Dakota State University (NDSU) Distance and Continuing Education (DCE) department makes learning resources of the institution available in a variety of traditional and non-traditional ways. It offers courses in various instruction modes to NDSU as well non-NDSU students. Courses offered by the DCE department must be approved at the department and college level. The DCE department currently uses a paper based process to prepare, submit and gain approvals. In this paper we propose, design and implement an online application to replace this paper-based process. The DCE department staff can use this application to electronically create and route course proposal requests for approval. Submitted course proposal requests are sequentially approved by the DCE department director, concerned department chairpersons and college deans by the using this application.
ACKNOWLEDGMENTS

I would like to thank my advisor Dr. Kenneth Magel for his continued support, guidance and encouragement. Without this help, this work would not have been possible. I would like to thank Dr. Rui Dai, Dr. Hyunsook Do and Dr. Kenton Rodgers for graciously agreeing to serve on the committee. I would like to thank Ms. Lisa Nordick, Assistant Dean/Director, Distance and Continuing Education department for providing me an opportunity to work on this project.
## TABLE OF CONTENTS

ABSTRACT ........................................................................................................ iii

ACKNOWLEDGMENTS ....................................................................................... iv

LIST OF TABLES ............................................................................................... x

LIST OF FIGURES .............................................................................................. xi

LIST OF ABBREVIATIONS .................................................................................. xii

LIST OF APPENDIX FIGURES .......................................................................... xiii

1. INTRODUCTION .......................................................................................... 1

1.1. Introduction ............................................................................................... 1

1.2. Problem Definition .................................................................................... 1

1.3. Drawbacks of the Current Method ............................................................. 2

1.4. Proposed Solution ..................................................................................... 3

1.5. CPAS Application Components ................................................................. 3

1.5.1. Data Loading ....................................................................................... 3

1.5.2. Proposal Creation and Submission ....................................................... 3

1.5.3. Email Notification ............................................................................... 4

1.5.4. Proposal Approval ............................................................................... 4

1.5.5. Post-Approval Processing .................................................................... 4

1.6. Software Development Method ................................................................ 4
1.7. Paper Organization

2. RELATED WORK

2.1. Introduction

2.2. Paper-based Process in Educational Institutions

2.3. Workflow Management Systems

2.4. WFMS Usage Scenario in Educational Institutions

2.5. WFMS Usage Scenarios in Organizations

2.6. General Purpose WFMS

2.7. Course Proposal Systems

2.7.1. Integrated Curriculum Management Request System at UCD

2.7.2. Course Request Approval System at UNC

2.7.3. Course Proposal and Approval System at NDSU DCE

2.7.3.1. Course Proposal System

2.7.3.2. Course Approval System

2.7.3.3. Course Request Approval System at UNC

2.7.3.4. Course Proposal and Approval System at NDSU DCE

2.8. ICMS, CRAS and CPAS Comparison

3. DESIGN AND DEVELOPMENT

3.1. Introduction

3.2. Analysis

3.3. Design
3.3.1. System Architecture for CPAS Application .......................................................... 17
3.3.2. Client Tier ............................................................................................................. 18
3.3.3. Application Tier .................................................................................................. 18
3.3.4. Data Tier ............................................................................................................. 19
3.4. Architecture Alternatives ...................................................................................... 19
3.4.1. Repository Model .............................................................................................. 19
3.4.2. Two-tier Architecture Model ............................................................................ 20
3.4.3. Model 1 Architecture ....................................................................................... 21
3.5. Application Architecture ...................................................................................... 21
3.5.1. Model .................................................................................................................. 23
3.5.2. Controller .......................................................................................................... 23
3.5.3. View .................................................................................................................... 24
3.5.4. Architecture Advantages .................................................................................. 24
3.6. Database Design ................................................................................................... 25
3.7. Implementation .................................................................................................... 26
3.7.1. Use Case Implementation Illustration ............................................................. 26
3.7.2. Analysis ............................................................................................................. 27
3.7.3. Design .............................................................................................................. 27
3.7.4. Implementation ................................................................................................. 27
3.7.5. Testing .............................................................................................................. 28
5.3.1. Data Inconsistencies ................................................................. 43
5.3.2. Mistaken Assumptions ............................................................ 44
5.3.3. NDSU Central Authentication Service ......................................... 44
5.4. Future Improvements .................................................................. 44
5.4.1. CPAS Application Integration with PeopleSoft .............................. 45
5.4.2. Extending CPAS Application ...................................................... 45
5.4.3. Email Remainders ..................................................................... 45
5.4.4. Unified Interface ....................................................................... 46
REFERENCES ..................................................................................... 47
APPENDIX A. COURSE PROPOSAL SYSTEM ........................................ 49
APPENDIX B. COURSE APPROVAL SYSTEM ....................................... 65
# LIST OF TABLES

<table>
<thead>
<tr>
<th>Table</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>1. Use case description for CPAS application.</td>
<td>16</td>
</tr>
<tr>
<td>2. CPAS application entities and their description.</td>
<td>25</td>
</tr>
<tr>
<td>3. Create course proposal request use case analysis components.</td>
<td>27</td>
</tr>
<tr>
<td>4. User login test cases.</td>
<td>35</td>
</tr>
<tr>
<td>5. Create Course proposal request test cases.</td>
<td>37</td>
</tr>
<tr>
<td>6. Approve course proposal request test cases.</td>
<td>39</td>
</tr>
<tr>
<td>7. Edit instructor test cases.</td>
<td>40</td>
</tr>
<tr>
<td>Figure</td>
<td>Page</td>
</tr>
<tr>
<td>--------</td>
<td>------</td>
</tr>
<tr>
<td>1. DCE course proposal request creation form.</td>
<td>2</td>
</tr>
<tr>
<td>2. ICMS course approval process workflow sequence.</td>
<td>9</td>
</tr>
<tr>
<td>3. CRAS course approval process workflow sequence.</td>
<td>10</td>
</tr>
<tr>
<td>4. CPAS course approval process workflow sequence.</td>
<td>12</td>
</tr>
<tr>
<td>5. High-level use case diagram for CPAS application</td>
<td>15</td>
</tr>
<tr>
<td>6. Three-tier client server architecture model for CPAS application</td>
<td>18</td>
</tr>
<tr>
<td>7. Repository architecture model.</td>
<td>19</td>
</tr>
<tr>
<td>8. Two-tier architecture model</td>
<td>20</td>
</tr>
<tr>
<td>9. Model 1 architecture components.</td>
<td>21</td>
</tr>
<tr>
<td>10. Model view controller design pattern.</td>
<td>23</td>
</tr>
<tr>
<td>11. Entity-relationship-diagram for CPAS application</td>
<td>26</td>
</tr>
<tr>
<td>12. UML class diagram for create course proposal request use case</td>
<td>28</td>
</tr>
<tr>
<td>13. UML sequence diagram for create course proposal request use case.</td>
<td>28</td>
</tr>
</tbody>
</table>
LIST OF ABBREVIATIONS

NDSU…………………………………………………………..North Dakota State University

DCE ………………………………………………………......Distance and Continuing Education

CPAS……………………………………………………..Course Proposal and Approval System

CPS…………………………………………………………Course Proposal System

CAS………………………………………………………Course Approval System

UCD…………………………………………………………University of California at Davis

ICMS……………………………………………………..Integrated Curriculum Management System

UNC………………………………………………………..The University of North Carolina at Chapel Hill

CRAS………………………………………………….Course Request and Approval System

UML…………………………………………………….Unified Modeling Language

JSP…………………………………………………………Java Server Pages
LIST OF APPENDIX FIGURES

<table>
<thead>
<tr>
<th>Figure</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>A1. NDSU central authentication login page for CPAS application.</td>
<td>49</td>
</tr>
<tr>
<td>A2. NDSU central authentication service invalid login result.</td>
<td>49</td>
</tr>
<tr>
<td>A3. Course proposal system homepage.</td>
<td>50</td>
</tr>
<tr>
<td>A4. List course proposals by term.</td>
<td>50</td>
</tr>
<tr>
<td>A5. Course proposal summary page.</td>
<td>51</td>
</tr>
<tr>
<td>A6. Fall 2012 accounting department course proposal request listing.</td>
<td>51</td>
</tr>
<tr>
<td>A7. Print view of ACCT 200 course proposal request.</td>
<td>52</td>
</tr>
<tr>
<td>A8. Pending courses in ADHM department for which proposal requests have not yet been fully entered.</td>
<td>53</td>
</tr>
<tr>
<td>A9. Create proposal request page for ADHM 710.</td>
<td>53</td>
</tr>
<tr>
<td>A10. Course proposal requests completed for accounting department courses.</td>
<td>54</td>
</tr>
<tr>
<td>A11. Course proposal request approval status page for ACCT 201.</td>
<td>54</td>
</tr>
<tr>
<td>A12. Approved course proposal page.</td>
<td>55</td>
</tr>
<tr>
<td>A13. Combine courses to create a single proposal for related courses.</td>
<td>56</td>
</tr>
<tr>
<td>A14. Create a combined proposal for BIO/BOT/PLSC/ZOO.</td>
<td>56</td>
</tr>
<tr>
<td>A15. Add a course that is currently unlisted in PeopleSoft system.</td>
<td>57</td>
</tr>
</tbody>
</table>
A16. Select an instructor type to create profile. ................................................................. 57
A17. Create a profile for an NDSU instructor. ................................................................. 58
A18. Create profile for adjunct instructor ......................................................................... 58
A19. List instructors by department page. ......................................................................... 59
A20. Instructor listing for computer science department .................................................... 59
A21. Edit instructor profile page. ....................................................................................... 60
A22. View courses taught by instructor for a term. .......................................................... 60
A23. Reassign department chairperson page ..................................................................... 61
A24. Reassign department chairperson confirmation page. ............................................. 61
A25. Reassign department chairperson success page. ....................................................... 62
A26. Instructor listing page reflecting chairperson change. .............................................. 62
A27. Courses proposal requests page with edit and delete options. ................................. 63
A28. Edit a saved course proposal request. ...................................................................... 63
A29. Choose term to view course catalog. ...................................................................... 64
A30. Course catalog listing page. .................................................................................... 64
B1. Course approvers login page. ..................................................................................... 65
B2. DCE department director home page. ...................................................................... 65
B3. DCE department director home page showing a list of approved and pending proposals..... 66

B4. DCE department director view of pending proposals. .......................................................... 66

B5. DCE department director view of proposal request detail page........................................... 67

B6. Course proposal listing page indicating an approved course has been moved from pending to approved status. ............................................................... 67

B7. Computer science department chairperson view of approved and pending proposals page.. 68

B8. Computer science department chairperson view of a submitted proposal request. .......... 68

B9. Computer science department chairperson approving a course........................................ 69

B10. College of math and science dean view of submitted proposal request details page........... 69

B11. College of math and science dean approving course after DCE department director and computer science department chairperson approval. ......................................................... 70

B12. Graduate dean approving course after DCE department director, computer science department chairperson and college of math and science dean approval. ......................... 70
1. INTRODUCTION

1.1. Introduction

North Dakota State University (NDSU) Distance and Continuing Education (DCE) is an outreach department at NDSU that makes learning resources of the institution available in a variety of traditional and non-traditional ways [1]. Distance and Continuing Education department at NDSU offers online and traditional classroom based courses for NDSU as well as non-NDSU students. All Courses offered by the DCE must be approved at the department and college level for undergraduate courses and additionally at the graduate school level for graduate courses. NDSU faculty senate committee must approve any new course offered by the DCE department in advance.

Approvals obtained by the DCE department for each course for a particular semester are valid for the duration of that semester the course is offered. Proposal request for a course is created by listing course details and approvers in sequential order. Course syllabus must also be attached to each proposal. The prepared proposals are forwarded for approval to next approver listed on the proposal request through campus mail.

1.2. Problem Definition

Distance and continuing education currently uses a predominantly paper based process to prepare, submit and gain approval for courses. At the beginning of each semester, a proposal is created for each course that is offered through DCE by listing the approvers in order using an HTML form. HTML form used in creating a proposal request in a pre-defined format is shown in Figure 1. The HTML form takes course details as input and outputs a neatly formatted proposal. The formatted proposal is printed with its course syllabus, signed off by the DCE department head and sent to the first approver by campus mail. The first approver makes a decision on the
proposal and forwards it to the next approver in chain. This campus mail based routing continues until the last person signs off the proposal. The approved proposal is finally sent to the DCE department. This is then stored in the department for post-processing and record keeping purposes. This manual process has several disadvantages.

Figure 1. DCE course proposal request creation form.

1.3. Drawbacks of the Current Method

1. The current method to prepare proposals is time consuming and error-prone as it requires manual input of courses details to prepare a proposal;

2. A considerable amount of time and effort is spent in routing the proposal through campus mail to the next approver in the approval process. The proposal requires a resubmission if it is lost in campus mail;

3. Tracking the proposal is time consuming as it entails calling each and every person in the approvers list to ascertain its location and status;
4. A considerable amount of office stationary is expended in this paper-based approach.

1.4. Proposed Solution

In this paper, an electronic Course Proposal and Approval System (CPAS) has been proposed, designed and implemented to replace the existing paper-based system. The implemented CPAS system incorporates functionalities that allow DCE staff to create, submit, approve, track, and manage proposal requests for courses offered by DCE during spring, fall and summer semesters. CPAS eliminates and minimizes the drawbacks that are present in the current paper-based approach in place at NDSU DCE.

1.5. CPAS Application Components

Following are the major components of the proposed CPAS application.

1.5.1. Data Loading

This involves querying the NDSU PeopleSoft campus administrative system for courses offered by the DCE department. At the beginning of each semester a detailed list of DCE course offerings are queried, extracted, transformed and loaded into the CPAS database. This data serves as the input for creating proposals. Also, a current list of campus departments, faculty members in those departments and college deans are queried and imported into CPAS database.

1.5.2. Proposal Creation and Submission

Authorized DCE department staff can log in to the course proposal system to select, view and create a proposal for each course that needs approval. The system provides functionality to electronically attach a course syllabus to the prepared request. Prepared course proposal request could be submitted for approval right away or stored in the system for later submission.
1.5.3. Email Notification

Once a proposal is submitted for approval, the system sends out an email notification to the first approver listed in the proposal request.

1.5.4. Proposal Approval

The Course Approval System is used by the approvers to approve or disapprove a course. At the receipt of an email, or at will, an approver can log in to the system to view and make a decision on received course proposal requests individually. When approver approves a proposal, the next approver in the approvers list gets notified that a new course proposal is waiting for his or her decision. The approval process and email notification continues until all approvers approve a course proposal request. When an approver disapproves a proposal, the person responsible for creating and submitting the course proposal request receives an email alert about the disapproval with reasons for disapproval included in the email message.

1.5.5. Post-Approval Processing

Administrative documents can be attached to each approved course proposal request. Relating these documents to course proposal enables the system to present a consolidated view of course proposal request, approval decisions and documents pertaining to the course. Past proposals can be queried for reporting and decision making purposes.

1.6. Software Development Method

CPAS application was developed using evolutionary prototyping. In the evolutionary prototyping model, initial version of the system can be quickly designed and developed to elicit user feedback. User feedback and lessons learned from the implementation are incorporated to iteratively and incrementally produce a fully functional application.
1.7. Paper Organization

The rest of this paper is organized as follows. Chapter 2 talks about workflow management systems and how organizations are using them to improve their business processes. Further, it describes and compares two systems similar to the proposed CPAS application. The third chapter outlines the analysis, design and development activities that were carried out as part of CPAS application development, including database design, software development methodology and design decisions that were considered. The fourth chapter presents evidence on how well the CPAS application fulfills its intended purpose. The final chapter lists the limitations of the CPAS application and recommendations for future work.
2. RELATED WORK

2.1. Introduction

This chapter discusses how paper-based processes are used at educational institutions and organizations. It also talks about workflow management systems and how educational institutions and organizations are using them to replace paper-based processes. Further it describes course proposal systems used at two universities and the proposed CPAS application.

2.2. Paper-based Process in Educational Institutions

Organizations in particular educational institutions collect a vast amount of information using paper forms. From admittance up until graduation each student would have filled out numerous forms at each point in their student career. The information gathered through these paper forms must be processed, secured, stored and retrieved when needed. The sheer amount of work, time and manual labor involved in this process motivated them to find better ways to manage this paper-based manual process.

Organizations started using the World Wide Web (WWW) to publish, collect and store information gathered from its customers. The WWW provided a partial solution to the problem by eliminating paper handling and physical presence required from customers. The collected information in many instances, needed to be routed to different entities for decision-making activities. Organizations started looking for a solution to automate this manual process.

2.3. Workflow Management Systems

The solution came in the form of workflow management systems. Workflow management systems (WFMC) coalition defines a workflow [3] as concerned with the automation of procedures where documents, information or task are passed between participants according to a defined set of rules to achieve or contribute to an overall business goal. Further,
WFMC defines a workflow management system (WFMS) as one that completely defines, manages, and executes "workflows" through the execution of software whose order of execution is driven by a computer representation of the workflow logic.

2.4. WFMS Usage Scenario in Educational Institutions

Educational institutions adapted these systems to replace time consuming and labor-intensive paper-based processes with workflow management systems. For example, educational institutions use WFMS to collect, process and decide on students’ admission applications. Prospective students use an online system to input their application and attach supporting documents. The admission office reviews the submitted materials before electronically routing it to the concerned academic department for processing. As soon as the academic department makes a decision on the students’ application, it is instantly available through the WFMS for further processing. Hiring new employees would be another scenario where WFMS are used by educational institutions. Hiring committee’s use WFMS to collect, screen, evaluate, document and make hiring decisions.

2.5. WFMS Usage Scenarios in Organizations

Hospitals and insurance companies use WFMS to submit and process patient claims. Claims submitted by hospitals on behalf of services rendered to patients are approved by many different entities before a final decision could be made on the claim. Banks use WFMS to process credit card applications submitted by a customer. A decision is made by the system in a few seconds on whether to approve or deny the request. Without a workflow management system keeping track of processing activities happening at each stage and routing it to the next entity for processing can be a time-consuming, error-prone process.
2.6. General Purpose WFMS

Several general-purpose WFMS are available for use by organizations. General-purpose workflow systems don’t attempt to automate any specific business processes. Instead these systems provide tools to model and construct workflows to represent business processes and entities’ involved in executing the workflow and the order of execution. It also provides a runtime environment for executing user created workflows and an interface for users to access and act on workflows. YAWL, jBPM, Processmaker, jFlow and Apache ODE (Orchestration Director Engine) are examples of general-purpose workflow systems.

2.7. Course Proposal Systems

A web search reveals that a majority of universities are using a paper-based process to create, submit and approve course proposal requests. A number of universities [4] have implemented workflow systems to electronically create, route and approve course proposal requests. All available course proposal workflow systems are proprietary and password protected. We could only get the capabilities of these systems by examining available help documents. These help documents detail step-by-step process for creating course proposal requests and the approval process.

Two such systems currently in use at University of California at Davis (UCD) and The University of North Carolina at Chapel Hill (UNC) have been analyzed. UCD has implemented Integrated Curriculum Management System (ICMS) [5] and UNC uses Course Request and Approval System (CRAS) [6] for creating, submitting and approving such requests.

2.7.1. Integrated Curriculum Management Request System at UCD

The University of California at Davis uses an application called Integrated Curriculum Management system for creating and managing curriculum requests, college course catalog and
auditing students’ degree requirements. The curriculum management system similar to our proposed Course Proposal and Approval System allows users to create electronic course proposal requests for approval.

Figure 2. ICMS course approval process workflow sequence.

Creating an electronic curriculum request for approval is a two-step process in the Curriculum Management System. The first step is to add a new course by filling out an elaborate course detail form that has several screens for data input. The second step is to create a curriculum request that would place the newly created course into the workflow. Approvers can examine the submitted requests to approve, disapprove or route the proposal to another person to take action. Approver sequence for approving a new course in ICMS at UC Davis is shown in
Figure 2. The approved curriculum request serves as the data source for the other two systems: college course catalog and student degree audit system.

2.7.2. Course Request Approval System at UNC

The University of North Carolina at Chapel Hill has developed and implemented an application named Course Request Approval System (CRAS). It is very similar in functionality to our proposed CPAS. Figure 3 shows course approval process workflow sequence in CRAS application. CRAS is used at UNC to create, submit, and approve course requests. Course requests is the term used at UNC for denoting course proposal requests.

A course request is created by inputting course details in a web form with multiple data input sections. Once a course is created it is immediately placed in the workflow queue and available for approvers to act on the proposal sequentially. Approvers could approve, disapprove
or change the course status to pending to make a decision at a later time. An approved proposal is added to the university course catalog.

2.7.3. Course Proposal and Approval System at NDSU DCE

The proposed CPAS system is a combination of two systems, one for creating and managing proposals used by NDSU DCE staff and the other for approvers to approve courses, used by DCE department director, department chairpersons and college deans.

2.7.3.1. Course Proposal System

Courses needing approval are queried, extracted, and loaded into the system data tables from NDSU PeopleSoft campus administrative system. Authorized course proposal creators can create a course proposal request and submit it for electronic approval. A course proposal system user can create a proposal by choosing the term, department and a specific course. They can view, modify course details and attach a course syllabus to create a proposal. The created proposal can be submitted for approval either at that time or saved in the system and submitted later.

2.7.3.2. Course Approval System

Approved users can log into the course approval system to select, view and make a decision to either approve or disapprove a proposal. Approver sequence for approving a new course is shown in Figure 4. When an approver approves a course, an email notification is sent to the next approver, alerting him/her that a proposal is waiting for their action. Disapproved proposal triggers an email notification to the DCE department that created the proposal and the approval process stops. Relevant administrative documents can be attached to course proposal request when all approvers approve it. Approved proposals are archived and available for generating reports.
2.8. ICMS, CRAS and CPAS Comparison

CPAS, ICMS and CRAS systems, when viewed at a higher abstract level allow users to create course proposal requests for electronic approval by a set of pre-defined approvers. Each individual system has its own process for creating, submitting and approving course proposal requests. Further each university has defined the number of approvers for each course and the sequence these approvers need to approve submitted requests. UCD course proposal requests are approved by the instructor, department chair, college dean and college senate academic committees. Course proposal requests created at UNC using the CRAS system must be approved at the department and division level and by the Office of University Registrar. Proposals prepared at NDSU DCE are also approved at the department and divisional level, but not by the Office of the University Registrar and college academic committees.
Courses offered by DCE department are queried, extracted and loaded into CPAS application database from NDSU PeopleSoft campus student administrative system. This data serves as the source for assembling and submitting a proposal request for electronic approval using CPAS application. At the UCD proposal creation is a two-step process. A course is created and saved in the system by inputting all relevant information pertaining to the course. The second step is to create and submit a request for approval for the course created in prior step. At UNC, extensive course data is manually input into the system to create and submit a proposal request for approval. Submitted proposal requests are immediately available to the first person in approval chain to act in all three systems.

In our proposed CPAS system, relevant documents pertaining to the course can be attached and saved to the approved proposal request. Courses offered by term and course taught by instructors in the past can be queried for reporting and making administrative decisions. Whereas at UCD and UNC approved proposal data are consumed by course catalog and degree audit systems. This is in contrast to the process at NDSU DCE where course offering data from PeopleSoft administrative system is used as the data source for generating course proposal requests. UCD and UNC help document states that all course proposal requests for approval at the university are handled by ICMS and UNC system. The proposed CPAS system is used only at the DCE department level.
3. DESIGN AND DEVELOPMENT

3.1. Introduction

Analysis, design, implementation and testing activities or phases are present in almost all software development methodologies. Methodologies primarily vary on how, when and at what level these activities are carried out and artifacts they generate. This chapter discusses the analysis, design, implementation and testing activities carried out as part of the CPAS application development. It outlines the design decisions that were made, alternatives that were considered and the rationale for those choices.

3.2. Analysis

This phase involved gathering and refining functional requirements for the CPAS system. Functional requirements describe the functionalities to be provided by the system. Informal discussions with the Distance and Continuing Education department staff helped identify initial set of requirements for the CPAS application. By examining existing documents and studying the paper-based process that is currently used to submit and gain approvals, functional requirements were fine-tuned. They were documented using the use case diagram, as shown in Figure 5, for obtaining feedback from stakeholders about the intended functionalities. The documented requirements served as the basis for another round of informal discussion in which the stakeholders were able to visualize the proposed system. This phase yielded an initial set of core, functional requirements as listed in Table 1. Requirements were prioritized based on the logical workflow sequence. For example, in CPAS application a proposal request must be created before it can be edited, deleted or printed. Proposal creation functionality was implemented followed by edit, delete and print functionalities.
Figure 5. High-level use case diagram for CPAS application.
Table 1. Use case description for CPAS application.

<table>
<thead>
<tr>
<th>No.</th>
<th>Actor</th>
<th>Action</th>
<th>Use case description</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>DCE staff</td>
<td>Create course proposal request.</td>
<td>DCE staff can create a course proposal by selecting a specific department and a term. In order to complete the proposal, a syllabus must be attached to the proposal.</td>
</tr>
<tr>
<td>2</td>
<td>DCE staff</td>
<td>Upload syllabus to course proposal request.</td>
<td>In order to complete the proposal user must select and upload a file either in Microsoft Word or PDF document format.</td>
</tr>
<tr>
<td>3</td>
<td>DCE staff</td>
<td>Add a course to create a course proposal request.</td>
<td>DCE staff may add a course to create a course proposal request.</td>
</tr>
<tr>
<td>4</td>
<td>DCE staff</td>
<td>Edit course proposal request.</td>
<td>DCE staff can make changes to existing course proposal requests.</td>
</tr>
<tr>
<td>5</td>
<td>DCE staff</td>
<td>Delete course proposal request.</td>
<td>DCE staff can delete course proposal requests created with erroneous information.</td>
</tr>
<tr>
<td>6</td>
<td>DCE staff</td>
<td>Add instructor.</td>
<td>DCE staff can add regular as well as adjunct instructors that are missing from CPAS application database.</td>
</tr>
<tr>
<td>7</td>
<td>DCE staff</td>
<td>Edit instructor information.</td>
<td>DCE staff can make modifications to individual instructor details as needed.</td>
</tr>
<tr>
<td>8</td>
<td>DCE staff</td>
<td>Upload instructor resume/CV.</td>
<td>DCE staff can upload instructor resume/CV for adjunct instructor.</td>
</tr>
<tr>
<td>9</td>
<td>DCE department head, Department chairpersons, College deans</td>
<td>Approve/disapprove course proposal request.</td>
<td>Approvers can make a decision on a submitted course proposal request.</td>
</tr>
</tbody>
</table>

As the DCE department staff was busy with their day-to-day work, scheduling formal interviews for eliciting CPAS application requirements was not an option. It was decided to...
conducted unstructured interviews at user’s convenience. These unstructured interviews didn’t work well as a number of functional requirements were left out. This became apparent during implementation phase. Missed requirements are listed below.

1. Functionality to add administrative documents to approved courses proposal requests
2. Ability to change department chairpersons.
3. Functionality to view all courses taught by an instructor by academic term.
4. Email notification for approvers that inform them of course proposal requests awaiting their action.

3.3. Design

An overall architecture for the CPAS application was defined during this phase. Database for the CPAS application was designed and important design decisions were considered and decided during this phase.

3.3.1. System Architecture for CPAS Application

The CPAS application was designed using the three-tier client-server architecture model, which is widely used in structuring web applications [7]. Client, application and data tiers encompass the three-tiers in CPAS application. Figure 6 shows the three-tier client server architecture model for CPAS application. In client server model a system is organized as a set of services and associated servers and clients that access and use the services [8]. In a client-server model, a server listens for requests from one or more clients. The server fulfills the client’s requests by providing a response. This request-response model forms the core of the client-server style architecture. The major motivation for choosing three-tier client server architecture for CPAS application is to centralize application processing and data storage. This centralization enables multiple users to access the application simultaneously.
3.3.2. Client Tier

In our CPAS system, users could use any internet browser application as client software to initiate a request to interact with the application. The application tier responds to this request by generating user interface elements and transferring them to the client. The client receives the response from the application server and knows how to render it. Another alternative to browser client would be a mobile app. There was no need from stakeholders to develop a mobile app front end for CPAS application.

3.3.3. Application Tier

Application tier includes application, data access and manipulation logic, user interface components and controller components. It handles and coordinates user requests to access and modify the underlying data store. In our CPAS application communication between the client tier and data tier must pass through the application tier.

Figure 6. Three-tier client server architecture model for CPAS application.
3.3.4. Data Tier

Data tier includes the database server that stores CPAS application data. In addition to storage space, a database server provides tools for querying and also manages access control to stored data.

3.4. Architecture Alternatives

Alternatives for structuring CPAS application are discussed below.

3.4.1. Repository Model

Figure 7. Repository architecture model.

CPAS application could have been designed using the repository architecture model. In this model, a central data repository is accessed and updated by several independent subsystems. Figure 7 shows the repository architecture model with subsystems connected to the central repository [8]. The Repository architecture model lacks the flexibility and scalability provided by client-server model. In an application based on client server architecture, processing and data storage could be assigned between clients and servers as needed. This flexibility is not available when using repository architecture model. Further, repository architecture model has been
discussed in various software engineering literature sources as more suited to large complex systems like Computer aided software engineering tools, computer aided design tools, artificial intelligence systems and software compilers tools [8][9]. For these reasons it was decided not to adapt repository model for CPAS application.

3.4.2. Two-tier Architecture Model

Figure 8. Two-tier architecture model.

CPAS could have been designed as a desktop application using the 2-tier architecture model as shown in Figure 8. In this model an application is organized as a server and a set of clients [8]. In this model, when most of the processing happens on the client, it is referred to as thick-client thin-server application. The server hosts the application data and responds to data access and retrieval requests. Application client is usually responsible for bulk of the processing. Client could be a graphical user interface residing on the desktop of user’s computers.

One of the main disadvantages of this model is distributing, installing and maintaining client software on each user computer that need to access the application. For this reason it was decided against using a two-tier architecture model for CPAS application.
3.4.3. Model 1 Architecture

The other alternative that was considered was to develop the CPAS application using the model 1 architecture [10]. Figure 9 shows the model 1 architecture components.

![Model 1 Architecture Components](image)

Figure 9. Model 1 architecture components.

In a web application implemented using this model, clients directly access Java Server Page (JSP) components residing on the application server. JSP technology enables web developers and designers to rapidly develop and easily maintain, information-rich, dynamic web pages that leverage existing business systems [11]. In this model, the JSP component invokes a java bean to access and manipulate the application data in the database. In addition to invoking java beans, the JSP component also includes logic to display the next view. Absence of a central controller component is one of the main drawbacks of this model. Further, presentation and application logic are intermixed in components residing on the server side. It is relatively simple to implement, but harder to extend and maintain the application using this model. For these reasons it was decided not implement CPAS application using model 1 architecture.

3.5. Application Architecture

CPAS is implemented using the java based struts application framework. The Java Programming Language is a general-purpose, concurrent, strongly typed, class-based object-
oriented language [12]. Java has rich set of libraries that enable development of desktop, client-server, enterprise and networking applications to name a few. A framework is a set of cooperating classes that make up a reusable design for a specific class of software. Struts framework uses the Model-View-Controller (MVC) design pattern and other proven design patterns. Figure 10 shows model-view-controller design pattern [13]. A design pattern is a general reusable solution to a commonly occurring problem within a given context. A design pattern names, abstracts, and identifies the key aspects of a common design structure that make it useful for creating a reusable object-oriented design [14]. Struts framework occupies the application tier in our three tier architecture design. Use of an application framework like struts has many advantages [14].

1. A Framework dictates the architecture of the application.

2. It defines the overall structure; it’s partitioning into classes and objects, the key responsibilities thereof, how the classes and objects collaborate, and the thread of control.

3. A framework predefines these design parameters so that the application designer/implementer can concentrate on the specifics of the application.

4. Framework captures the design decisions that are common to its application domain.

5. Frameworks are usually implemented using design patterns.
3.5.1. Model

In a MVC design pattern, model includes components that implement and enforce application, data access and manipulation logic. Further it represents the state of the application. In our CPAS application model contains courses, proposals, instructors, approvals, documents, departments and other domain objects.

3.5.2. Controller

In a MVC design pattern, Controller component coordinates requests between the view and model components. Struts framework provides a controller component named ActionServlet, which implements the front controller design pattern. This front controller pattern defines a single component that is responsible for processing application requests. A front controller centralizes functions such as view selection, security, and templating and applies them consistently across all pages or views [15]
In our CPAS application all requests from the client are handled by a single struts frameworks ActionServlet object. An ActionServlet is a java servlet. A servlet is a java programming language class used to extend the capabilities of servers that host applications accessed via a request response programming model [16].

The ActionServlet analyzes the request path and incoming request parameters. It consults a centralized configuration file to select and invoke the model component to perform the operation requested by the user and selects an appropriate user interface to return to the client that initiated the request.

3.5.3. View

In an MVC application, view incorporates components that render the user interface. Users interact with the application through this user interface to access services provided by the application. In our CPAS application user interface components are developed using the Java Server Pages (JSP) technology.

There are alternate MVC based frameworks available in java as well as in other programming languages. Developer familiarity and past experience in implementing web application using struts framework was the main motivation behind this choice. Besides this, struts is also a widely used popular MVC framework for implementing web applications.

3.5.4. Architecture Advantages

Choosing a 3-tier system architecture model with MVC design pattern for the application architecture helped design, develop and maintain the CPAS application in many ways.

1. Replacing or modifying components in one tier without the affecting the rest of the two tiers became possible as long as interface between the tiers remains the same. For
example, internet browser client software could be replaced with a mobile app client suitable for accessing the application using hand held devices.

2. Use of MVC design pattern allows for clear separation of presentation, application and data access logic. Further, a central controller component directs the flow of the application. Identifying and isolating application problems in this model is much easier when compared to an application implemented using model 1 architecture where application, presentation and data logic are intermixed.

3.6. Database Design

Using the functional requirements data entities, their attributes and the relationship among entities were identified for implementing the database for the CPAS system.

Table 2. CPAS application entities and their description.

<table>
<thead>
<tr>
<th>No.</th>
<th>Entity</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Course</td>
<td>A course is offered by NDSU DCE department taken by students to receive credits.</td>
</tr>
<tr>
<td>2</td>
<td>Proposal</td>
<td>A proposal is formal request to offer a course through DCE department.</td>
</tr>
<tr>
<td>3</td>
<td>Approver</td>
<td>An approver is a person authorized to approve proposal request initiated by DCE department staff.</td>
</tr>
<tr>
<td>4</td>
<td>Document</td>
<td>Details related to the proposal request uploaded by DCE staff</td>
</tr>
<tr>
<td>5</td>
<td>Instructor</td>
<td>An instructor is a person assigned to teach a DCE department course.</td>
</tr>
<tr>
<td>6</td>
<td>Department</td>
<td>A department is either an academic department or an administrative department at NDSU.</td>
</tr>
</tbody>
</table>

Figure 11 shows data entities and their relationship using entity-relationship diagram. Table 2 lists CPAS application entities and their descriptions. This database enables CPAS system to capture, process and store data pertinent to the application.
3.7. Implementation

A use case was chosen for implementation. Requirements for that specific use were analyzed to identify user interface, application logic and model objects required to realize the use case. The identified components were designed and implemented by specifying the responsibilities of each object and the sequence in which they communicate.

3.7.1. Use Case Implementation Illustration

Illustration of the create proposal request use case implementation is shown below.
3.7.2. Analysis

Create proposal request use case was analyzed to identify the user interface, application and model components needed to fulfill user request as shown in Table 3.

Table 3. Create course proposal request use case analysis components.

<table>
<thead>
<tr>
<th>No.</th>
<th>Component</th>
<th>Description</th>
<th>Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.</td>
<td>User Interface</td>
<td>This component allows the user to enter course proposal details and submit it to the CPAS application for processing.</td>
<td>JSP</td>
</tr>
<tr>
<td>2.</td>
<td>Application Logic</td>
<td>This object encapsulates application logic to enforce business rules and delegate proposal creation to data access object.</td>
<td>Java</td>
</tr>
<tr>
<td>3.</td>
<td>Data Transfer Object</td>
<td>This object is used by the application logic object to transfer user submitted data to data access object.</td>
<td>Java</td>
</tr>
<tr>
<td>4.</td>
<td>Data Access Object</td>
<td>In this use case, this object is mainly responsible for storing user submitted data in the database.</td>
<td>Java</td>
</tr>
<tr>
<td>5.</td>
<td>User Interface</td>
<td>This component displays a message indicating the status of course proposal request creation request imitated by the user.</td>
<td>Java</td>
</tr>
</tbody>
</table>

3.7.3. Design

User interface required in initiating a request to create and store a course proposal is quickly designed and refined using paper prototyping. Application and model objects identified during the previous use case analysis step were designed by specifying the responsibilities of each object and the sequence in which they communicate. Unified Modeling Language (UML) class diagram for create course proposal request use case is shown in Figure 12. Figure 13 shows the interaction among various components using UML sequence diagram for create course proposal request use case.

3.7.4. Implementation

Realization of the use case was accomplished by implementing objects as per the design.
Figure 12. UML class diagram for create course proposal request use case.

Figure 13. UML sequence diagram for create course proposal request use case.

3.7.5. Testing

The implemented functionality was tested for errors and inconsistencies. Chapter 4, section 4.2.2 discusses this in more detail.
3.7.6. Use Case Implementation Evaluation

User feedback was obtained to ensure that the implemented functionality is in line with user expectations. We have listed user feedback for create course proposal request implementation.

1. Input field to enter course topic for courses that have title and topic when creating course proposal requests.
2. Combine and present academic term and year when selecting to search courses offered by DCE department as opposed to having to choose term and year separately.
3. Users wanted to combine related multiple courses to create a single proposal as opposed to creating multiple proposals for each related course.
4. Class numbers must be displayed on create proposal request page.
5. Input field to enter variable credits for courses that don’t have fixed credit hour/s

Above listed user feedback were implemented in the following iteration of create proposal request use case.

3.8. CPAS Software Development Methodology

The software development activities and the sequence in which they were carried out resemble evolutionary prototyping model. In evolutionary prototyping an initial version of the system with well-understood requirements is quickly designed and developed to elicit user feedback. User feedback and lessons learned from the implementation are incorporated to iteratively and incrementally produce a fully functional application. Analysis, design, implementation, testing and user feedback activities are carried out in several iterations in evolutionary prototyping.
3.8.1. Evolutionary Prototyping

CPAS was designed and implemented as a custom workflow application to replace the manual paper-based process used at DCE department to create and approve course proposal requests. Developer exposure in implementing such workflow systems was limited. Some parts of the system implementation were not well understood at the start of the project. One such example would be electronic course proposal request approval workflow. Keeping track of course proposal request approval at each stage of the approval process would be another example. Further, users were unclear on what they wanted from the new system. For these reasons, it was decided to iteratively evolve the application by using prototyping and improving the prototype through user feedback until a satisfactory version of application was produced.

Agile models similar to exploratory prototyping places high priority on user feedback while incrementally and iteratively developing software. Agile models vary from evolutionary prototyping on the specifics of analysis, design, development and testing activities. For example, extreme programming recommends user stories, test first development, frequent releases, pair programming and on-site customer among others. Evolutionary prototyping differs from the incremental process model by the absence of number software increments and iterations needed to produce the software.

3.9. Design Decisions

During the course of the design and development of the CPAS application a number of design decisions were made. Important ones have been highlighted here.

3.9.1. Database Versus File System

One of the requirements for the CPAS application is to allow users to attach files to new as well approved course proposal requests. User submitted files together with identifying
information must be stored by the CPAS system and retrieved on demand. Two viable options were considered regarding the storage of these files in the backend.

1. Store user submitted files in the file system with identifying information in the database.

2. Store user submitted files with identifying information in the database server.

Establishing and maintaining relationship between user submitted files and other records in the database is much easier to implement when user submitted files are stored in the CPAS application database. Storing file metadata in the database and the actual file in the file system makes it harder, if not impossible to replace or delete files when compared with storing them in the database. Deleting or replacing a user submitted file in the file system is a two-step operation. File metadata record in the database must be modified before traversing the file system to locate and modify the actual file.

If user submitted files are stored in the database, same operation can be accomplished using structured query language facility provided by the database in a single atomic operation and it leaves no room for dangling records in the database. Also, storing files in the database adds an additional level of access control provided by the database. By storing all user submitted files in the database, an additional level of security is created that restricts unauthorized users to access and modify the files. Enforcing access rights on user submitted files is not a requirement at this time, but we anticipate this to be a requirement in the near future.

One major disadvantage of storing files in database might be performance degradation that might occur as the number of files and their size, stored in the database increases over time. This we predict, won’t become a bottleneck for CPAS application performance as present day
relational database management system capabilities to handle and manipulate large objects far exceed the size and number of files expected to be stored, retrieved and manipulated.

3.9.2. Authentication

Allowing access only to authenticated users is one of the implicit requirements for the CPAS application. User credentials must be accessible to the CPAS system in order to authenticate users that require access to the web application. Two alternatives for maintaining user credentials were considered. These alternatives differed in what was stored and where it was stored.

1. Store user name and password in the CPAS application database.

2. Use the Central Authentication Service (CAS) provided by the NDSU.

If we choose to maintain authentication credentials inside CPAS application, users might have to remember an additional set of user name and password. Password creation, distribution and change request adds another layer of unnecessary complexity. To bypass these difficulties it was decided to make use CAS provided by the NDSU.

NDSU CAS application, as the name implies provides authentication service to web applications. NDSU students, faculty and staff can utilize the same credentials, which they use to access computing resources by provided by NDSU to access the CPAS application. The CPAS application restricts access to unauthorized users by maintaining a list of users that are authorized to access the system.

Users that need access to CPAS application are forwarded to the CAS system with CPAS application name embedded in the forwarded URL. This request is received by the CAS server and it presents a login screen to the user. The CAS system takes the username and password supplied by the user to check its validity. If it is valid, users are forwarded to the CPAS web
application with a unique identifier. Before allowing access to the user, CPAS system makes another request to the CAS to validate the unique identifier it received. The CAS server validates the identifier, if valid, sends the username of the user that it authenticated earlier. CPAS web application matches the user name it received form the CAS server with the list of valid users stored in the application database. If there is a match, user is granted access to the system.

3.9.3. Data Loading

In order to create and manage course proposal requests the CPAS application is required to maintain data about courses offered, instructors assigned to each course, department chairs, divisional heads and other relevant information. The required data that need to be maintained by the application is public and freely available from multiple sources, but it has to compiled from various sources and imported into the CPAS application data tables. This is a cumbersome task and it demands substantial time, effort and manual labor.

After investigating and exploring for a better solution, it was decided to query and import required data directly from PeopleSoft campus administrative system instead of assembling from disparate sources. PeopleSoft provides an interface to compose and run queries to extract data. Queries were constructed on the PeopleSoft system to extract required data. The extracted data is transformed and loaded into database tables for use by CPAS application.
4. TESTING

4.1. Introduction

In chapter 1, we discussed the manual process that is currently used at NDSU DCE to create and approve course proposal requests. We proposed and developed course proposal and approval system application to replace the manual process. In chapter 3 we discussed the design and implementation of CPAS system. In this chapter we discuss functional testing activities that were carried out to demonstrate that the implemented CPAS application performs as intended.

4.2. CPAS Application Evaluation

The CPAS system is meant to replace the paper-based process currently in use at DCE department. DCE department staff currently uses the CPAS application to create and manage course proposal requests for courses offered through NDSU DCE. More than 570 course proposals requests have created during spring, summer and fall of 2012 academic terms. Implementing this system eliminated the need to manually type, print and route course proposal requests for approval.

In order to test the course approval system, select department chairpersons, college deans were asked to use the application to approve proposal requests created and electronically routed using the CPAS application. The implemented approval workflow process transitioned smoothly from one approver to the next approver while enforcing the order in which each approver must approve a course. As soon as each approver made a decision on a course proposal request, an email notification was triggered to the next approver in chain indicating them of the proposal waiting for their action. Email notification functionality was verified by checking the system logs. During any point in time in this online approval process, DCE department staff was able to track the approval process as it progressed through each stage of approval. All this evidence plus
the fact that the system has been in use for three consecutive academic terms, without any problems indicate that CPAS application is functioning as intended.

4.3. Functional Testing

Functional testing was carried out after each use case implementation. Implemented use cases were functional tested by itself and in combination with other previously implemented use cases. Important functional test cases for user login, create proposal request, approve proposal and edit instructor are shown below to highlight the testing activities carried while during iteratively and incrementally evolving the CPAS application.

4.3.1. Login

Test cases for testing CPAS application login are shown in Table 4.

Table 4. User login test cases.

<table>
<thead>
<tr>
<th>No.</th>
<th>Test case</th>
<th>Expected Result</th>
<th>Result</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Type URL of the CPAS application in the browser address bar.</td>
<td>User is forwarded to the NDSU Central Authentication Service (CAS) page, which displays login page for the user to submit username and password to gain access to the CPAS application.</td>
<td>Pass</td>
</tr>
<tr>
<td>2</td>
<td>Select a bookmarked CPAS application page.</td>
<td>User is forwarded to NDSU CAS page, which display the login page for the user to submit username and password to gain access to the CPAS application.</td>
<td>Pass</td>
</tr>
<tr>
<td>3</td>
<td>Enter valid NDSU username and password.</td>
<td>If submitted user login credentials match with CAS credentials, user is forwarded to the CPAS application home page.</td>
<td>Pass</td>
</tr>
<tr>
<td>4</td>
<td>Input invalid NDSU username and password.</td>
<td>The CAS login page is displayed with a message that authentication failed due to incorrect login is shown.</td>
<td>Pass</td>
</tr>
<tr>
<td>5</td>
<td>Disable cookies on browser. Log in using a valid username and password.</td>
<td>User is unable to login when cookies are disabled on the client computer browser.</td>
<td>Pass</td>
</tr>
</tbody>
</table>
4.3.2. Create Course Proposal Request

The following steps list the sequence of activities that users need to perform before testing course proposal request creation functionality. Test cases for testing create course proposal request functionality are shown in Table 5.

1. User logs in to the CPAS system.
2. User selects Proposals menu on home screen.
3. User chooses Term for which the proposal needed to be created.
4. The page reloads to display all departments in which courses are offered with total number of courses shown under completed, pending and in progress column headings.
5. User selects a specific department for which the proposal request needed to be created by activating the hyperlinked number shown in the third column on the same line as department.
6. User is forwarded to the page that shows all courses under that department for which a proposal request could be created.
7. User selects a particular course to create a proposal request for approval.
8. Course details for which a proposal could be created are displayed. User must attach a syllabus and select a value for Submit for Approval radio button field in order to submit the proposal. Course topic, course credits, variable credits, course start time, end time could be optionally modified by the user before submitting the proposal request creation request.
Table 5. Create Course proposal request test cases.

<table>
<thead>
<tr>
<th>No</th>
<th>Test Case</th>
<th>Expected Result</th>
<th>Result</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Submit proposal without uploading syllabus.</td>
<td>Error message indicating user to attach course syllabus is displayed.</td>
<td>Pass</td>
</tr>
<tr>
<td>2</td>
<td>Upload syllabus. Select “Now” for “Submit For Approval” radio button field. Submit to create proposal request.</td>
<td>Proposal request is created and available for approvers to make a decision.</td>
<td>Pass</td>
</tr>
<tr>
<td>3</td>
<td>Upload syllabus. Select “Later” for “Submit For Approval” radio button. Submit to create proposal request.</td>
<td>Course proposal request is created. It is not available for approvers to make a decision.</td>
<td>Pass</td>
</tr>
<tr>
<td>4</td>
<td>Select a value for both “Credits” as well as “Variable Credits” and submit to create proposal request.</td>
<td>An error message indicating user to select either “Credit” or “Variable Credit” but not both.</td>
<td>Pass</td>
</tr>
<tr>
<td>5</td>
<td>Select a value for “Credits” but not “Variable Credits” and submit to create proposal request.</td>
<td>Course proposal request is created.</td>
<td>Pass</td>
</tr>
<tr>
<td>6</td>
<td>Select “Variable Credits” but not “Credits” and submit to create proposal request.</td>
<td>Course proposal request is created.</td>
<td>Pass</td>
</tr>
<tr>
<td>7</td>
<td>Do not choose an option “Submit for Approval” radio button and submit to create proposal request.</td>
<td>Error message indicating user to choose a value for “Submit for Approval” radio button before submitting for processing it displayed</td>
<td>Pass</td>
</tr>
<tr>
<td>8</td>
<td>Upload a JPEG file for syllabus and submit to create proposal request.</td>
<td>Error message indicating user to upload a file either in Microsoft word or PDF format is displayed.</td>
<td>Pass</td>
</tr>
<tr>
<td>9</td>
<td>Upload a syllabus file that exceeds 2 megabytes and submit to create proposal request.</td>
<td>Error Message prompting the user upload a file that does not exceed 2 megabytes is shown.</td>
<td>Pass</td>
</tr>
<tr>
<td>10</td>
<td>Select Course End Time earlier than course Start time and submit to create proposal request</td>
<td>Error message prompting user that course ending time cannot be earlier than course starting time is displayed.</td>
<td>Pass</td>
</tr>
<tr>
<td>11</td>
<td>Select Course End Time after Course Start Time and submit to create proposal request.</td>
<td>Course proposal request is created.</td>
<td>Pass</td>
</tr>
</tbody>
</table>
4.3.3. Approve Course Proposal Request

At this point, a particular course proposal request has been created and submitted by Distance and Continuing Education staff at NDSU. Approvers can log in to view and make a decision on proposal request that they are authorized to act on. All requests are first approved by NDSU DCE department director followed by department chairperson, college dean and graduate dean, if it is a graduate level course proposal request. The following steps list the sequence of activities that users need to perform before testing course approval functionality. Test cases for testing course proposal request approval functionality are shown in Table 6.

1. Approver user logs in to the Course Approval System (CAS)
2. Home page is displayed with navigation menu.
3. Approver activates the proposal link on the navigation menu.
4. Approver chooses the term for which the proposal requests needed to be approved.
5. Department names with total number of approved and pending courses are displayed for that term.
6. Approver selects a pending link for a specific department.
7. Unapproved proposal requests for that department are shown with hyperlinked course title.
8. Approver selects a specific course to make a decision.
9. Course proposal request details with a list of all approvers for that course are displayed.
<table>
<thead>
<tr>
<th>No.</th>
<th>Test Case</th>
<th>Expected Result</th>
<th>Result</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Authenticated DCE department director approves a course proposal request by selecting approve button on the course proposal details page and hits submit</td>
<td>DCE department director is forwarded to the term summary page. The approved course is moved from pending to approved list. Course proposal request must now be available to the department chair to act on.</td>
<td>Pass</td>
</tr>
<tr>
<td>2</td>
<td>Authenticated department chairperson approves a course proposal request by selecting approve button on the course proposal details page and hits submit</td>
<td>Department chairperson is forwarded to the term summary page. The approved course is moved from the pending to approved list. Course proposal request must now be available to the college dean to act on.</td>
<td>Pass</td>
</tr>
<tr>
<td>3</td>
<td>Authenticated college dean approves a course proposal request by selecting approve button on the course proposal details page and hits submit</td>
<td>College dean is now forwarded to the term summary page. The approved course is moved from pending to approved list. Course proposal request approval is now complete.</td>
<td>Pass</td>
</tr>
<tr>
<td>4</td>
<td>Department chair logs in to the course approval system.</td>
<td>Only course proposal requests that fall under the department chair administration are displayed</td>
<td>Pass</td>
</tr>
<tr>
<td>5</td>
<td>College dean logs in to course approval system.</td>
<td>Only course proposal requests that fall under the college dean administration are displayed</td>
<td>Pass</td>
</tr>
<tr>
<td>6</td>
<td>Graduate dean logs in to course approval system.</td>
<td>Only graduate level course proposal requests that fall under the administration of graduate dean are displayed</td>
<td>Pass</td>
</tr>
<tr>
<td>7</td>
<td>Approver selects the View Syllabus hyperlink to view syllabus for the course that the approver is currently viewing to make a decision.</td>
<td>Approver can download and view the syllabus.</td>
<td>Pass</td>
</tr>
<tr>
<td>8</td>
<td>Approver makes a Do Not Approve decision on a course proposal request.</td>
<td>Course proposal request processing stops. The proposal request is not available to the next approver.</td>
<td>Pass</td>
</tr>
<tr>
<td>9</td>
<td>Execute 8. Check if disapproved course is counted as pending</td>
<td>Disapproved courses is counted as pending</td>
<td>Pass</td>
</tr>
<tr>
<td>10</td>
<td>Execute 8. Check if an email notification is sent to DCE staff regarding this disapproval</td>
<td>DCE staff confirms receiving email notification for disapproved course. System logs show outgoing email.</td>
<td>Pass</td>
</tr>
</tbody>
</table>
4.3.4. Edit Instructor

Test cases for testing edit instructor functionality are shown below in Table 7.

Table 7. Edit instructor test cases.

<table>
<thead>
<tr>
<th>No.</th>
<th>Test Case</th>
<th>Expected Result</th>
<th>Result</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Delete Title form field value and submit to modify instructor profile.</td>
<td>The page reloads prompting the user to input value for Title field</td>
<td>Pass</td>
</tr>
<tr>
<td>2</td>
<td>Delete Address1 form field value and submit to modify instructor profile.</td>
<td>The page reloads prompting the user to input value for Address1 field.</td>
<td>Pass</td>
</tr>
<tr>
<td>3</td>
<td>Delete Email form field value and submit to modify instructor profile.</td>
<td>The page reloads prompting the user to input value for Email field.</td>
<td>Pass</td>
</tr>
<tr>
<td>4</td>
<td>Delete Title, Address1, and Email form field values submit to modify instructor profile.</td>
<td>The page reloads prompting the user to input values for Title, Address1 and Email fields.</td>
<td>Pass</td>
</tr>
<tr>
<td>5</td>
<td>Edit Title, Address1, Email form field values and submit to modify instructor profile.</td>
<td>Instructor profile is saved with new values for Title, Address 1, and Email. Edit action success message is shown.</td>
<td>Pass</td>
</tr>
<tr>
<td>6</td>
<td>Input non-numeric values for Postal Code and submit the form to modify instructor profile.</td>
<td>The page reloads prompting the user to input valid value for Postal Code field.</td>
<td>Pass</td>
</tr>
<tr>
<td>7</td>
<td>Enter 5-digit numeric value for Postal Code and submit to modify instructor profile.</td>
<td>Edit action success message is shown</td>
<td>Pass</td>
</tr>
<tr>
<td>8</td>
<td>Upload a Microsoft Word or PDF file less than 5 megabytes for Vita/Transcript and submit to modify instructor profile.</td>
<td>Instructor profile has been edited successfully message is shown.</td>
<td>Pass</td>
</tr>
<tr>
<td>9</td>
<td>Upload Microsoft Word or PDF file exceeding than 5 megabytes for Vita/Transcript and submit to modify instructor profile.</td>
<td>File exceeds allowed size error message is shown.</td>
<td>Pass</td>
</tr>
<tr>
<td>10</td>
<td>Select Yes for Dept. Chair and submit form to modify instructor profile.</td>
<td>Message informing the user that a chair already exists for the department with the option to confirm currently edited instructor as new chair.</td>
<td>Pass</td>
</tr>
<tr>
<td>11</td>
<td>Select Yes to confirm currently edited instructor as department chair</td>
<td>Instructor update success message is displayed.</td>
<td>Pass</td>
</tr>
</tbody>
</table>
The following steps list the sequence of activities that users need to perform before testing edit instructor functionality.

1. User logs in to the CPAS system.
2. User selects Edit Instructors from the navigation menu
3. User is forwarded to the List Instructors by Department page
4. User selects a department.
5. A list of instructors in that department is displayed with chairperson listed at the top with an asterisk in front of chairperson’s name.
6. User selects a specific instructor to modify their profile.
5. CONCLUSIONS

5.1. Introduction

In the preceding chapters we discussed the design, development and testing of the CPAS application. The main motivation in developing CPAS application is to automate the manual course proposal request creation and approval process at NDSU DCE. This was accomplished by the design, development and implementation of the CPAS application. Section 5.1 talks about the merits of using the CPAS applications. Section 5.2 discusses the limitations of CPAS application. This is followed by suggestions on how to improve the CPAS application.

5.2. Results

DCE department staff currently uses the CPAS application to create and approve course proposal requests. Data for creating and approving these course proposal requests are queried, extracted, transformed and loaded from NDSU PeopleSoft administrative system into the CPAS application database. The CPAS application eliminates the need to print course proposal requests and supporting documents and routing it through campus mail for approval. Created proposals are immediately available to approvers for decision-making. When each approver approves a course proposal request, it is instantly available to the next approver in the approval chain. An email notification is sent to the next approver and so on until all approvers approve the proposal request.

Using the CPAS application allows DCE department staff to track proposal request approval status at any point during the approval process. Administrative documents related to the approved proposal request can be uploaded and saved by the DCE department staff. Being able to do so saves time, space and effort in archiving and retrieving proposal requests.
The CPAS application has been used in the DCE department since spring 2012. More than 570 course proposal requests for approval had been created and stored in the system. Select department chairs and college deans were requested to use CPAS application to approve proposal requests instead of the manual process. Approvers reported that were able to approve the courses without problems. DCE department staff was able to track the progress of course proposal request at each stage while each approver made a decision. Approvers who were asked to test-drive the system were in favor of the CPAS application over the manual paper process for approving course proposal requests. All these facts indicate that the CPAS application is functioning as intended.

5.3. Limitations

The following sections discusses limitations of CPAS application

5.3.1. Data Inconsistencies

CPAS application uses data from PeopleSoft campus administrative system to create proposal requests. PeopleSoft campus administrative system is used to query, retrieve and upload DCE course offerings data several weeks before the start of each semester. Courses offered by DCE are usually added, deleted and edited frequently in PeopleSoft system during the start of each term. CPAS application data must be manually updated to keep current with the course offering data stored in PeopleSoft system. Data inconsistencies between the CPAS application and PeopleSoft system could result in creating proposals request for courses that are deleted in the PeopleSoft system but not in CPAS application database. CPAS application currently doesn’t keep track of changes made to its database when it is updated with latest course offering data from PeopleSoft.
5.3.2. Mistaken Assumptions

Course approval functionality was implemented with the assumption that only department chairpersons approve courses that fall under that department. Another assumption was that a department chairperson is in charge of a single department. These two assumptions proved to be wrong. As the system was used for the past three semesters, it was learned that a department chair might temporarily be assigned to hold additional charge for another department. This initial assumption about department chairperson in charge of a single department proved to be erroneous. Also, it was learned later that one or more faculty members in addition to department chairperson could be authorized to approve course proposal requests. The initial assumption that only department chairpersons are authorized to approve course proposal requests proved to wrong too.

The current version the CPAS application doesn’t account for these mistaken assumptions. As a result of these assumptions, proposal request cannot be electronically approved for courses falling under a department where one or more faculty members in addition to department chairperson are charged with approving courses and when a department chairperson holds additional charge for another department.

5.3.3. NDSU Central Authentication Service

CPAS application uses central authentication service (CAS) provided by NDSU to authenticate users wanting to access the system. The CPAS application will be unavailable when NDSU CAS goes offline.

5.4. Future Improvements

The following sections suggest improvements for CPAS application in the future.
5.4.1. CPAS Application Integration with PeopleSoft

Integrating PeopleSoft system with CPAS application will eliminate the need to query, extract, transform, and load data from PeopleSoft system into CPAS application. This would allow the CPAS application to access PeopleSoft data in real-time as courses are added, modified and changed. Data inconsistencies arising out stale CPAS application data could be eliminated when the application directly accesses DCE department course offering data stored in PeopleSoft system.

5.4.2. Extending CPAS Application

NDSU faculty senate is responsible for approving, amending and deleting courses offered by NDSU. Faculty senate currently uses a paper-based process to create, route and approve proposal requests. The request to offer a new course is routed from one approver to the next approver until all approvers make a decision on the proposal. Faculty senate course approval process looks similar to CPAS application functionality.

CPAS application could be extended to handle faculty senate course approval requests. Thus CPAS application could both be used at the college level and at DCE department level to propose and approve course proposal requests.

5.4.3. Email Remainders

At present an email notification is sent to the first approver when a course proposal request is submitted for approval. This is to alert the user that course proposal request is waiting for his or her action. When this approver makes a decision next person in charge of approving the course proposal is notified of the waiting proposal. This Email notification continues until the last person approves the course.
Approvers could be busy and might fail to make note of pending proposal request email alert. This could introduce delays as course proposal requests could end up waiting for approvers’ decision indefinitely until the approver acts on the proposal. Periodic email notification to each approver summarizing pending course proposal requests could be implemented to overcome this problem.

5.4.4. Unified Interface

There are two types of users that currently use the CPAS application. DCE department staff that create and manage proposals and the course proposal requests approvers comprised of DCE department director, department chairs and college deans. Each of them uses a separate interface to access the system. Future work could involve implementing a single interface for CPAS users. The application would present the right interface based on the user type accessing the system.
REFERENCES


4. Google. Search for Course Proposal Systems Availability in Educational Institutions. Retrieved September 16, 2012 from https://www.google.com/ - hl=en&sclient=psy-ab&q=site:.edu+course+proposal+systems&oq=site:.edu+course+proposal+systems&gs_l=serp.3...4155.6308.0.6576.10.10.0.0.0.0.117.840.8j2.10.0.les%3B..0...1c.1.CqwnfzocuPA&pbx=1&bav=on.2,or.r_gc.r_pw.r_qf.&fp=e0911ht


14. E. Gamma, R. Helm, R. Johnson, J. Vlissides. Design Patterns - Elements of Reusable Object-Oriented Software. Addison-Wesley, 1995, Boston, MA


APPENDIX A. COURSE PROPOSAL SYSTEM

Figure A1. NDSU central authentication login page for CPAS application.

Figure A2. NDSU central authentication service invalid login result.
Figure A3. Course proposal system homepage.

Figure A4. List course proposals by term.
Figure A5. Course proposal summary page.

Figure A6. Fall 2012 accounting department course proposal request listing.
Figure A7. Print view of ACCT 200 course proposal request.

<table>
<thead>
<tr>
<th>REQUEST FOR APPROVAL</th>
<th>DISTANCE &amp; CONTINUING EDUCATION</th>
</tr>
</thead>
<tbody>
<tr>
<td>TO: Lisa Nordick, Assistant Dean</td>
<td>William Bowlin, Former Chair/and/or Dept. Head</td>
</tr>
<tr>
<td>Distance &amp; Continuing Ed.</td>
<td>Accounting &amp; Info Systems</td>
</tr>
<tr>
<td>Donald Johnson, Dean</td>
<td>Distance Office, Business Admin.</td>
</tr>
</tbody>
</table>

| FROM: Karan Marie, Degree Credit, Great Plains IDEA, and Scheduling Coordinator |
| TITLES: Elements of Accounting I, ACCT 200 (5985) |
| CREDITS: 3 Undergraduate Credit(s) |
| DELIVERY: Online |
| TIMES: NA |
| LOCATION: Online |
| GRADE: Five Category (A-F) |
| STATUS: New Course | Repeat | Last Approved |

INSTRUCTOR: James Gatt, Senior Lecturer, Accounting & Info Systems, Richard E. Barry Hall 202
701-031-0820 W
james.gatt@ncsu.edu
NCSU, 1985

Please indicate your approval or disapproval of the proposed course and instructor by checking the appropriate space below, making explanatory remarks as necessary, and forwarding this approval to the next person listed. Courses numbered 500, once approved, are good for three years before needing re-approval. Catalog courses generally require annual approval.

<table>
<thead>
<tr>
<th>Approved*</th>
<th>Do not approve*</th>
</tr>
</thead>
<tbody>
<tr>
<td>Approved*</td>
<td>Do not approve*</td>
</tr>
<tr>
<td>Approved*</td>
<td>Do not approve*</td>
</tr>
</tbody>
</table>

*Comments/reasons:

Please return to Distance & Continuing Education, 1919 North University Drive
Back to Download Status
Figure A8. Pending courses in ADHM department for which proposal requests have not yet been fully entered.

Figure A9. Create proposal request page for ADHM 710.
Figure A10. Course proposal requests completed for accounting department courses.

Figure A11. Course proposal request approval status page for ACCT 201.
Figure A12. Approved course proposal page.
Figure A13. Combine courses to create a single proposal for related courses.

Figure A14. Create a combined proposal for BIO/BOT/PLSC/ZOO.
Figure A15. Add a course that is currently unlisted in PeopleSoft system.

Figure A16. Select an instructor type to create profile.
Figure A17. Create a profile for an NDSU instructor.

Figure A18. Create profile for adjunct instructor
Figure A19. List instructors by department page.

Figure A20. Instructor listing for computer science department.
Figure A21. Edit instructor profile page.

Figure A22. View courses taught by instructor for a term.
Figure A23. Reassign department chairperson page.

Figure A24. Reassign department chairperson confirmation page.
Figure A25. Reassign department chairperson success page.

Figure A26. Instructor listing page reflecting chairperson change.
### Figure A27. Courses proposal requests page with edit and delete options.

<table>
<thead>
<tr>
<th>Date</th>
<th>Number</th>
<th>Title</th>
<th>Instructor</th>
<th>Action</th>
</tr>
</thead>
<tbody>
<tr>
<td>08-21-12</td>
<td>CSCI 114</td>
<td>Microcomputer Packages</td>
<td>Katala</td>
<td>View</td>
</tr>
<tr>
<td>08-21-12</td>
<td>CSCI 116</td>
<td>Business Use of Computers</td>
<td>Schatz</td>
<td>View</td>
</tr>
<tr>
<td>09-21-12</td>
<td>CSCI 122</td>
<td>Visual BASIC</td>
<td>Nynsooych</td>
<td>View</td>
</tr>
<tr>
<td>09-21-12</td>
<td>CSCI 139</td>
<td>Computer Science/Problem Solving</td>
<td>Katala</td>
<td>View</td>
</tr>
<tr>
<td>10-09-12</td>
<td>CSCI 143</td>
<td>Software Development Processes</td>
<td>Magret</td>
<td>Edit Del</td>
</tr>
<tr>
<td>09-21-12</td>
<td>CSCI 247</td>
<td>Software Complexity Metrics</td>
<td>Do</td>
<td>View</td>
</tr>
</tbody>
</table>

<p>| | | | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
</table>

### Figure A28. Edit a saved course proposal request.
Figure A29. Choose term to view course catalog.

Figure A30. Course catalog listing page.
APPENDIX B. COURSE APPROVAL SYSTEM

Figure B1. Course approver login page.

Figure B2. DCE department director home page.
Figure B3. DCE department director home page showing a list of approved and pending proposals.

Figure B4. DCE department director view of pending proposals.
Figure B5. DCE department director view of proposal request detail page.

Figure B6. Course proposal listing page indicating an approved course has been moved from pending to approved status.
Figure B7. Computer science department chairperson view of approved and pending proposals page.

Figure B8. Computer science department chairperson view of a submitted proposal request.
Figure B9. Computer science department chairperson approving a course.

Figure B10. College of math and science dean view of submitted proposal request details page.
Figure B11. College of math and science dean approving course after DCE department director and computer science department chairperson approval.

Figure B12. Graduate dean approving course after DCE department director, computer science department chairperson and college of math and science dean approval.