Elizabeth Lor
CSCI 5448
Nov. 9, 2016
Extra Credit – Project 2

Modify Project Information

Manager is able to modify the information of a project that they have created.

Use Case Diagram

To modify project information, the user must first be able to view their list of projects. This will ensure that they can modify the project that they created. With modify project information almost, but not all fields are required to create a project.
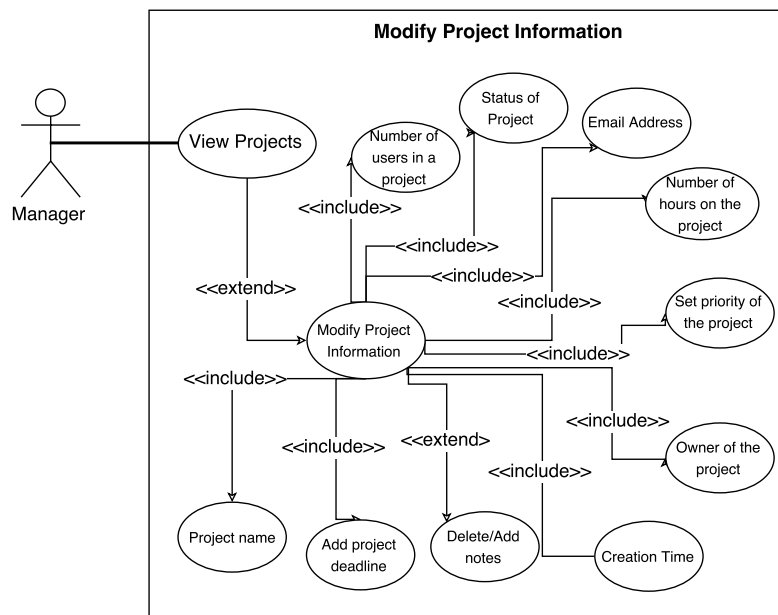
Use Case Documents

The view project use case is needed in order to access the modify project information. The user can select from the projects that they created and modify the project information.

Use Case Name:	View projects
Description:	Allows any user to view what projects they are the owners of or are assigned to

| Actors:	Managers, Employees	
Pre-conditions:	User is logged in	
Post-conditions:	User will be presented with all projects in which the user is a participant	
Flow of Events:	**Main**	**Response**
	1. User selects "View Projects"	**1. System queries the "owner" and "participants" tables of all projects in the database and returns all projects for which the user is either project owner or one of the participants.** **System displays a list of the returned projects**
Exceptional Events flow:	1. User exits the application	1. No action
Variations:		
Notes and Issues:		

Use Case Name:	Modify project	
Description:	Allows managers to make changes to the attributes of their own projects	
Actors:	Manager	
Pre-conditions:	User is logged in; user must be manager/owner of the selected project	
Post-conditions:	Selected existing project in database will have its attributes reassigned with the newly user defined values.	
Flow of Events:	**Main**	**Response**
	1. From the view project page, user selects to make changes to a	**1. System verifies if the user is the project owner. System prompts the user with a web form to define the project attributes (the same form to as when creating a new project) the**

	project	system queries the existing project's attributes and pre-fills the fields with the current values.
	2. User fills in web form, submits	**2. System reassigns project attributes with the newly defined values.**
Exceptional Events flow:	1. User is not a manager	1. System notifies the user, exits the process, and redirects the user to the home page.
	2. User exits the application	2. No action
Variations:		
Notes and Issues:		

Activity Diagram

The project manager request to view their projects. The system takes in the request and shows a list of projects pertaining to the project manager. If the project manager chooses not to modify the project, then the use case is done. If the project manager chooses to modify the project information, the project manager will fill in the necessary information. When the project manager is done, the use case is done, if not, the system will take the project manager back to the modify project information screen.
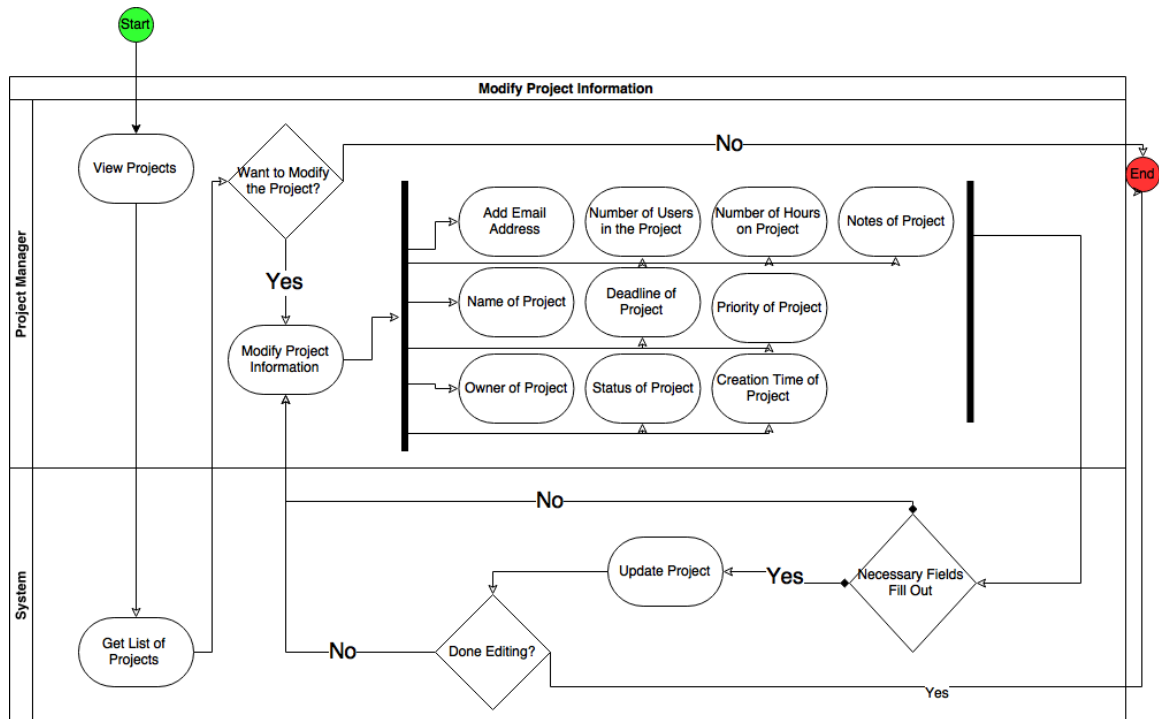
Sequence Diagram

The user selects on a project that they want to modify. The request is sent to the ProjectController, which then creates a ProjectFormView. This ProjectFormView is where the user will modify the information of a project. When the user is done modifying the project, the ProjectController write it to the ProjectModel and also sends a message to the ProjectListView to let the user know that the information has been saved.
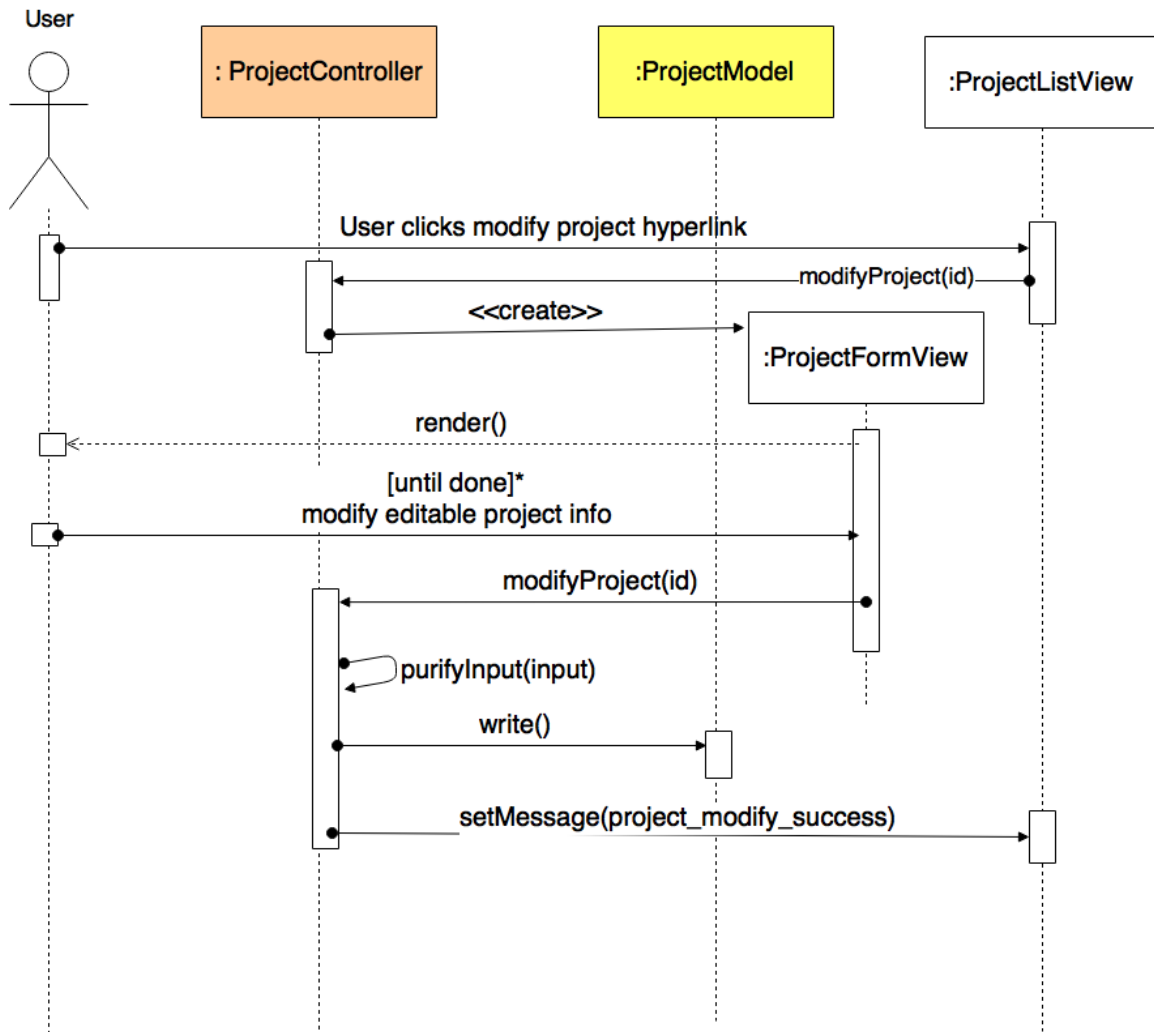
Class Diagram

This is a class diagrams that shows all of the classes that pertains to this use case. There are also a couple of other classes shown to give a visual of what is connected to the other main classes. For example, the Project class shows the attributes that are in a project. These attributes are what the user can modify when trying to modify project information.

ProjectModel

+ project: Project

+__constructor(project : Project)
+valid(): bool
+read(id: int): dbArray
+write(): void

Project

-id : int
-name : string
-owner_Id : int
-deadline : string
-status: int
-priority: int
-hours: string
-note: string
-creation_Time: string
+usersIds: Array<int>

+__constructor(name : String, owner_id : int, deadline : String, creation_Time : String, note : String) :

ProjectController

+ projectsListPage: ListPage
+ projectPage: ProjectFormPage
+ pmodel: ProjectModel
+ numberOfRows: int
+ startIndex: int
+ currentPage: int
+ header: bool
+ headerLabels: ArrayList

+ __construct (title:string, link:string, numberOfRows: int=10, startIndex: int=0, currentPage:int=0, header:bool=true, table:string)
+ createProject(): void
+ viewProject(projectId: int): void
+ modifyProject(projectId: int): void
+ deleteProject(projectId: int): void
+ viewProjectsList(): void
+ exportProject(projectId: int): void
+ sortProjectsList(): void
+ nextPageIndex(): int

implement

<<interface>>
FormValidationController

+purifyInput(input: string): void

implement

ProjectFormPage

+ __construct (title:string, link:string)
+ render(): void

loginFormPage

+ __construct (title:string, link:string)
+ render(): void

Extends

AuthenticationController

- loginForm: loginFormPage

- __construct()
+ login(): bool
+ logout(): bool
- initialize(userId: int) : void
+createUserAccount(): void
+checkAuthentication(): bool

Extends

Page

title: string
link: string
msg: Message
- stmt: dbArray

+setFields(stmt: dbArray): void
+ *render*(): void
+setMessage(text: string): void

implement

<<interface>>
Displayable

+ render(): void