electronic Proposal Application System
Software Requirements Specification

Version 1 Draft 3
05/06/13

Project Director: Annalee Werkmeister, Administrative Associate

Project Manager: Celia Schahczenski, Professor of Computer Science

Clients:
Bev Hartline, Vice Chancellor of Research and Dean of Graduate School
Amy Kuenzi, Professor of Biology and Dept. Chair
Keith Vertanen, Professor of Computer Science
Annalee Werkmeister, Administrative Associate

Document Authors:
Cade Foster, Student of Software Engineering
Jake Jones, Student of Software Engineering
Matt Morris, Student of Software Engineering
Celia Schahczenski, Professor of Computer Science
Frank Sholey, Student of Software Engineering
Alex Thompson, Student of Software Engineering

Version History

Version	*Date*	*Authors*	*Comment*

TABLE OF CONTENTS

1 Introduction

This Software Requirements Specification (SRS) describes the required functionality of the electronic Proposal Application System, ePAS. In this first section, ePAS is described from a high level, the purpose and layout of this document are given and terms and abbreviations are defined.

1.1 Software Purpose and Scope

The ePAS will allow users to track and manage grant requests during the grants proposal process. The existing software, a Microsoft Access database and interface, has been ineffective due to incorrect reports and poorly structured queries. This product will replace the Access database and improve work flow for the proposal office. It also tracks completion of the proposal certification process, through development, signing, funding results, closing and archiving.

The goals of ePAS include:

- Allow investigators to electronically submit proposals for grants, along with supporting documents, in a standardized fashion.
- Store these proposals and facilitate their management by the Office of Research.
- Allow searches of proposal records by the Chancellor, Provost, Deans, and Department Heads.
- Enable Research Office personnel to print reports.

The tracking of financial information associated with a proposal is handled in another system called Banner Contracts. This SRS assumes that these systems will be independent of each other. If a decision is made for ePAS to interface directly with Banner Contracts, significant portions of this SRS would need to be re-written.

1.2 Document Purpose and Contents

This SRS describes a system that assists faculty, administration, and the staff of the Office of Research in managing and tracking grant proposals.

The purpose of this document is to ensure that the product developed is what the clients specified. This document shall contain all necessary functional and non-functional requirements for the Grants Database. This document also details the program’s typical use cases and user characteristics, as well as the operating environment and any necessary external interfaces.

This document precisely describes all functionality needed in Grants Database. In this way, this document can serve as a contract between the customers and developers. Developers will find this document very helpful. Testers and final users may also find this document helpful. Customers sometimes find sample interfaces easier to understand than documents such as this SRS. Sample user interfaces demonstrate one way that the software could appear. This document goes further to tell precisely what functionality is needed.

This document does not attempt to tell how this software should be implemented except in those cases where the customers want the application to be developed in a particular way. Deciding exactly what a system should do before deciding how it will do it reduces development time considerably.

1.3 Definitions, Acronyms, and Abbreviations

1.3.1 Definitions

Active Project	
Banner	
Co-Principal Investigator	
Fringe Benefits	
Grant	
Indirect Cost	
Pre-Proposal	
Principal Investigator	
Proposal	
Proposal Record	
Research Office	Research Office and Graduate Studies

1.3.2 Acronyms and Abbreviations

Co-PI	Co-Principal Investigator
DB	Database
IDC	Indirect Cost
ORGS	Office of Research and Graduate Studies
PI	Principal Investigator
RFP	Request for Proposal
SRS	Software Requirements Specification

1.3.3 Technical Definitions/Data Dictionary

Item Name	Type	Brief description of data item
ePAS	System	The database storing all information associated with proposals and PIs.

1.4 References

Add reference to the research office.

2 General Factors

This section describes the perspective, functions, environment, and user characteristics of the Grants Database.

The final software will be dependent on the pre-proposal grant protocol and if that changes, the software will have to change as well.

2.1 Product Perspective

EPAS may be dependent on a Montana University System Banner Proposal module and Montana Tech human resource database. The Research Office is in the process of looking into the Banner proposal module to determine if this system should interface with it. It is also still undecided as to whether faculty and department information will be maintained separately in this database application or if the information will be synchronized with the Tech human resource database.
EPAS serves three basic user groups: Researchers, Administrators, and Research Office Personnel. Below is a brief list of the functions available to each user class.

Researcher:

- Create Proposal
- Upload Document
- Sign Proposal
- Update Profile

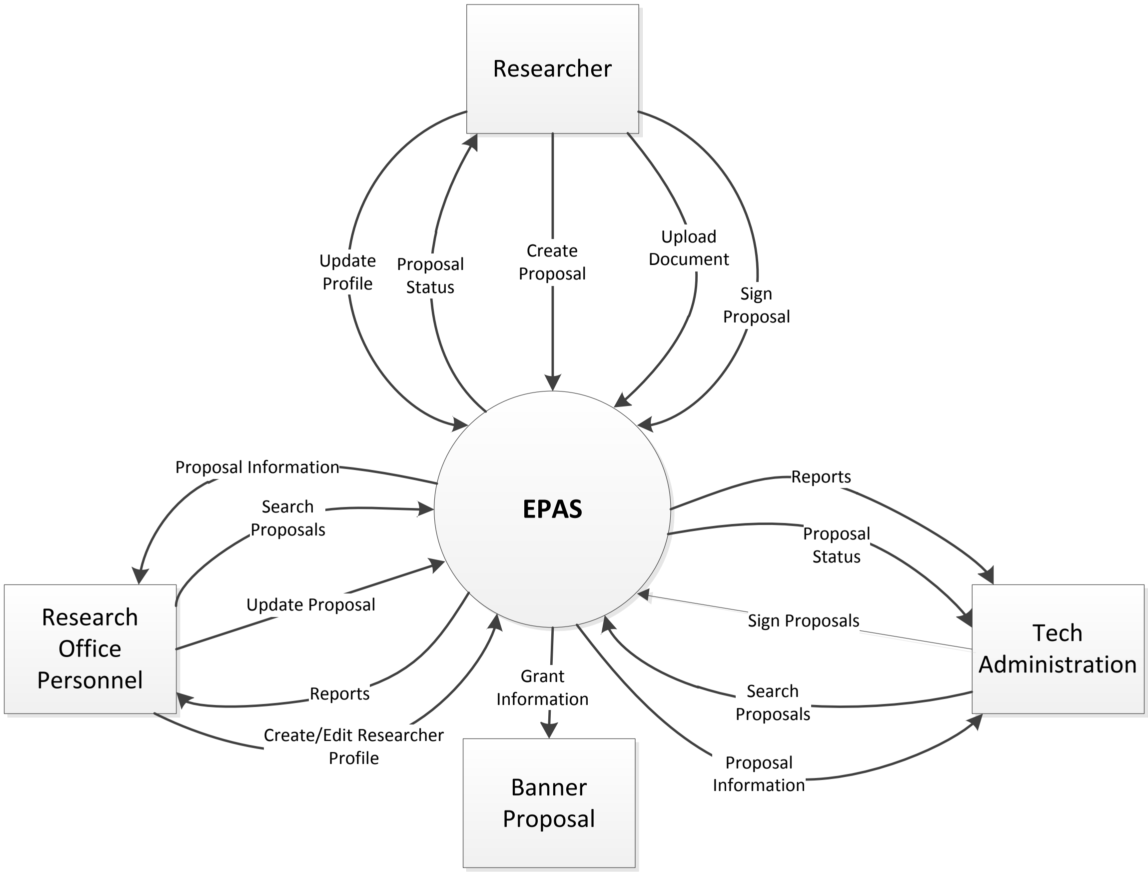
Administrator:

- Researcher Capabilities, and
- Generate Reports
- Search Proposals

Research Office Personnel:

- Edit Researcher Profile
- Generate Reports
- Search Proposals
- Update Proposal\

2.1.1 Context Diagram



2.2 Product Functions

The Grants Database will be used to:

- Facilitate/ease PI submission of PCF and related documents.
- Populate database from electronic submissions from PIs.
- Store and manage (add, edit, delete) information about individual proposals, including:
  - Proposal abstract.
  - PDF document of actual proposal (even grants.gov proposals)
  - Proposal ID number
  - Proposal date
  - Title

- Funding agency
- Start date
- End date
- Total requested amount
  - Direct cost
  - Indirect cost
- Match funding(Y/N) or (T/F) and amount
- Reinvested IDC requested (Y/N) or (T/F) and amount
- Announcement date
- Status
  - Pending/Declined/Approved/Closed
- human subjects (IRB)

- Store and manage (add, edit, delete) information about individual PIs and Co-PIs, including:
  - Safety training/certification of all participants
  - up-to-date conflict of interest
  - First and last name
  - Employee ID number
  - Department
  - College/Institute
  - Affiliation
  - % of funding per PI/Co-PI of total requested
  - Awarded amount

Report: submissions/dollars requested by date range
Report: Actual awards versus amount requested

The program should be able to perform the following functions with the above data:

- Reporting features:
  - Create proposal summary page for a proposal with its pertinent information.
  - List active projects to date
  - List proposals by department and status
  - List proposals by PI and status
  - List proposals by date
  - List proposals by funding source and status
  - List by status
- Search by:
  - PI/Co-PI
  - Department
  - College/Institution
  - Funding Source
  - Date, including range of dates

- Keyword (in title)
- Banner Number
- Contract Number

2.3 User Characteristic

This section describes the expected users of the system, how often they are expected to use the system and the features that they will use.

Actors/Roles	Frequency	Summary of Features Used
Dean	Monthly/Quarterly	Search for information and view reports relating to proposals in which faculty in their college are PI or Co-PI. Sign such proposals.
Department Chair	Monthly/Quarterly	Search for information and view reports relating to proposals in which faculty in their department are PI or Co-PI. Sign such proposals.
PI/researcher	Few times each time a proposal is submitted	Submit, update and view proposals.
Required signatures	Infrequently	Sign a proposal.
Research office personnel	Daily	Update proposal and researcher information, search for information and view reports.
Tech administrator	Monthly/Quarterly	Search for information and view reports relating to. Sign such proposals.

2.4 Dependencies

This application may be dependent on the Banner system and on the Montana Tech human resource database.

3 Analysis Use Cases

This section describes how users will interact with the system. Users interact with the system as "actors" playing a role to achieve a useful goal.

Primary Actor	*Use Cases*
Research Office Personnel	1. Create Researcher Profile 2. Create/Delete/Submit Proposal Record 3. Create/Update/Delete Department, Colleges, Funding Agencies 4. Generate Report -PI, proposal, requested, granted amounts 5. Search for Proposal Record 6. Update Proposal Record 7. Update Researcher Profile 8. Upload Supporting Document 9. View Proposal Record 10. View Researcher Profile
Researcher	1. Add Signature to Completed PCF 2. Create/Delete/Submit Proposal Record 3. Update Own Profile 4. Update Proposal Record 5. Upload Supporting Documentation 6. View Proposal Record
Department Chair / Dean / Tech Administrator	1. Add Signature to Completed PCF – low priority 2. Generate Report - PI, proposal, requested, granted amounts 3. Search for Proposal Record 4. View Proposal Record 5. View Researcher Profile
Required Signatories	1. Add Signature to Completed PCF – low priority

The use cases below carefully describe interactions with the system.

3.1 Add Signature to Completed PCF

Use Case Name:	Add Signature to Completed Proposal Record		
Created By:	ESOF 328 s13	Last Updated By:	Jake
Date Created:	4/22/13	Date Last Updated:	4/22/13

Actors:	Required Signatories
Description:	After following a link included in an automated e-mail notification, the Actor can sign into the system and electronically sign a PCF.
Trigger:	Actor follows link from signature solicitation e-mail.
Preconditions:	1. System has sent automated signature solicitation e-mail to Actor.
Postconditions:	1. Actor has electronically signed PCF.
Normal Flow:	**1.0 Actor signs completed PCF.** 1. If Actor is not logged in, this triggers the *Log In* use case. 2. System displays proposal record that the Actor is to sign. 3. Actor electronically signs proposal record's PCF. 4. System saves modified proposal record and notes addition of a required signature. 5. Actor logs out.
Alternative Flows:	**1.1 Actor chooses to decline to sign** (branch after step 2). 1. System prompts Actor for the reason they have declined to sign. 2. Actor provides reason and confirms. 3. System notifies PI that a Required Signatory has declined to sign and includes Actor's explanation provided in (2), above. 1. Actor logs out.
Exceptions:	none
Includes:	*Log In*
Priority:	Imperative
Frequency of Use:	Few times a year.
Business Rules:	See Business Rule [section]: Order of signatures
Special Requirements:	See Functional Requirement [section]: Signature Solicitation
Assumptions:	Actor exists in the system.
Notes and Issues:	

3.2 Create/Delete/Submit Proposal Record

Use Case Name:	Create/Delete/Submit Proposal		
Created By:	ESOF 328 s13	Last Updated By:	Jake
Date Created:	2/23/13	Date Last Updated:	4/22/13

Actors:	Researcher
Description:	User can create a proposal record and begin gathering signatures once all required fields are completed. Alternatively a proposal record can be saved at any time, to be completed later. The proposal itself will need to be

	uploaded (see the *Upload Supporting Documentation* use case). The User can delete a proposal record before signature gathering has begun. The User chooses whether to gather signatures manually or use the system's electronic solicitation functionality.
Trigger:	User chooses to create a new proposal record
Preconditions:	1. User is logged in to the system
Postconditions:	1. The proposal record is ready for signatures, or saved, or deleted.
Normal Flow:	**1.0 User fills out PCF and finishes, including only Co-PIs that exist in the system.** 1. System displays PCF. 2. User fills in PCF fields. For Co-PIs and departments, User may select an entity that already exists in the system rather than entering a name. 3. User attaches PDF/Doc of Proposal according to the *Upload Supporting Documentation* use case. 4. User chooses to submit PCF. 5. System verifies that all necessary fields have been completed. 6. System prompts for choice of signature gathering: manual or electronic solicitation. 7. User chooses to solicit signatures electronically. 8. System returns User to previous screen/page. 9. System stores PCF in database and begins solicitation of signatures. 10. System notes any requirements for *Conflict of Interest* or *PI Training*. 11.
Alternative Flows:	**1.1 User doesn't submit the form** (branch after step 1, 2, or 3) 1. User chooses to save form. 2. System confirms form has been saved. 3. User logs out. **1.2 User includes the name of a Co-PI that does not exist in the system** (branch during step 2). 1. System displays area for PI to enter contact information, address and affiliation of Co-PI. 2. User enters Co-PI's information into fields. 3. Return to step 2. **1.3 User chooses to cancel without saving** (branch after step 1, 2, or 3) 1. User chooses to cancel and discard changes. 2. System prompts User to confirm cancel without saving. 3. User confirms cancel without saving. 4. System returns to previous screen/page. **1.4 User deletes proposal record** (branch after step 1) 1. User chooses to delete proposal record. 2. System asks User to confirm delete proposal record. 3. User confirms delete proposal record. 4. System returns to previous screen/page. **1.5 User attaches additional supporting documentation to proposal record** (branch after step 1 or 2). 1. User uploads documents according to the *Upload Supporting Documents* use case. 2. Return to step 2. a. **User chooses to gather signatures manually** (branch after step 6).

	1. System presents User with ability to print PCF. 2. User prints PCF. 3. Return to step 9.
Exceptions:	none
Includes:	*2. Upload Supporting Documentation*
Priority:	Imperative
Frequency of Use:	Once or twice a year
Business Rules:	See Business Rule [section]: Order of signatures
Special Requirements:	See Functional Requirement [section]: Signature Solicitation
Assumptions:	User exists in the system
Notes and Issues:	Co-PI and department fields may not be the only ones to feature selection in addition to editing.

3.3 Create Researcher Profile

Created By:	Frank and Jake	Last Updated By:	Frank and Jake
Date Created:	4/7/13	Date Last Updated:	4/7/13

Actors:	Office of Research Personnel
Description:	The Office of Research Personnel can create a profile for a new researcher that includes first and last name, employee ID number, department, college/institute, affiliation, current and previous proposals, and total amount of money awarded to date.
Trigger:	Office of Research Personnel chooses to create a new researcher profile.
Preconditions:	1. User is logged in to the system
Postconditions:	1. The researcher profile is created and saved into the system.
Normal Flow:	**1.0 User creates a new researcher profile and fills in the necessary fields.** 1. System displays a form with the necessary fields to fill out. 2. User fills in fields. 3. User chooses to save the researcher profile into the system. 4. System confirms researcher profile has been saved. 5. System stores researcher profile in the database. 6. User chooses to quit editing researcher profile. 7. System returns to previous page.
Alternative Flows:	**1.1 User chooses to cancel progress without saving** (branch after step 2) 1. User chooses to quit editing researcher profile. 2. System asks user to confirm cancel without saving. 3. User confirms to quit and discard all changes. 4. Return to step 7. **1.2 User saves partially completed form** (branch during step 2) 1. User chooses to save without filling out all fields. 2. System warns user that the form is not complete, but still gives the user the choice to save as is and quit. 3. User chooses to save the form. 4. System saves the partially completed form. 5. Return to step 6.
Exceptions:	

Includes:	
Priority:	Imperative
Frequency of Use:	Every time a new researcher needs to be added to the system.
Business Rules:	none
Special Requirements:	none
Assumptions:	1. The researcher does not already exist in the system. 2. In alternate flow 1.2, the user can finish filling out the partially completed form at a later time.
Notes and Issues:	

3.4 Generate Report

Created By:	ESOF 328 s13	Last Updated By:	Alex
Date Created:	2/23/13	Date Last Updated:	4/1/13

Actors:	Research Office Personnel, Administrator
Description:	The process of generating a report of summarized data.
Trigger:	One of the aforementioned actors decides to generate a report
Preconditions:	1. The user is logged in to the system.
Postconditions:	None
Normal Flow:	**1.0 User Generates a Report** 1. User navigates to report section of software. 2. User selects report type and clicks "generate." 3. System displays report with proposal data. 4. User exits the report viewing screen.
Alternative Flows:	**1.1 User Decides to Save Report** (branch after step 3) 1. User chooses to save report. 2. System asks user where they would like to save the report to. 3. System confirms that report has been saved. 4. Return to step 1. **1.2 User Decides to Print Report** (branch before or after step 3) 1. User chooses to print report. 2. User chooses printer to send print job to. 3. System sends report to printer. 4. Return to step 3.
Exceptions:	None
Includes:	None
Priority:	High
Frequency of Use:	Monthly
Business Rules:	None
Special Requirements:	None
Assumptions:	Connection to a printer is available
Notes and Issues:	Result of these will be summarized data Reporting features: • Create proposal summary page for a proposal with its pertinent information. • List active projects to date

	• Reports(include report date) • Canned Reports- number of proposal submitted, proposals funded, total money requested, total money approved,

3.5 Search for Proposal Record

Created By:	ESOF 328 s13	Last Updated By:	Frank
Date Created:	3/10/13	Date Last Updated:	4/23/13

Actors:	Research Office Personnel, Administration
Description:	The process by which a user searches for a proposal record and receives a list of proposal records matching the search criteria. The user can select an individual proposal from the search results to perform some action on, triggering another use case such as *Update Proposal Record or Delete Proposal Record*.
Trigger:	User chooses to search for proposal record(s).
Preconditions:	1. User is logged into system.
Postconditions:	1. System has displayed a list of proposal records pertaining to the search criteria.
Normal Flow:	**1.0 User chooses search criteria, enters values, and searches for proposal records** 1. System displays search criteria. 2. User enters values to be searched for. 3. User initiates search. 4. System displays search results (The result may be the empty set).
Alternative Flows:	**1.1 User searches again to further filter results (branch after step 4)** 1. User chooses to search again to filter the results even more. 2. The system displays the original search screen and displays the previously entered values. 3. User chooses to either change the previously entered values or enter new values. 4. Return to step 3. **1.2 User changes how results are sorted (branch after step 4)** 1. User chooses to change how the results are sorted. 2. User chooses new field to sort the results by. 3. System displays the results sorted as specified by the user. 4. Return to step 4. **1.3 User exports results (branch after step 4)** 1. User chooses to export results to a Microsoft Excel file. 2. User chooses where to save the file. 3. System prompts user for the file name to save the results to. 4. User saves the file. 5. Return to step 4.
Exceptions:	**1.3 E.1 Excel file name already exists.** 1. System displays message: “That file name already exists.” 2. System asks user whether he wants to overwrite the existing file, enter a new file name, or cancel. 3a. User selects to overwrite the existing file.

	4a. User saves the file. 5a. Return to step 4 of the normal flow. 3b. User enters a new file name. 4b. User saves the file. 5b. Return to step 4 of the normal flow. 3c. User selects to cancel. 4c. Return to step 4 of the normal flow. **1.3 E.2 User chooses an illegal path on where to save the file** 1. System displays message: "Path is illegal." 2. System asks user whether he wants to change the path or cancel. 3a. User changes the path. 3b. Return to step 4 of alternative flow 1.3 4a. User selects to cancel. 4b. Return to step 4 of the normal flow.
Includes:	
Priority:	Imperative
Frequency of Use:	Frequently
Business Rules:	
Special Requirements:	1. The search results will consist of the following fields: a. Title b. PI c. PI's Department d. Funding Source e. Submittal Date f. Total Amount Requested g. Status h. An indication to display the entire proposal record. 2. The system will filter the search results based on who is logged in: a. Research Office Personnel will be able to search for every proposal record. b. PIs and Co-PIs can only search for a proposal record that they have been involved with. c. Deans can search for proposal records relating to proposals in which faculty in their college are PIs or Co-PIs. d. Department Chairs can search for proposal records relating to proposals in which faculty in their department are PIs or Co-PIs.
Assumptions:	1. Search results are assumed to be initially sorted by most recent date.
Notes and Issues:	

3.6 Update Own Profile

Actor: Researcher
Description: Change data associated with researcher's own profile.
Comments: What information are researchers allowed to change? Is there any information that Research Office Personnel can't change?

3.7 Update Proposal Record

Use Case Name:	Update Proposal Record		
Created By:	ESOF 328 s13	Last Updated By:	Jake Jones
Date Created:	2/23/13	Date Last Updated:	4/22/13

Actors:	Researcher, Research Office Personnel
Description:	The process to update (modify values in, status of, or documents associated with) a proposal in the database.
Trigger:	Actor chooses to view a proposal record and has permission to edit it.
Preconditions:	1. Actor has located a proposal record. 2. Actor has permission to edit the proposal record
Postconditions:	2. The proposal record is unchanged or updated.
Normal Flow:	**1.0 Actor updates proposal record** 1. System displays fields associated with proposal. 2. Actor edits one or more fields in the proposal record. 3. Actor chooses to save changes. 4. System saves changes. 5. Actor chooses to quit editing proposal record. 6. System returns to previous page.
Alternative Flows:	**2.1 Actor updates Proposal and Cancels** (branch after step 2). 1. Actor chooses to cancel update. 2. System prompts Actor to confirm cancel without saving. 3. Actor confirms cancel without saving. 4. Return to step 3. **2.2 Actor attaches supporting documentation to proposal record** (branch after step 1 or 2). 1. Actor chooses to upload supporting documentation. This triggers use case *Upload Supporting Documents.* 2. Return to step 1. **1.3 Another user is currently updating this proposal** (branch after trigger) 1. System displays dialog informing Actor that another user is currently updating this Proposal Record, and allowing the Actor to either view the proposal or cancel. 2. If the Actor chooses to view the proposal, this is a trigger for use case *View Proposal*. If the Actor chooses to cancel, return to step 4. **1.4 Proposal record's status is changed when Actor saves** (branch after step 4). 1. System sends notification of status change to this proposal record's PI and Co-PIs. If the status has changed to "Accepted by Research Office," a copy of the PCF (and associated signature files, if necessary) is included with the notification to the PI. 2. Return to step 5.
Exceptions:	None
Includes:	*View Proposal* *Upload Supporting Documents*
Priority:	Imperative
Frequency of Use:	Quarterly
Business Rules:	Research Office Personnel can edit the proposal record at any time.

	Researchers can edit proposals on which they are PIs or Co-PIs and whose status is 'Draft' or 'Submitted to Research Office."
Special Requirements:	
Assumptions:	None
Notes and Issues:	None

3.8 Upload Supporting Document

Created By:	Matt and Jake	Last Updated By:	Matt
Date Created:	3/5/13	Date Last Updated:	4/2/13

Actors:	Research Office Personnel, Researcher
Description:	The process to upload and tag a supporting document associated with a proposal record.
Trigger:	User requests to attach a document.
Preconditions:	1. Proposal Record exists in database or is being added to the database. 2. Proposal Record has been located by the user.
Postconditions:	1. Document is tagged and attached to proposal.
Normal Flow:	**1.0 User uploads document.** 1. User chooses to upload supporting document. 2. System requests path to file. 3. User provides file path. 4. User confirms selection. 5. User tags document with an existing tag or "Other" If "Other" is chosen, the user must also enter a tag name. 6. System uploads file and attaches the file to proposal record. 7. User confirms upload.
Alternative Flows:	**1.1 Document with Tag Already Exists** (branch after step 5) 1. System alerts user of conflicting documents. 2. User chooses to overwrite and remove the previous document (return to step 6), or not upload the new document (return to step 2). **1.2 User Cancels After Document Upload** (branch after 6) 1. User cancels process after uploading document. 2. System removes file from Proposal Record.
Exceptions:	Document Not Found
Includes:	
Priority:	Medium
Frequency of Use:	
Business Rules:	
Special Requirements:	
Assumptions:	
Notes and Issues:	Included in proposal creation and update Tags: Proposal Cover Sheet Proposal Researcher Bio Budget SF 424 SF LLL Data Management Plan Other

3.9 Update Researcher Profile

Actor: Office personnel, Researcher
Description: Change data associated with researcher.
Comments: What information are researchers allowed to change? Is there any information that Research Office Personnel can't change?

3.10 View Proposal Record

Created By:	ESOF 328 s13	Last Updated By:	Cade
Date Created:	2/23/13	Date Last Updated:	04/08/13

Actors:	Researcher, Research Office Personnel, Administrator
Description:	The process to view a proposal in the database.
Trigger:	Actor decides to view a proposal.
Preconditions:	1. Actor is logged in to the system. 2. The proposal exists in the system. 3. Actor has located and selected a proposal to view. 4. For Alternative Flow 1.1: Proposal Record has at least one attached document.
Postconditions:	
Normal Flow:	**1.0 PI/Researcher Views Proposal** 1. System displays form with proposal data. 2. PI/Researcher chooses to navigate away from proposal form.
Alternative Flows:	**1.1 Actor Views Attached Documents** (branch after step 1) 1. Actor locates and selects the attached document that he/she wishes to view. 2. System displays dialogue box for actor to save the document. 3. Return to step 1.
Exceptions:	
Includes:	
Priority:	Imperative
Frequency of Use:	Daily
Business Rules:	
Special Requirements:	The proposal form is read-only. Actor must choose to edit the form for it to be written to, which is handled in the Update Proposal Record use case.
Assumptions:	
Notes and Issues:	Alternative Flow 1.1: actor would then navigate to and open the downloaded document once it is saved to his/her machine in order to view it.

3.11 View Researcher Profile

Actor: Office Personnel, Researcher?
Description: View data associated with researcher

- Safety training/certification of all participants
- up-to-date conflict of interest
- First and last name
- Employee ID number
- Department
- College/Institute
- Affiliation
- % of funding per PI/Co-PI of total requested
- Awarded amount

4 Specific Requirements

Specifics of the system are described via functional and non-functional requirements. Functional requirements, given in the first section, tell the functions that the system must be able to perform. Non-functional requirements, given in the second section, tell how a system must behave.

4.1 Functional Requirements

Functional requirements specify the functions that the system must be able to perform. These are organized according to major system functions.

4.1.1 Searching for Proposal Records

4.1.1.1 Search Criteria — Imperative

The system shall allow the user to search for proposal records based on one or more of the following criteria:

a. Keyword in title
b. Funding source
c. Researcher
d. Department
e. College
f. Status
g. Proposal due date
h. Quarter
i. Fiscal Year

The system shall allow the user to specify values for each desired criteria.
Verification type: Test, Observe, Inspect

Rationale: Clients will need to search for proposal records on one or more of these fields.
Comments: None.

4.1.1.2 Reset Search Criteria Fields **Important**

4.1.1.3 Search Option **Imperative**

The system shall allow the authorized users to initiate a search using the current state of the search criteria fields.
Verification type: Test, Observe, Inspect
Rationale: Clients must be able to initiate searches.
Comments: None.

4.1.1.4 No Search Criteria Error Message **Important**

4.1.1.5 Proposal Records Included in Search Results **Imperative**

The search results shall include proposals satisfying the search criteria, and no proposals that do not. Further, the system shall filter the search results based on the user class:

a. Research Office Personnel will be able to search for every proposal record.
b. Deans can search for proposal records relating to proposals in which faculty in their college are PIs or Co-PIs.
c. Department Chairs can search for proposal records relating to proposals in which faculty in their department are PIs or Co-PIs.

Verification type: Test, Observe, Inspect
Rationale: Some faculty members expressed concern over allowing Department Chairs and Deans from other departments/colleges see their proposals.
Comments: The Provost and Chancellor should also be allowed to view proposals.

4.1.1.6 No Search Results **Important**

If the search results is the empty set, the system shall display a message indicating that no proposal records satisfied the given search criteria.
Verification type: Test, Observe, Inspect
Rationale: Positive indication that a search yielded no results eliminates suspicion of a system error.
Comments: None.

4.1.1.7 Search Results Fields **Imperative**

The system shall display following items for every proposal record included in search results:

a. Title
b. PI
c. PI's Department
d. Funding Source
e. Proposal due date
f. Total Amount Requested
g. Status

Verification type: Test, Observe, Inspect
Rationale: Clients want to visually scan the above fields in search results without viewing the entire proposal.
Comments: None.

4.1.1.8 View Navigation **Imperative**

The search results shall enable the user to navigate to view the entire proposal record.
Verification type: Test, Observe, Inspect
Rationale: Users will need to view proposal records they find via searching.
Comments: None.

4.1.1.9 View/Update Navigation **Imperative**

For users allowed to update proposal records, the search results shall enable the user to navigate to update the entire proposal record.
Verification type: Test, Observe, Inspect
Rationale: Users may need to update proposal records they find via searching.
Comments: None.

4.1.1.10 Default Sorting **Important**

The system shall display search results sorted by proposal due date, most recent first.
Verification type: Test, Observe, Inspect
Rationale: Clients will most often wish to see most recent proposals first.
Comments: None

4.1.1.11 Export Search Results to Excel File **Important**

4.1.1.12 Excel Filename Already Exists **Important**

4.1.1.13 Illegal Pathname When Exporting to Excel **Important**

4.1.1.14 Results Sorting **Important**

4.1.1.15 Filter Current Search Results **Important**

4.1.2 Researcher Profile Requirements

4.1.2.1 View user information **Important**

Administrators shall be able to a view a list of all users of the system sorted alphabetically by last name.
Verification type: Test, Observe, Inspect
Rationale: Administrators may need to see a list of all system users.
Comments: This requirement may get changed because it seemed that Jeff did not want this list.

4.2 Non-Functional Requirements

Non-functional requirements tell how a system must behave.

4.2.1 Security

4.2.1.1 View user information

All network/internet communication between system components shall be secure.

4.2.1.2 Access to the System

The system shall only be available to users who have first logged in.

4.2.1.3 Creation of a Proposal Record

The system shall only allow a researcher to create a Proposal Record.

4.2.1.4 Update/Delete Proposal Record

The system shall only allow the PI, or Co-PIs from Tech, to update or delete Proposal Records. PI and Co-PIs can only update/delete the proposal record while the status of the Proposal Record is ‘Draft’ or ‘Submitted to Research Office’.

4.2.1.5 Search for Proposal Records

The system shall only allow authorized users to search for Proposal Records. (Include a reference here to the business rule telling who is authorized to search for proposal records.)

4.2.1.6 View Proposal Records

The system shall only allow authorized users to view Proposal Records. (Include a reference here to the business rule telling who is authorized to search for proposal records.)

4.2.1.7 Simultaneous Updates

The system shall ensure that any proposal record is being updated by at most one user at a time.

4.2.2 Availability

4.2.2.1 Simultaneous Updates

The system shall be 98% available Monday through Friday, 8:00 a.m. to 5:00 p.m. It shall be 95% percent available at other times, except for scheduled maintenance downtime Saturday and Sunday mornings, 6:00 a.m. to 9:00 a.m.

4.2.3 Usability

4.2.3.1 Usability Creating, Updating and Submitting Proposal Records

A researcher familiar with the paper version of the Proposal Certification Form, and with all of the pertinent information at hand, should be able to create, complete and submit a Proposal Record, within 20 minutes.

4.2.3.2 Usability for Research Office Personnel

Research Office personnel familiar with the business rules shall be able to search for Proposal Records, update Proposal Records, and update researcher profiles with no more than 15 minutes of orientation.

4.2.3.3 Search of Proposal Records

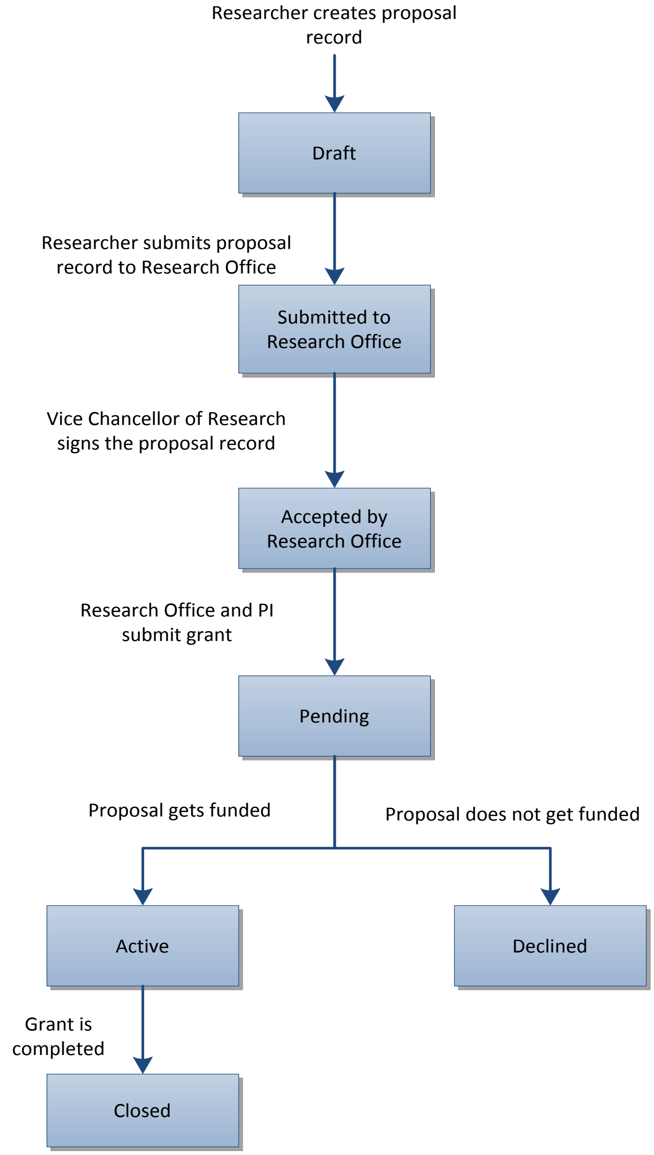
Administrators shall be able to use the system to search for Proposal Records with no orientation.

4.3 Requirements Models

This section shows visual models of different aspects of ePAS to aid in clarifying and validating the requirements.

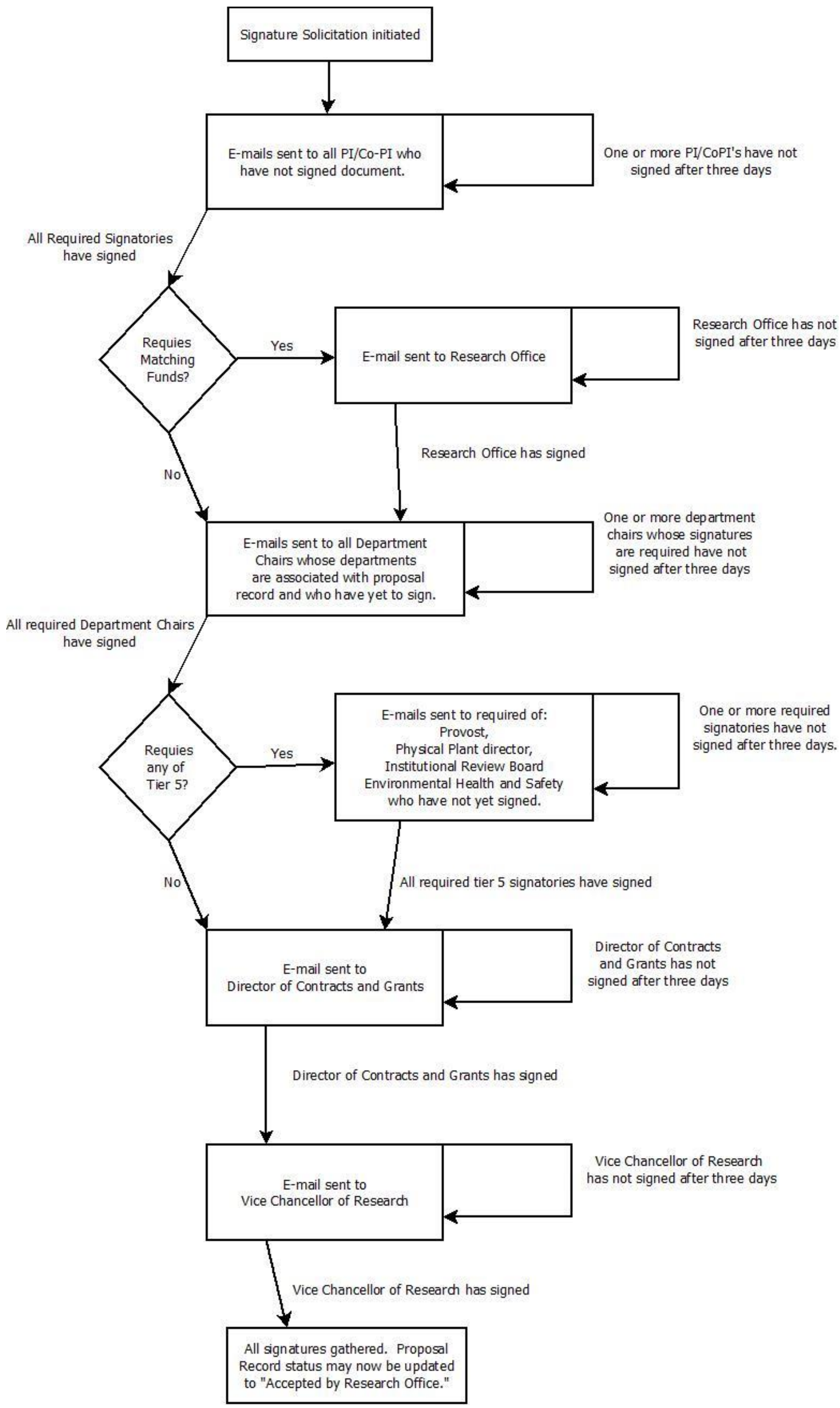
4.3.1 State-Transition Diagram for the status values of a Proposal Record

This diagram shows the different statuses that a proposal record can have throughout its lifecycle, as well as the conditions for a status to change to another status.



4.3.2 Signing of a Proposal Certification Form, Activity Diagram

This diagram shows the process of obtaining signatures on a Proposal Certification Form, along with explanation of when and how the signatures are obtained.



4.3.3 Dialog Map

This diagram shows the different screens and dialogs available to each user class and the navigation paths between them. The decision table at the top illustrates how the 'credentials' decision box decides what class a user belongs to upon login.