

# Trillium Health Grant Management Requirements Document

Version: Draft

Prepared by: Matthew Metcalf

10/6/2014

# Requirements Document Sign Off

Karen Elam \_\_\_\_\_

AJ Blythe \_\_\_\_\_

Russ \_\_\_\_\_

Matthew Metcalf \_\_\_\_\_

Brian To \_\_\_\_\_

Akshay Karnawat \_\_\_\_\_

Shannon Trudeau \_\_\_\_\_

Daniel Krutz \_\_\_\_\_

## Table of Contents

1.	Introduction .....	6
1.1.	Purpose .....	6
1.2.	Scope.....	6
2.	Requirements.....	6
2.1.	Description.....	6
2.2.	Requirements Matrix .....	7
3.	Use Cases.....	19
3.1.	<b>Documents</b> .....	19
3.1.1.	<b>Upload Document</b> .....	19
3.1.2.	<b>Lock Document</b> .....	20
3.1.3.	<b>Unlock Document</b> .....	20
3.1.4.	<b>Revert Document</b> .....	20
3.1.5.	<b>Examine Document History</b> .....	21
3.1.6.	<b>View Document</b> .....	21
3.1.7.	<b>Download Documents for Submission</b> .....	21
3.1.8.	<b>Delete Document</b> .....	22
3.1.9.	<b>Finalize Document</b> .....	22
3.2.	<b>Tasks and Workflows</b> .....	24
3.2.1.	<b>Monitor Workflow Progress</b> .....	24
3.2.2.	<b>Create a Task Reminder</b> .....	24
3.2.3.	<b>Edit a Task Reminder</b> .....	24
3.2.4.	<b>Update a Task</b> .....	24
3.2.5.	<b>Create a New Task</b> .....	25
3.2.6.	<b>Define a Task Template</b> .....	25
3.2.7.	<b>Sign-off a Completed Task</b> .....	26
3.2.8.	<b>Send Task Back for Review</b> .....	26
3.2.9.	<b>Schedule a Workflow</b> .....	26
3.2.10.	<b>Finish a Workflow</b> .....	26
3.3.	<b>System Administration</b> .....	28
3.3.1.	<b>Add User</b> .....	28

<b>3.3.2.</b>	<b>Deactivate User</b> .....	<b>29</b>
<b>3.3.3.</b>	<b>Impersonate User</b> .....	<b>29</b>
<b>3.3.4.</b>	<b>Reassign Grant Owner</b> .....	<b>29</b>

## Revision History

<b>Version</b>	<b>Description</b>	<b>Updated By</b>	<b>Date</b>
Draft	Initial	Matt Metcalf	9/3/2014
Draft .1	Adding Fluff	Matt Metcalf	9/17/14
Draft .2	Adding Requirements	Matt Metcalf	9/19/14
Draft .3	Add Use Case Descriptions	Brian To	9/22/14
Draft .4	Updated Formatting	Matt Metcalf	9/22/14
Draft .5	Updated Requirements based off Sponsor meeting on 9/23	Matt Metcalf	9/24/14
Draft .6	Added in the missing acceptance tests	Shannon Trudeau	10/6/2014
Draft .7	Added new Requirements	Matt Metcalf	10/6/14

# 1. Introduction

## 1.1. Purpose

This document is going to outline the Requirements for the Trillium Grant Management System. This will be done by outlining the scope of the project, defining the system environment, assumptions and dependencies, and finally the requirements matrix. The project scope will define the bounds of the project and say what will and will not be part of the project. The operating environment will describe the condition that the system will run in. Assumptions will go over any assumptions that will be taken moving forward with the project. Dependencies will go over any required systems that already exist that the system will depend on.

## 1.2. Scope

Trillium Grant Management System will standardize the process for Trillium Health. This is limited to notification of grant status, grant information, task creation, task reminders, and document upload, revision and approval. Grant status shall be communicated to the user by use of the dashboard by showing the status of the current tasks on a grant. Grant information shall be stored in the system and will be able to be view by viewing each individual grant. The system will allow for tasks to be created for grants. The system will notify users when a task is coming closer to its due date. The system will allow for upload of documents that are related to the grants. The system will keep track of previous revisions of documents so documents can be turned back to previous revisions. The system will allow for a process which requires grant owners to approve of documents before they are finalized.

# 2. Requirements

## 2.1. Description

The Requirements Matrix is broken up by Requirement type. The Requirement types are General, Gui, Security, Tasks, Usability, Grants, and Documents. Each requirement has 4 different parts; ID, Title, Description, and Acceptance Test. The ID column is a unique identifier for each requirement. Title column is the name of each of the requirement. The Description column details how the requirement is done. The Acceptance Test column describes how the completion of the requirement will be proved.

## 2.2. Requirements Matrix

ID	Title	Description	Acceptance Test
2.2.1.	General		
2.2.1.1.		The system shall ensure Timely and accurate submission of information	Make sure that it does not take forever to submit and store documents.
2.2.1.2.		The system shall standardize and keep a consistent the grant management process	Make sure that the process for each grant type is consistent.
2.2.1.3.		The system shall e-mail users regarding the status of grants.	The system emails the user when the grant is near its due date
2.2.1.4.		The system shall provide Task Management	The users are able to assign tasks and keep track all the tasks that are currently in progress, waiting, completed, or not yet started.
2.2.1.5.		The system shall provide Document Management	The users are able to upload documents, revise documents, revert to other documents, add new documents, etc.
2.2.1.6.		The system shall provide Event Preparation	The system is able to prepare a list of all the artifacts needed for an event, like CFA, audit, etc.
2.2.1.7.		The system shall provide a central place to manage grants that are being sought	The users are able to manage grants that are currently in research
2.2.1.8.		The system shall provide a central place to manage grants that are active	The users are able to manage grants that are currently in progress
2.2.1.9.		The system shall provide a storage location for additional grant documents	The users are able to upload documents for a grant
2.2.1.10.		The system shall provide status updates on major tasks, sub tasks and overall grant process.	The system sends a notification update when tasks and subtasks are assigned.

<b>ID</b>	<b>Title</b>	<b>Description</b>	<b>Acceptance Test</b>
2.2.1.11.		The system shall run on existing Trillium servers	Make sure that the system is running only on servers provided by Trillium Health
2.2.1.12.		The system shall run on Microsoft Windows Server 2008 or higher	duplicate
2.2.1.13.		The system should lock the document if someone is editing it	After a document is downloaded, it is locked by the system and then unlocked after the person uploads the edited version back.
2.2.1.14.		The system should have a logging system used for auditing.	Change a document, edit and then delete the document. Add a task, then change task, then delete the task. Repeat process for a grant.
2.2.1.15.		The system may have a central library of common documents.	Go to common documents library and copy a document into a grant.
<b>2.2.2.</b>	<b>Gui</b>		
2.2.2.1.		The GUI should have a Dashboard as the landing page.	As any user, the first page I should land on after login is the dashboard
2.2.2.2.		The Dashboard shall list only and all grants the user is allowed to access.	As any user, I am able see a list of all my grants that I am working on
2.2.2.3.		The Dashboard shall only list grants that are close to a due date for Managers.	As a Manager, I am only able to see grants within 1 month of a due date to completing their workflow on my Dashboard.
2.2.2.4.		The Dashboard shall list next Major/parent items required	As any user, I am able to see my next major/parent tasks
2.2.2.5.		The Dashboard shall list date frames for current tasks.	As any user, I am able to see the date of when the task is due and how far away that date is compared to today.
2.2.2.6.		The Dashboard should have a colored status icon next to the grant	As a user, I am able to see a yellow, green, or a red status icon next to the grant.



ID	Title	Description	Acceptance Test
2.2.2.7.		The Dashboard should have status next to each task followed by an icon.	As a user, I am able to see and read that task x was completed, or that task y is in progress, etc.
2.2.2.8.		The GUI should be able to display all the tasks that are assigned to a person	As a specific user, I am able to see only the tasks that are assigned to me.
2.2.2.9.		The GUI should display all the information for that grant when "clicked"	As any user, when a grant is "clicked", they are able to see information for that grant
2.2.2.10.		The GUI should have a navigation for all grants	When clicked on the "grants in progress", or "grants in research" there should be a navigation frame on the side for all the things associated with the grant.
2.2.2.11.		The Navigation frame should include cover page, contacts, documents, notes, and tasks	The Navigation page, should include a cover page, documents, notes, and tasks, etc.
2.2.2.12.		The GUI should display add and delete buttons for documents	The delete button should be present in case, the document has to be deleted.
2.2.2.13.		The GUI should display grant information for the grants currently in progress	Basic Information about the grant is displayed when clicked on that grant. Things such as Name, Source, Potential Funding, Primary Contact, Documents, current tasks, and any Additional Information.
2.2.2.14.		The GUI should have a Trillium Health logo on all pages	Make sure that the Trillium Health logo appears on each page
2.2.2.15.		The Gui should have a navigation bar	Make sure that the navigation bar exists, and is consisted of "Home", "Grants In Progress", "Grants in Research", "Admin", and "Logout"
2.2.2.16.		The Gui should have a page with a list of all the contacts that are associated with the grant	There is a list of all the contacts that are part of the grant. The page should include their name, addresses, phone numbers, type, and if a person is a primary contact person.
2.2.2.17.		The Gui should have a documents page with a list of all the documents associated with that grant	As a user, I should be able to see a list of all the documents associated with the grant. Some documents may be hidden from one individual, but are visible to another.

ID	Title	Description	Acceptance Test
2.2.2.18.		The Gui should include a page where the users are allowed to add notes	As a user, I should be able to add and delete a comment/note
2.2.2.19.		The Gui should have a place for external links that are related to that grant	As a user, I should be able to access any external link for a specific grant.
2.2.2.20.		The Navigation frame should have a submissions page for each grant	As a user, I should be able to see all the submissions for each stages of the grant process
<b>2.2.3.</b>	Security		
2.2.3.1.		The system shall only allow approved people to access grants	As a Technical Admin, Manager, or Owner, I can give users permissions to access grants. Only those people can view the existence of grants and their associated tasks and documents.
2.2.3.2.		The system shall allow for documents to have added security, based on roles.	As a Technical Admin, Manager, or Owner, I can give users permissions to edit grants. Only those people can edit grants and their associated tasks and documents.
2.2.3.3.		The system shall allow for some information to be public to all at Trillium for reference.	As anyone who has access to trillium's internal network, I would like to be able to go to the server address of the grant management system and see some kind of dashboard/homepage
2.2.3.4.		The system shall allow for the following user types; Grant Manager, Grant Owner, Financial Analyst, Auditor, Technical Administrator, Grant Worker (Base User), Read-Only	An account can be assigned any of the listed roles and given that respective set of permissions.
2.2.3.5.		The system shall allow for anyone to assign tasks	As a grant manager, I want to ability to assign tasks to anyone in the company with an account in the system? to do a task

ID	Title	Description	Acceptance Test
2.2.3.6.		The system shall allow Financial Analysts to view all Financial data.	As a Financial analyst user, I would like to be able to download any financial documents I want. As any other user, I shall not be able to view financial documents
2.2.3.7.		The system shall allow Auditors to view everything under audit in the system.	As an Auditor, I would like to be able to view all documents associated with a grant
2.2.3.8.		The system shall allow Technical Administrators to perform system upkeep.	Technical Admins should be able to assigning roles and permissions within the system, and accessing all documents and grants
2.2.3.9.		The system should allow for Technical Administrators to become a "proxy user" of any user.	As a Technical Administrator, I would like to view exactly what another user sees when they log on, so that I am able to help them troubleshoot
2.2.3.10.		The system shall allow Technical Administrators to assign levels of privilege to users' accounts.	As a Technical Administrator, I would like to change the permissions of any users' account
2.2.3.11.	(Deleted)	<del>The system shall not allow for Technical Administrators to proxy into Financial Analyst roles and view their sensitive information</del>	<del>As a Technical Administrator in proxy, I shall not be able to open any documents that I wouldn't normally be able to see</del>
2.2.3.12.		The system shall allow for Grant Managers, Grant Owners, or Technical Administrators to add and remove users from Grants they are associated with.	As a Grant Manger, Owner, or Technical Administrator, I can add and remove user privileges related to certain grants.
2.2.3.13.		The system shall allow only Grant Managers, Grant Owners, and Technical Administrators to access the Recycling bin.	As a Grant Manager, Grant owner, or Technical Administrator, I can view the contents of the recycling bin, empty the recycling bin, or restore particular documents.
2.2.3.14.		The system may have the ability allow external users to view documents when they pass a check.	An external user can access a document by a given link and provide an authentication password.
<b>2.2.4.</b>	Tasks		
2.2.4.1.		Tasks shall be able to contain child tasks	As a grant owner, create a task and a child task for the first task

ID	Title	Description	Acceptance Test
2.2.4.2.		Tasks shall allow at least 9 levels of sub-tasks past the parent task	As a grant owner, create 10 levels deep of child tasks
2.2.4.3.		Tasks shall have due dates	As a grant worker, I can assign the due date of a task
2.2.4.4.		Tasks shall remind grant owners and grant participants at manager or owner-defined times	Given a task with a 1 day reminder until due, As an assignee, I should get an e-mail notification that the task is due in one day
2.2.4.5.		Tasks may be signed off as completed by grant owners, managers, or admins	As a grant owner, I can make a task require sign-off before it is considered completed. As a grant owner, I can sign-off a task when someone completes it.
2.2.4.6.		Tasks may be assigned to a grant participant by anyone	As a grant owner, I can assign another member to a task
2.2.4.7.		Tasks may be created from a template (specific to each grant)	As a grant owner, I can create a new task from a task template
2.2.4.7.1.		Templates may be public or private.	As a grant owner, I can assign a template the status of “private” meaning only I can see it or “public” meaning anyone that has access to the grant also has access to the associated templates.
2.2.4.7.2.		Template may not be overwritten.	Once a template has been created, any alterations to that template will be saved as either a new template or a general task item.
2.2.4.8.		Tasks may be one-time or recurring if made into a template.	As a grant owner, I can create a task template. As a grant owner, I can create a new empty task, or I can create a task from a template.
2.2.4.9.		Tasks shall be shown on the dashboard only if you are a owner or participant, and the task is assigned to you	As a grant owner, because I have tasks that I need to do, I can view the tasks that are assigned to me

<b>ID</b>	<b>Title</b>	<b>Description</b>	<b>Acceptance Test</b>
2.2.4.10.		Tasks shall be shown on the dashboard only if you are a manager, and the task is past the first reminder	As a grant owner or manager, because I am worried the workflow is behind, I can view tasks whose due dates are close
2.2.4.11.		Tasks shall belong to a workflow	As a grant owner, I can schedule a new workflow. When the workflow starts, associated tasks should be created automatically.
2.2.4.12.		Tasks should add a person to a grant as a base worker when they are assigned a task on a grant they aren't a part of yet.	As a grant worker, if I am assigned a task to a grant that I am not a part of yet, I am automatically given base permissions so that I can read the task that I have been assigned
2.2.4.13.		Tasks may have priority over others, separate from their due dates.	As a grant worker, I can create a task with a high, medium, or low priority which will affect the status of the task in addition to the number of days until the task is due.
2.2.4.14.		Tasks may have multiple grant workers, assigned to them.	As a grant worker, I can assign two grant workers to a task.
<b>2.2.5.</b>	Usability		
2.2.5.1.		The System Should integrate Task Due dates with Microsoft Outlook Calendar	Assign a user to a task and verify that the user now has the due date in their Outlook Calendar.
2.2.5.2.		The System Shall use either links or buttons to indicate an actionable item.	Verify that clicking on anything that doesn't look like a link or button does not do anything
2.2.5.3.		The System Shall have hover text over all icons	Hover over every icon and verify that the icons name shows as hover text.
2.2.5.4.		The System Shall show relevant information on all grants the user is allowed to see	When a screen change the system shows only the relevant information.
2.2.5.5.		The System Shall only use approved icon images.	All icons will be approved by AJ or Russ for understandability.

ID	Title	Description	Acceptance Test
2.2.5.6.		The System Should use links when possible to help with digging down into information.	Click on every Grant title and Task title to see if the system take the user to the Grant or Task.
2.2.5.7.		The System May be used via phone or tablet.	Verify that all Read-Only actions are usable on both a phone and tablet.
2.2.5.8.		The System Should Not use colors that would hinder a color blind user.	Have a color blind user move through the site and verify that no viewable area is hard to see.
2.2.5.9.		The System Should have at least 2 ways to reach every page sub-part	Verify that there are 2 ways to move to every page sub-part.
2.2.5.10.		The System Should have at least 1 way to reach the home page from every page.	Clicking on Trillium logo will bring you to the home page/dashboard.
<b>2.2.6.</b>	Grants		
2.2.6.1.		Grants shall have ability to be associated with a workflow	As a grant owner, I can schedule a workflow. When the workflow start date approaches, Then the workflow should automatically create associated workflow tasks.
2.2.6.2.		Grants shall have the ability to have workflows created for them.	As a grant owner, I can create a workflow for a grant, specifying a start date for the workflow, creating tasks to automatically be added to the workflow.
2.2.6.3.	(deleted)	<del>Grants may have ability to be associated with a audit workflow</del>	<del>As a grant owner, I can schedule an audit. When the audit workflow start date approaches, Then the workflow should automatically create associated workflow tasks.</del>
2.2.6.4.	(deleted)	<del>Grants may have ability to be associated with a re-application workflow</del>	<del>As a grant owner, I can schedule a re-application. When the re-application workflow start date approaches, Then the workflow should automatically create associated workflow tasks.</del>

<b>ID</b>	<b>Title</b>	<b>Description</b>	<b>Acceptance Test</b>
2.2.6.5.		Grants shall have the ability to copy documents such as payroll, budgets, org charts, mission statements, and outcomes from other workflows in the grant.	As a grant owner, I can associate (add) a needed document to the grant. As a grant worker, I can view associated grant documents, assuming I have permission.
2.2.6.6.		Grants shall have links to needed documents all located on the same page (1 click away)	As a grant owner, I can click on a link to view a document, and then click on another link to view another document without leaving "the page"
2.2.6.7.		Grants shall keep track of funder, contacts, contract numbers, and award amount, only	As a grant manager, I can find/view the funder/contacts/contract numbers/award for a specific grant
2.2.6.8.		Grants shall have at least one grant owner at all times	As a grant manager, I can assign someone (including myself) as the grant owner of a grant
2.2.6.9.		Grants shall have the ability to change owners	As a grant owner or manager, I can assign someone else as the grant owner
2.2.6.10.		Grants must keep a history of previous submissions	As a grant owner or manager, I can view the previous submissions of a grant workflow (CFA, audit, ...)
2.2.6.11.		Grants must keep track of past, current, and recurring tasks	As a grant owner, I can view previous tasks of a grant's workflow. As a grant owner, I can view all current tasks for a grant's workflow. As a grant owner, I can add a new task template (recurring task)
<b>2.2.7.</b>	Documents		
2.2.7.1.		Documents should track changes	As a grant worker, when I download a document I would like to see the line by line changes made to said document.
2.2.7.2.		Documents shall have a revision history	As a grant worker, when I am looking at a document in the system, I would like to be able to see all the past revision dates and names, as well as who created that revision.

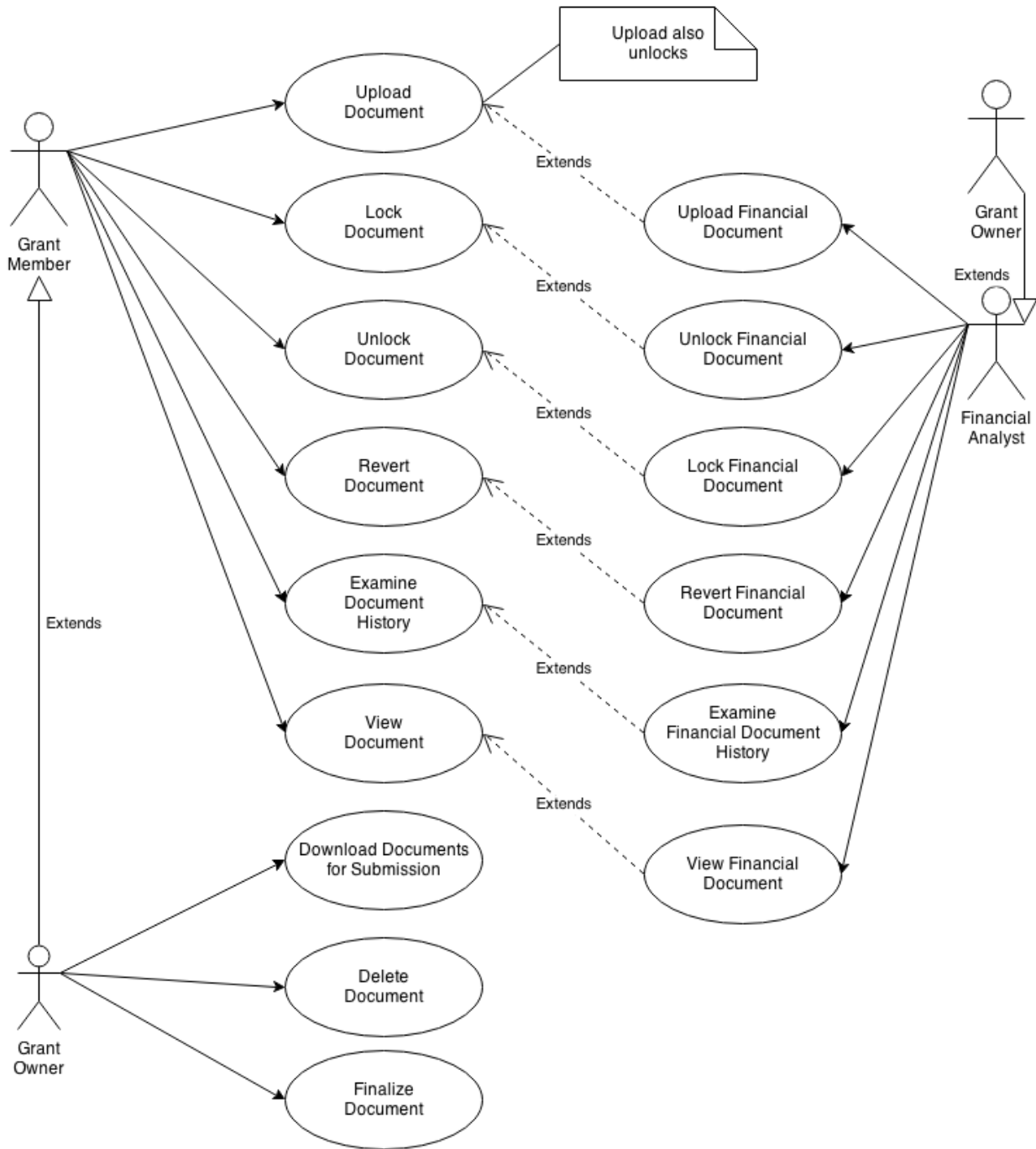
ID	Title	Description	Acceptance Test
2.2.7.3.		Documents shall be able to be downloaded	As a user who has permissions to use this document, I can click a button and download it
2.2.7.4.		Documents should be able to be downloaded "in bulk"	As a user, I would like to be able to pick and choose from a list of documents associated with a grant and download them all in a zip folder, either for my own personal use or for giving a set of documents to an auditor
2.2.7.5.		Documents shall be able to be uploaded to a document library	As a user, I need to be able to click on a grant and have the option to upload a document under no particular task or subfolder
2.2.7.6.		Documents should be able to be uploaded within a specific task	As a grant worker, if I am viewing a task which requires a document upload as part of the completion process, I can go directly to the task and submit the document there.
2.2.7.7.		Document revisions shall have a state of either; Major or Minor	All Major revisions will be saved in the revision history, all minor revisions will only be saved until the next major revision, at which time the minor revisions will be permanently deleted.
2.2.7.8.		Documents shall record times and users who change or review them.	As a grant worker, I can look at the revision history of a document and see for each major and minor revision the time the document was uploaded and who uploaded it.
2.2.7.9.		Documents shall be able to revert to a previous version of itself.	Documents can revert the most recent copy to any of the previous saved versions, major or minor. The new copy of the document would be shown in the revision history as the next minor revision.



ID	Title	Description	Acceptance Test
2.2.7.10.		Documents shall be able to copy themselves from previous submissions.	As a user, I would like to make a copy of a document from a previous submission/revision to be used in a new submission. Revision. Afterwards, this document should be marked in some way as "unchecked" so that the user knows to revisit the document, as it is out of date.
2.2.7.11.		Documents shall have the option of being reviewed (or not)	As a grant manager, I need to be able to mark documents as "needs review" or not, so that all documents are up to date and accounted for
2.2.7.12.		Documents should be marked with the date of when the review occurred (or when the doc was marked off as reviewed).	As a grant manager, I want to know when this document was reviewed to determine if it was recent enough or if it should be reviewed again.
2.2.7.13.	(deleted)	<del>Documents shall be updatable for new application or CFA.</del>	
2.2.7.14.		Documents should be able to be viewed within the system.	As a grant worker, I would like to view a document that I have permission to view in the browser (without having to download and open the file in an external program)
2.2.7.15.		Documents shall be able to be "checked out" of the system	As a user, I need to be able to "check out" a document when editing it, so that no one else edits that document at the same time as me.
2.2.7.16.		Documents shall be locked when a user checks out a document	As a user, I should not be able to download a document if the document has been checked out by someone else
2.2.7.17.		Documents shall be able to be checked into the system by a user who check it out or by a Technical Administrator.	As the user who checked out this document, I should be the only one allowed to check it back in, or to upload a revised version of this document.

ID	Title	Description	Acceptance Test
2.2.7.18.		Documents shall require the Grant Manager to approve its deletion.	As a grant manager, I want to be the only one who can delete documents from a grant.
2.2.7.19.		Documents shall only be edited by financial analysts if they contain financial information	As a Financial analyst user, I would like to be able to download any financial documents I want. As any other user, I shall not be able to view financial documents
2.2.7.20.		Documents shall be given the status of Major prior to submission	As a grant worker, I need to know when I document is completed and ready to be sent. When a document is completed, it will be marked as a Major revision and saved in the revision history.
2.2.7.21.		Documents may have the ability to have their older versions viewed in the system.	As a grant worker, I would like to see the contents of the previous revisions of the documents.
2.2.7.22.		Documents may have the ability to be moved to a “Recycling Bin” when deleted by a user.	As a grant worker, when I delete a document, I would like the document removed from my viewing abilities and sent to a recycling bin to be reviewed by either a Grant Owner, Manager, or Technical Administrator
2.2.7.23.		Documents may have the ability to be permanently deleted from the Recycling Bin.	As a Grant Owner, Manager, or Technical Administrator, I am able to permanently delete items from the recycling bin.
2.2.7.24.		Documents have the ability to delete a minor version.	As a grant worker, I can delete a minor version of a document.
2.2.7.25.		Documents have the ability to retain a minor version.	As a grant worker, I can mark a minor document version so that it can't be deleted.
2.2.7.26.		Documents have the ability to be shared by a link.	As a grant manager, I can send a link to an external user so they can view and edit the document.

### 3. Use Cases



#### 3.1. Documents

*NOTE: Financial Analysts have the same use cases as below, but documents are strictly financial. Grant Owners can do everything Financial Analysts can only for the grant they own.*

##### 3.1.1. Upload Document

- **Primary Actor:** Grant worker, Grant Owner

- **Purpose and Goal:**
  - Because CFAs, audits, and other workflows/tasks may many required documents, it is necessary for people to be able to upload a relevant document to fulfill the requirement of a specific workflow.
  - A grant worker/owner uploads a document for a specific workflow of a grant.
- **Preconditions:**
  - The document is not financial
  - The uploader is a member/owner of the grant
  - The document is not locked or is locked by the uploader
- **Success Scenario:** Document is uploaded
- **Postconditions (aside from success scenario):**
  - The document is unlocked

### 3.1.2. Lock Document

- **Primary Actor:** Grant worker, Grant Owner
- **Purpose and Goal:**
  - Ideally, only one person works on a document at a time since it is easy to “step over each other’s foot” when people are making changes to the same document at the same time.
  - Protect against accidentally writing over someone else’s changes.
- **Preconditions:**
  - The document exists
  - The document is not financial
  - The uploader is a member/owner of the grant
  - The document is not already locked
- **Success Scenario:** Document is locked
- **Postconditions (aside from success scenario):**
  - None

### 3.1.3. Unlock Document

- **Primary Actor:** Grant worker, Grant Owner
- **Purpose and Goal:**
  - Protect against accidentally writing over someone else’s changes.
- **Preconditions:**
  - The document exists
  - The document is not financial
  - The uploader is a member/owner of the grant
  - The document is locked by the user who is unlocking this document
- **Success Scenario:** Document is unlocked
- **Postconditions (aside from success scenario):**
  -

### 3.1.4. Revert Document

- **Primary Actor:** Grant worker, Grant Owner
- **Purpose and Goal:**

- If someone made a mistake and uploads a faulty document, it should be easy to rectify that mistake.
- Make a previous document version the current document version.
- **Preconditions:**
  - The document exists
  - The document is not financial
  - The uploader is a member/owner of the grant
  - The document is not locked or is locked by the reverter
  - The workflow has not been completed
  - The reverted version is not the latest version (makes no sense otherwise)
- **Success Scenario:** Document is now a previous version
- **Postconditions (aside from success scenario):**
  - The reverted version is still in the document history

### 3.1.5. Examine Document History

- **Primary Actor:** Grant worker, Grant Owner
- **Purpose and Goal:**
  - Since documents are usually not written by one person exclusively, knowing who made changes helps grant workers figure out what is left to do.
  - View change history of a document
- **Preconditions:**
  - The document is not financial
  - The user is a member/owner of the grant.
- **Success Scenario:** Complete history of document is shown
- **Postconditions (aside from success scenario):**
  - None

### 3.1.6. View Document

- **Primary Actor:** Grant worker, Grant Owner
- **Purpose and Goal:**
  - Having a document stored somewhere is useless unless we can get it back.
  - View a document (for review or for curiosity)
- **Preconditions:**
  - The document is not financial
  - The user is a member/owner of the grant.
- **Success Scenario:** TODO
- **Postconditions (aside from success scenario):**
  - None

### 3.1.7. Download Documents for Submission

- **Primary Actor:** Grant Owner
- **Purpose and Goal:**
  - Each workflow has submission requirements (which is why we're doing this project) and these artifacts should be easy to download.
  - Download relevant documents for a workflow (submission)
- **Preconditions:**

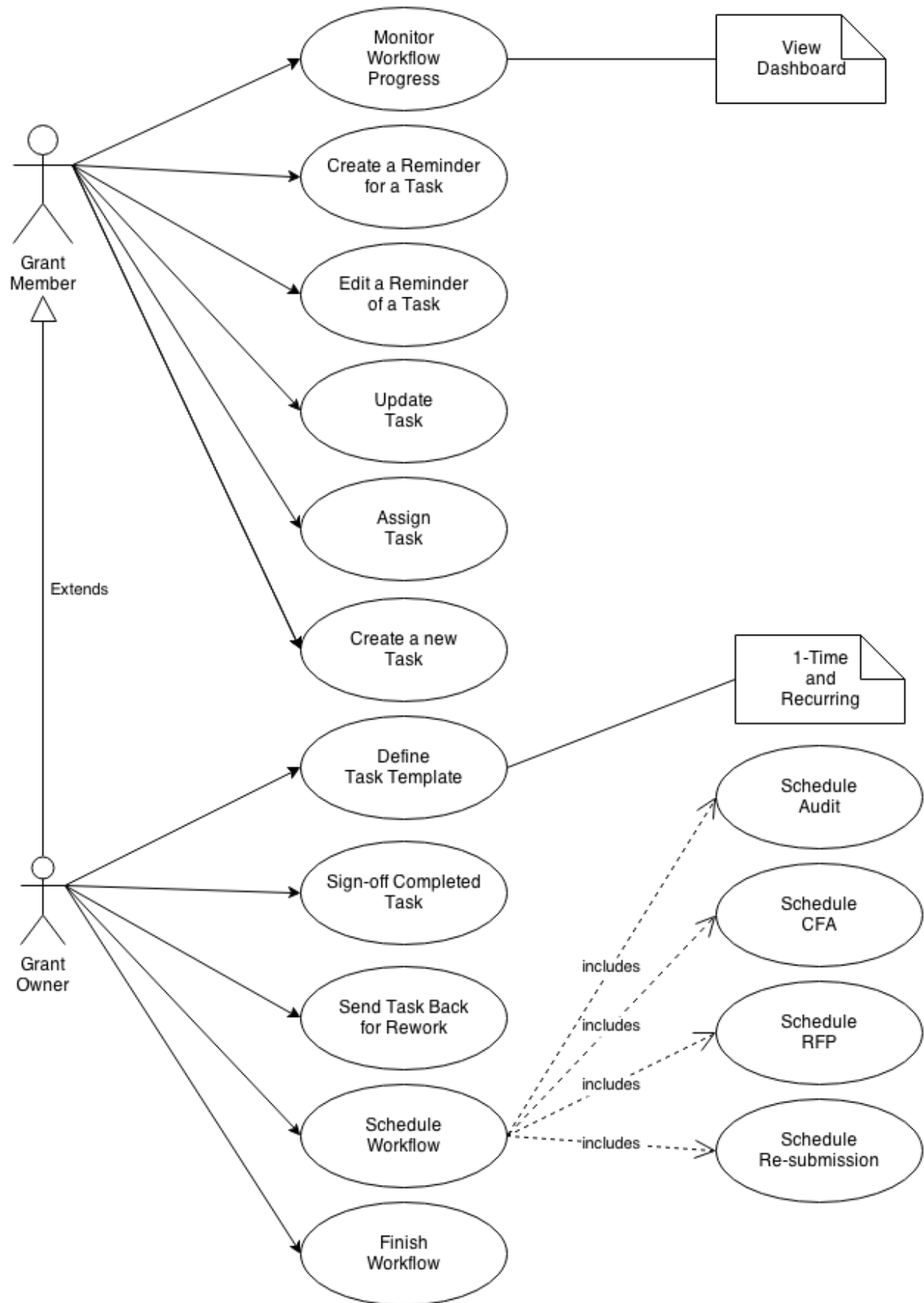
- None
- **Success Scenario:** Document is saved to user's hard drive
- **Postconditions (aside from success scenario):**
  - None

### 3.1.8. Delete Document

- **Primary Actor:** Grant Owner
- **Purpose and Goal:**
  - Perhaps an incorrect document was uploaded and there is not enough history to warrant a reverted change, or the document is incorrectly named
  - Delete a faulty, historyless document
- **Preconditions:**
  - The document is not locked or is locked by the reverter
  - The workflow has not been completed
- **Success Scenario:** Document is deleted from the system (irreversible)
- **Postconditions (aside from success scenario):**
  - All relevant history is also deleted

### 3.1.9. Finalize Document

- **Primary Actor:** Grant Owner
- **Purpose and Goal:**
  - Once a document is “completed”, no more changes are needed and it should be left alone for the current workflow/submission.
  - Prevent potentially erroneous document updates from happening and to indicate to people who are working on this grant that this document is “done”
- **Preconditions:**
  - Document has at least one version
- **Success Scenario:** Document is finalized
- **Postconditions (aside from success scenario):**
  - None



## 3.2. Tasks and Workflows

### 3.2.1. Monitor Workflow Progress

- **Primary Actor:** Grant worker
- **Purpose and Goal:**
  - I want to monitor the status of a workflow to make sure we're not missing our deadlines. For example, I want to know what the immediate upcoming tasks are, if tasks are overdue, and who the responsible parties are.
- **Preconditions:**
  - None
- **Success Scenario:** Member sees relevant status of workflow
- **Postconditions (aside from success scenario):**
  - None

### 3.2.2. Create a Task Reminder

- **Primary Actor:** Grant worker
- **Purpose and Goal:**
  - Since tasks are deadline-sensitive, I want to create reminders for myself or whoever is working on the task so they don't let these tasks slip through the cracks.
  - Set up a reminder for a single task
- **Preconditions:**
  - The reminder is in the future relative to today
  - There is someone assigned to the task
  - The task is not already complete
- **Success Scenario:** A task reminder is created
- **Postconditions (aside from success scenario):**
  - None

### 3.2.3. Edit a Task Reminder

- **Primary Actor:** Grant worker
- **Purpose and Goal:**
  - If deadlines need to be shifted around, we should be able to accommodate.
- **Preconditions:**
  - The task is not already complete
- **Success Scenario:** The task reminder date is updated
- **Postconditions (aside from success scenario):**
  - None

### 3.2.4. Update a Task

- **Primary Actor:** Grant worker
- **Purpose and Goal:**
  - Once a task is logically in progress, completed, pending review, or such, the task should be updated to reflect reality in order to communicate the accurate status to interested parties that are [monitoring workflow progress](#).



- Update a task
  - Update status (in progress, reviewing, completed, ...)
  - Add/Remove drop boxes
  - Change task title
  - Change assignee
  - Toggle sign-off requirement
- **Preconditions:**
  - Updater is assigned to the task
  - Task is in a state that is updateable, see [possible task states](#).
- **Success Scenario:** Task is updated
- **Postconditions (aside from success scenario):**
  - Status, assignee, and title cannot be null

### 3.2.5. Create a New Task

- **Primary Actor:** Grant worker
- **Purpose and Goal:**
  - Create a task for him/herself, or for someone else.
- **Preconditions:**
  - None
- **Success Scenarios:**
  - Task is created from a blank template
  - Task is created from a defined template
  - Task is created as a root task
  - Task is created as a child task
- **Postconditions (aside from success scenario):**
  - Task has a due date that is in the future
  - Task has an assignee

### 3.2.6. Define a Task Template

- **Primary Actor:** Grant Owner
- **Purpose and Goal:**
  - For recurring tasks, it makes sense to “template out” tasks so that creating families of tasks is much less time-consuming.
  - Create a “task template” where (below) is copied over or smartly guessed:
    - Name or Title
    - Dropbox(es)
    - Require/Does not require sign-off
    - Due date (as duration after creation)
    - Assignee (as user who creates the task, will be re-assigned manually)
- **Preconditions:**
  - None
- **Success Scenario:** Task template is created
- **Postconditions (aside from success scenario):**
  - Task created (and associated children tasks) have the creator as the assignee

### 3.2.7. Sign-off a Completed Task

- **Primary Actor:** Grant Owner
- **Purpose and Goal:**
  - Some tasks require verification that it is done correctly (review) since ramifications of a low-quality output are high for this task.
  - Sign-off a task under review as completed
- **Preconditions:**
  - Task requires sign-off
  - Assignee is not the grant owner (why would you need to sign-off yourself?)
  - Task is in a review state
- **Success Scenario:** Task state is set to completed
- **Postconditions (aside from success scenario):**
  - Assignee is notified of sign-off

### 3.2.8. Send Task Back for Review

- **Primary Actor:** Grant Owner
- **Purpose and Goal:**
  - Some tasks require verification that it is done correctly (review) since ramifications of a low-quality output are high for this task.
  - Sign-off a task under review as completed
- **Preconditions:**
  - Task requires sign-off
  - Assignee is not the grant owner (why would you need to sign-off yourself?)
  - Task is in a review state
- **Success Scenario:** Task state is set to rework
- **Postconditions (aside from success scenario):**
  - Assignee is notified for rework

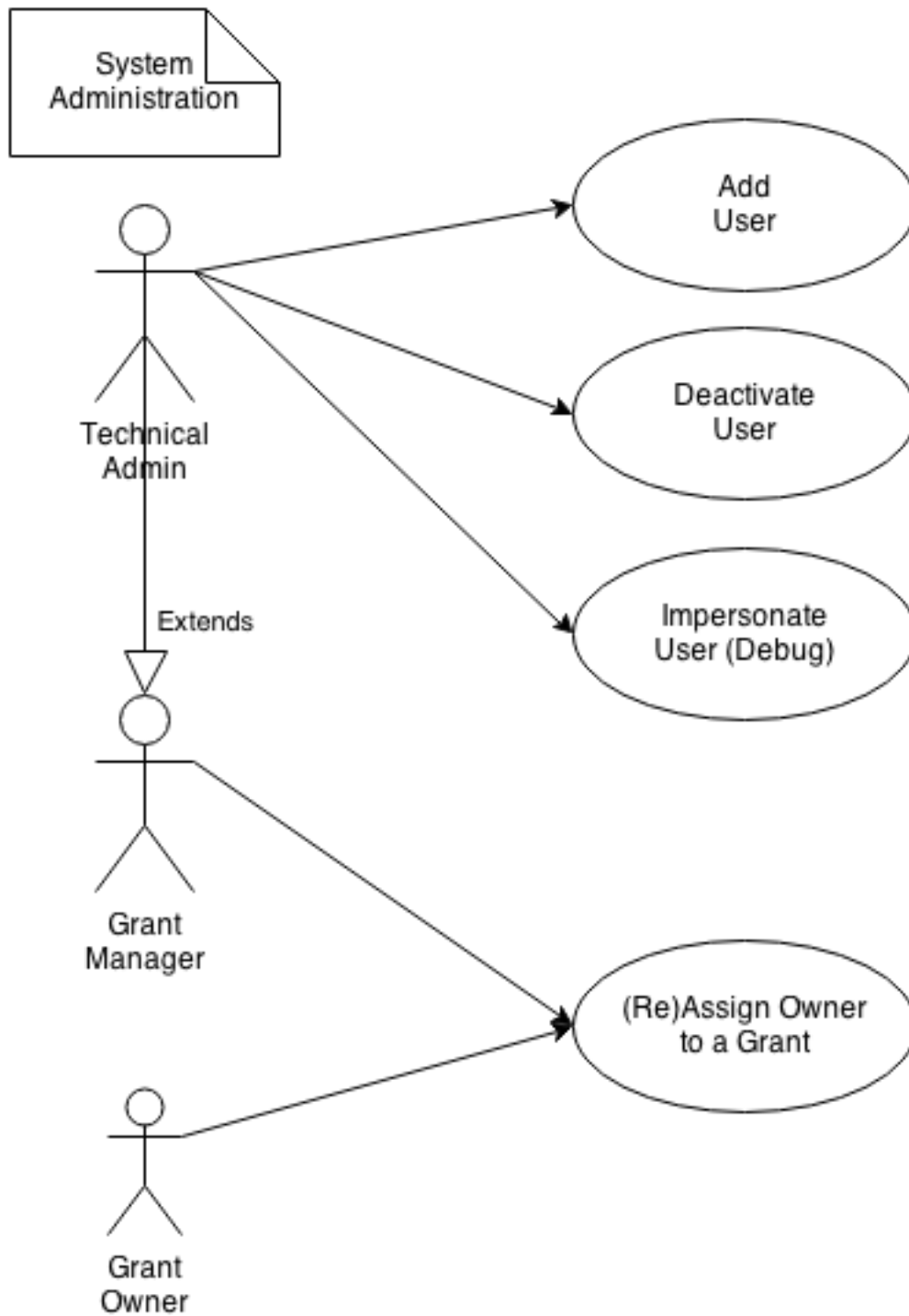
### 3.2.9. Schedule a Workflow

- **Primary Actor:** Grant Owner
- **Purpose and Goal:**
  - Events such as CFA, CAP, RFP, audit, and reapplication represent the “major” tasks that represent the source of work for members, and the source of tasks.
  - Schedule a CFA, RFP, audit, or reapplication.
- **Preconditions:**
  - The start date is in the future
  - The to-be-scheduled workflow isn’t already a current workflow
  - The workflow is not CAP (CAP is started automatically after CFA finishes)
- **Success Scenario:** Workflow is scheduled
- **Postconditions (aside from success scenario):**
  - None

### 3.2.10. Finish a Workflow

- **Primary Actor:** Grant Owner
- **Purpose and Goal:**

- Once all tasks are done (document changes, submission, ...), a workflow is done and no more work is needed for a grant (because there is no event that drives more tasks).
  - Finish a workflow
- **Preconditions:**
  - All tasks are completed
  - The workflow is not CFA. Finishing CFA causes a CAP to start.
- **Success Scenario:** Workflow is scheduled
- **Postconditions (aside from success scenario):**
  - None



### 3.3. System Administration

#### 3.3.1. Add User

- **Primary Actor:** Technical Administrator

- **Purpose and Goal:**
  - Employees come and go.
  - Add a user (grant worker, owner, manager, tech. admin)
- **Preconditions:**
  - None
- **Success Scenario:** User is added
- **Postconditions (aside from success scenario):**
  - None

### 3.3.2. Deactivate User

- **Primary Actor:** Technical Administrator
- **Purpose and Goal:**
  - Employees come and go.
  - Deactivate a user
- **Preconditions:**
  - The user is not yourself
- **Success Scenario:** User is deactivated
- **Postconditions (aside from success scenario):**
  - All of the deactivated users tasks are re-assigned to the tech admin who deactivated the account
  - All grants that the deactivated user owns is now transferred to the tech admin who deactivated the account

### 3.3.3. Impersonate User

- **Primary Actor:** Technical Administrator
- **Purpose and Goal:**
  - Users (especially newer users) may have trouble using the grant management system that may be caused by a myriad of things (within the system, or outside).
  - Impersonate a user in order to diagnose and correct problems they face or find.
- **Preconditions:**
  - The user is not another technical admin or yourself
- **Success Scenario:** The technical admin can view pages as if he/she is another user
- **Postconditions (aside from success scenario):**
  - None

### 3.3.4. Reassign Grant Owner

- **Primary Actor:** Grant Owner, Grant Manager, Technical Administrator
- **Purpose and Goal:**
  - Employees come and go. The technical administrator probably does not want to be the owner of the grant *forever*.
  - Reassign the owner of a grant to someone else
- **Preconditions:**
  - The new owner is not yourself
- **Success Scenario:** Grant has a new owner
- **Postconditions (aside from success scenario):**
  - None