Project Part 2
CSCI 5448
10/20/15

1. Project Summary
Title: Time Management Web Application

Team: Mohammad Alasmary, Kelly Goodman, Kevin Holligan, Elizabeth Lor

A web-based application that tracks the hours employees work on specific projects. This organizational project tracker that is tailored for project contributors to be able to manage their individual time. It additionally sums contributor hours to allow project managers to manage human resources.

Platform/Environment: PHP for the web application; MySQL for database storage; Apache for server hosting.

2. Project Requirements

Functional Requirement				
ID	**Requirement**	**Area**	**User**	**Priority**
FR-1	Users can create an account by registering an email and username.	Database, Authentication	Any user	High
FR-2	Users can login and logout from their account	Database, Authentication	Authenticated users	High
FR-3	Only the user of the account can modify account information	Database	Authenticated users	Low

Non-Functional Requirement			
ID	**Requirement**	**Area**	**Priority**
NR-1	Use third party libraries to enhance UI widgets	User Interface	Very-low
NR-2	The system should encrypt sensitive information such as users' passwords	Security	High

NR-3	Unauthorized users should not be able to use or make changes to the system	Security	High

User Requirement			
ID	**Requirement**	**User**	**Priority**
UR-1	As a project leader, I want to be able to create projects so that I can track hours against them	Project manager	High
UR-2	As a project leader, I want to be able to add users to projects I've created so that they can enter their hours on those projects.	Project manager	High
UR-3	As a project participant, I want to be able to enter time on projects that I'm assigned to as well as projects that I have created so that I can record my hours.	Project participant	High
UR-4	As a user, I want to be able to view the details of my account so that I can verify the information and make changes if necessary.	All users	Medium
UR-5	As any user, I want to be able to modify the priority of my projects so that I can organize my time.	All users	Low
UR-6	As any user, I want to be able to export my hours worked into an xls (csv) spreadsheet so that I can review all of my projects.	All users	Very Low
UR-7	As a project leader, I want to be able to edit the details of my projects (including name, notes, etc) so that I can adjust my projects to reflect changing scenarios	Project manager	Low
UR-8	As a project leader, I want to be able to remove users from projects I've created.	Project manager	High
UR-9	Allows any user to view what projects they are the owners of or are assigned to	All users	High
UR-10	A list of users will be displayed when the user selects a project.	All users	High

Business Requirements

ID	Requirement	Area	Priority
BR-1	All users must have a registered account.	Authentication	Low

3. Users and Tasks (Use-Case Diagrams)

Use Case Name:	FR-01 Create a new user account	
Description:	The user can create an account by registering a username and valid email address. The user enters name, date of birth, and employer name.	
Actors:	User, Database	
Pre-conditions:		
Post-conditions:	The user's name, date of birth, employer, username, and email address are stored in the database.	
Flow of Events:	**Actor Action**	**System Response**
	1. The user clicks a link to sign up for the time management service.	The system goes to a new landing page with fields for entering registration information
	2. The user enters prompted information into text entry fields (first and last name, date of birth, employer name, email address, username, and password). Duplicate entries for email address and password as validation	
	3. The user hits a button to create account once all the required fields are entered	The system takes the user to the main landing page and records all the entered information in a database
Exceptional Events flow:	1. The user enters an email address that has already been used by another account	The interface prevents the user from registering the account and prompts the user with an error message stating that the email has already been used.
	2. The user enters a username that has already been used by another account	The interface prevents the user from registering the account and prompts the user with an error message stating that the email has already been used.
	3. The user's password does not meet the password requirements	The interface
Variations:		

Notes and Issues:	

Use Case Name:	FR-02 Login	
Description:	A user can login into the system via this function by providing the correct username and password. The function also supposed to handle failure to provide correct info or make a connection to the DB.	
Actors:	Registered users	
Pre- conditions:	A user should have an account in the system database.	
Post- conditions:	● A user should be marked as logged in. ● Date, time, should also be activated to start registering. ● A user sees the main page or login page.	
Flow of Events:	**Main**	**Alternative**
	1. A user clicks on login link. 2. The system displays the login page. 3. A user fill the username and password fields. 4. The user click login button 5. The user will be flagged as signed in the DB system and other information will be initialized. 6. The system forwards the user into the main page of the system.	
Exceptional Events flow:	1. The user provides wrong username or password. The system counts attempts and print message. 2. The user reaches the limit of failure attempts, the account will be suspended and message provided. 3. The system cannot make a connection to the DB. Print message. 4. The user already logged-in from another place. Print message	
Variations:	1. A user can sign-out and all the required information will be registered in his file and then erased from the session. 2. The user forwarded to the login page.	
Notes and Issues:	The function also implements NR-3 security feature.	

Use Case Name:	FR-03 View a user account
Description:	The user can access their profile information
Actors:	User, Database

Pre-conditions:	The user has an account created and is logged in to their account	
Post-conditions:		
Flow of Events:	**Actor Action**	**System Response**
	1. User clicks a link to access their account/account information	The system takes them to a new landing page and loads the user information from the database
Exceptional Events flow:		
Variations:		
Notes and Issues:		

Use Case Name:	FR-04 Modify a user account	
Description:	The user can make changes to their profile information	
Actors:	User, Database	
Pre-conditions:	The user has an account created and is logged in to their account. They have landed on the user account page.	
Post-conditions:	Any changes made to the account are saved to the database.	
Flow of Events:	**Actor Action**	**System Response**
	1. The user clicks an icon that allows them to edit the information in the field (e.g. pencil icon).	The field becomes enabled for editing
	2. Once the user enters information they hit enter or select a check mark to confirm the changes	The new information is represented in the field
	3. The user hits a “Save Changes” button to confirm all the edits they’ve made and save the profile.	The user’s information is updated in the database.
Exceptional Events flow:		

Variations:	1. Passwords must be reset through an email link and cannot be edited directly in the account information page	An automatic email is sent to the users registered email address which contains the link to enter a new password
	2. The email associated with the user cannot be changed as this is the primary key for the user	
Notes and Issues:		

Use Case Name:	UR-01 Create new project	
Description:	Allows managers to create new projects in the system	
Actors:	User	
Pre-conditions:	User is logged in	
Post-conditions:	A new project will be added to the database; The project will have attributes (e.g. owner, description, deadline, project participants) that may or may not be defined at creation	
Flow of Events:	**Main**	**Response**
	1. User selects to create new project 2. User fills in web form, submits	1. System authenticates use. System prompts the user with a web form to define the project attributes (e.g. description, owner, deadline, participants). 2. System generates a new project with the defined attributes and adds it to the system project database.
Exceptional Events flow:	1. The creator listed another user as the project owner	1. System notifies the user listed as owner to accept or reject the new project. If accepted, project is added to database. Else, project is not added.
Variations:		
Notes and Issues:		

Use Case Name:	UR-02 Add users to a project
Description:	The system will allow the managers to add users to a project.
Actors:	Managers

Pre- conditions:	Manager is logged in; user is not already on the project	
Post- conditions:	Users will be added to the project	
Flow of Events:	Action	Response
	1. Manager selects to see the list of projects that they manage	System will show a list of projects that the manager manages
	2. Manager selects which project they would like to view	System will show a list of users on that project
	3. Manager selects add users	System will list users that are not currently on the project
	4. Manager selects the user	System will add the user to the project and update the list
Exceptional Events flow:	1. User closes the application	System makes no change
Variations:	1. User closes the prompt without making a selection	System makes no change
	2. User does not make a selection (due to not noticing prompt or lack of activity).	System removes prompt after a fixed interval and makes no change
Notes and Issues:		

Use Case Name:	UR-05 Change priority	
Description:	The authenticated user can sort the projects based on some criteria (deadlines, number of users, priority). The user also able to change the priority of the projects.	
Actors:	Authenticated users	
Pre- conditions:	- The user has logged in. - The has more one than one project. - The user has already displayed the projects.	
Post-conditions:	- The projects sorted based on the user choice criteria.	
Flow of Events:	**Main**	**Alternative**
	1. On the display projects page, the user selects the criteria. 2. The user selects sort button. 3. The system sorts the project based on the chosen option. 4. The system displays the sorted projects.	1. On the display projects page, the user selects custom. 2. The user specify low, normal, or high for a project 3. The rest is the same starting from step 2 in the main flow.

Exceptional Events flow:	
Variations:	
Notes and Issues:	All the projects considered normal priority until the user changes them under custom option.

Use Case Name:	UR-06 Export Information	
Description:	The users save their own projects information into downloadable formats, e.g. xls.	
Actors:	Authenticated users	
Pre- conditions:	- The user has logged in. - There is at least one project owned by this user.	
Post- conditions:	- A file downloading process should have started.	
Flow of Events:	**Main**	**Alternative**
	1. The user clicks on export-to-file link for a specific project. 2. The system starts downloading.	
Exceptional Events flow:		
Variations:	1. The user could be provided with several file formats and the user needs to choose before step 1 in the main flow.	
Notes and Issues:	Assuming the user will export one project at a time, one format (xls), with predefined structure for simplicity.	

Use Case Name:	UR-07 Modify project	
Description:	Allows users to make changes to the attributes of their own projects	
Actors:	User	
Pre- conditions:	User is logged in; user must be 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	1. System verifies if the user is authenticated and

	selects to make changes to a project 2. User fills in web form, submits	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 system queries the existing project's attributes and pre-fills the fields with the current values. 2. System reassigns project attributes with the newly defined values.
Exceptional Events flow:	1. User is not owner of project 2.One of the edits the user makes is to transfer ownership of the project	1. System notifies the user, exits the process, and redirects the user to the home page. 2. System notifies the new user as owner to accept or reject the new project. If accepted, the new user becomes project owner.
Variations:		
Notes and Issues:		

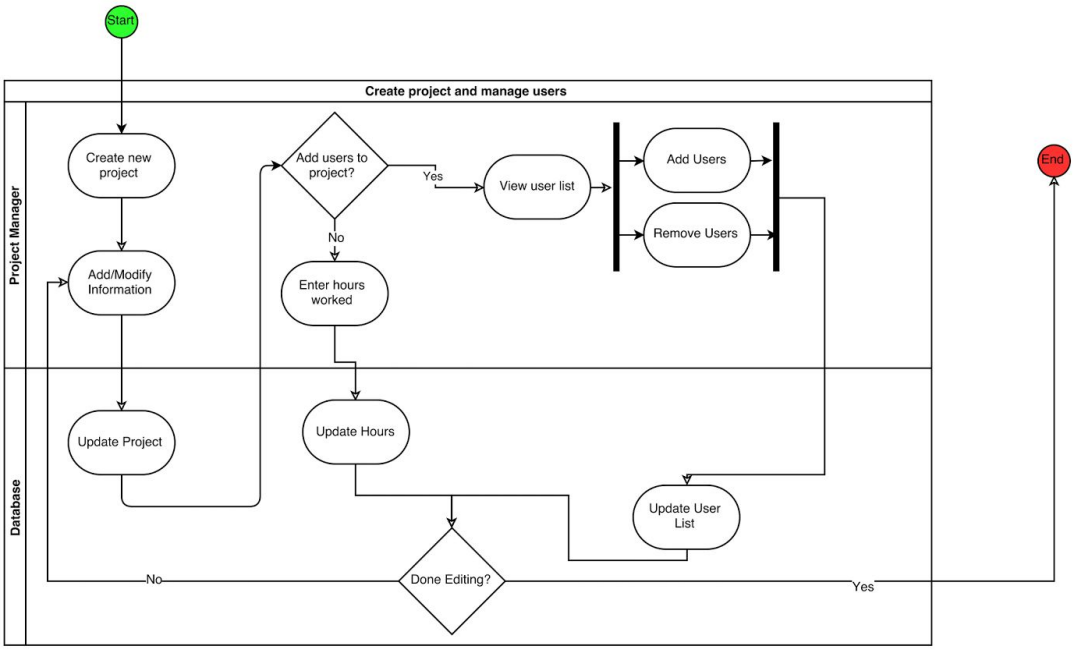
Use Case Name:	UR-08 Remove users from a project	
Description:	The system will allow the managers to remove users off a project.	
Actors:	Managers	
Pre- conditions:	Manager is logged in; user is working on the current project	
Post- conditions:	User will be removed from the project	
Flow of Events:	Action	Response
	1. Manager selects to see the list of projects that they manage	System will show a list of projects that the manager manages
	2. Manager selects which project they would like to view	System will show a list of users on that project
	3. Manager selects the user they would like to remove from the project	System will remove the user from the project and update the list
Exceptional Events flow:	1. Manager closes the application	System makes no change
Variations:	1. Manager does not make a selection (due to not noticing prompt or lack of activity).	System removes prompt after a fixed interval and makes no change
Notes and Issues:		

Use Case Name:	UR-09 View projects	
Description:	Allows any user to view what projects they are the owners of or are assigned to	
Actors:	Users	
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. For projects the user is an owner of, options to edit or add users appear alongside the listing. For projects the user is a participant of, options to enter hours or leave project appear alongside the listing.
Exceptional Events flow:		
Variations:		
Notes and Issues:		

Use Case Name:	UR-10 View users in a project	
Description:	The system will allow the users to view who is working in a project. A list of users will be displayed when the user selects a project.	
Actors:	Managers and Project Participants	
Pre- conditions:	User is logged in; user has been assigned to a project	
Post- conditions:	User will get a list of other users working on the project	
Flow of Events:	Action	Response
	1. User selects to see the list of projects they are working on	System will show a list of projects that the user is on

	2. User selects which project they would like to view	System will show a list of users on that project
Exceptional Events flow:	1. User sign out	System will take user back to the login screen
Variations:	1. User does not make a selection (due to not noticing prompt or lack of activity).	
Notes and Issues:		

4. Activity Diagram



5. Data Storage

We will use MySQL to store all the system's data as it is easy to interface with PHP. The data will be organized into three tables, users, projects, and user_project.

The users table will have the fields id, username, password, and date_joined.

```
mysql> DESCRIBE users;
+-------------+------------------+------+-----+-------------------+-----------------------------+
| Field       | Type             | Null | Key | Default           | Extra                       |
+-------------+------------------+------+-----+-------------------+-----------------------------+
| id          | int(10) unsigned | NO   | PRI | NULL              | auto_increment              |
| username    | varchar(16)      | YES  |     | NULL              |                             |
| password    | varchar(16)      | YES  |     | NULL              |                             |
| date_joined | timestamp        | NO   |     | CURRENT_TIMESTAMP | on update CURRENT_TIMESTAMP |
+-------------+------------------+------+-----+-------------------+-----------------------------+
```

The projects table will have the fields id, name, description, deadline, and owner. The values of owner are the user_id's of the employees who created the respective projects.

```
mysql> DESCRIBE projects;
+-------------+------------------+------+-----+---------+----------------+
| Field       | Type             | Null | Key | Default | Extra          |
+-------------+------------------+------+-----+---------+----------------+
| id          | int(10) unsigned | NO   | PRI | NULL    | auto_increment |
| name        | varchar(128)     | YES  |     | NULL    |                |
| deadline    | date             | YES  |     | NULL    |                |
| description | varchar(256)     | YES  |     | NULL    |                |
| owner       | int(10) unsigned | YES  |     | NULL    |                |
+-------------+------------------+------+-----+---------+----------------+
```

Because MySQL tables cannot store arrays, the user_project table will store information regarding which users participate on which projects. It will have the fields user_id, and project_id.

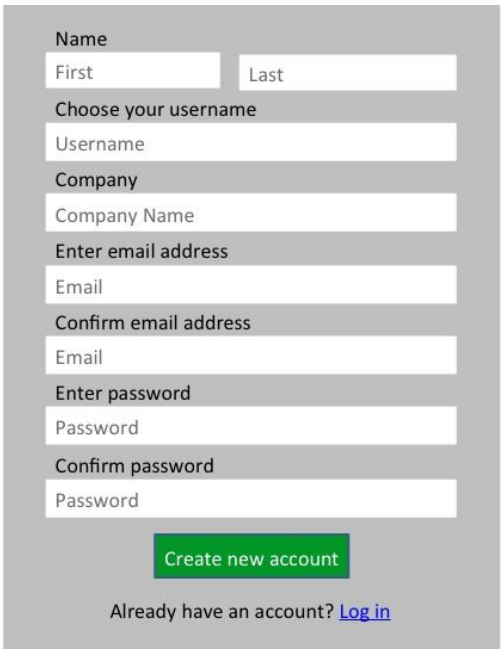
```
mysql> DESCRIBE user_project;
+------------+------------------+------+-----+---------+-------+
| Field      | Type             | Null | Key | Default | Extra |
+------------+------------------+------+-----+---------+-------+
| user_id    | int(10) unsigned | NO   | PRI | NULL    |       |
| project_id | int(10) unsigned | NO   | PRI | NULL    |       |
+------------+------------------+------+-----+---------+-------+
```

PHP classes, UserAcountController and ProjectController, will be developed to interact with the MySQL database. These controller classes provide separation between the database model and the client-side view. For instance, UserAccountController() has the method modifyUserAccount() that modifies the users table depending on the $_POST variables the user submitted in the "register account" or "edit settings" view. Similarly, ProjectController has methods createProject() and modifyProject() that modifies the projects and user_project tables depending on the client-submitted data.

6. UI Mockups

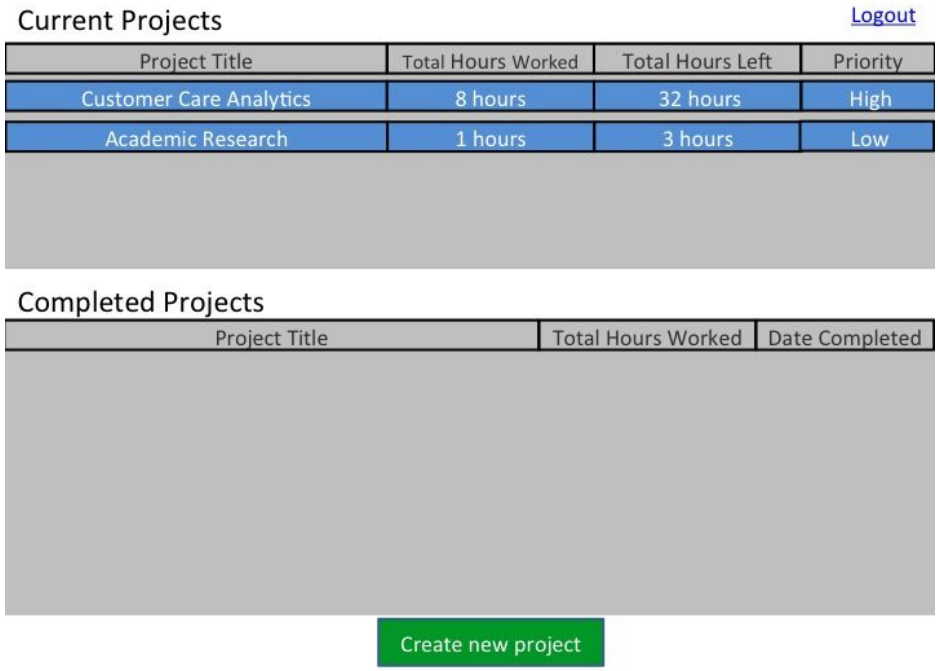
Login Screen

User logs in with their email address and password. If the user doesn't have an account, they can sign up for a free account.



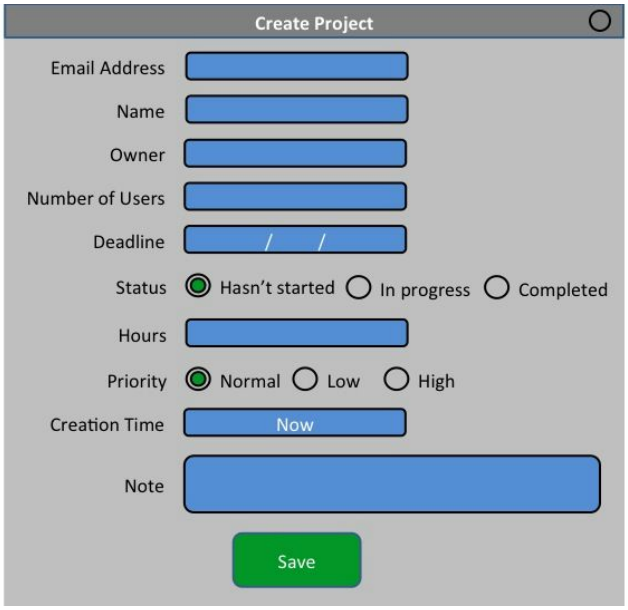
Register Screen

By filling out the required fields, the user is able to create a new account. Clicking Create new account will take the user to the Main Screen. Clicking the Log in link will take the user back to the Login Screen.



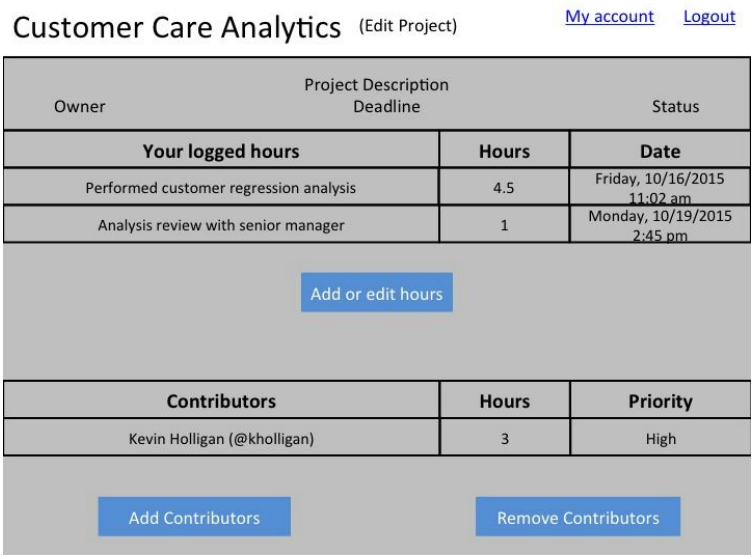
Main Screen

This screen will show the list of projects that the user is currently on or have completed. The list of projects can be sorted by the Project Title, Total Hours Worked, Total Hours Left and Priority by clicking on the labels. Clicking on the Logout (top right) will take the user back to the Login Screen. Clicking on the Create new project (bottom) will take the user to the Create New Project Screen. Clicking on the project name will take the user to the Project Screen.



Create New Project Screen

To create a new project, the user will fill out the necessary field. Clicking the Save button will create the project and take the user back to the Main Screen. The new project will be listed in the Current Project table.

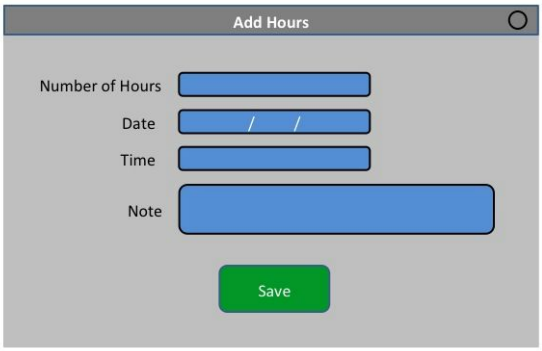


Project Screen

Details about the project along with the amount of hours that the user has worked and the contributors to the project are display. The users can view how many hours they have worked on the project along with who is also working on the project with them. Clicking on the Add or edit hours button will take the user to the Add Hours Screen. Clicking on the Edit Project link will take the user to the Modify Project Screen. Clicking on the Add Contributors button will take the user to the Add Users Screen. Clicking on the Remove Contributors button will take the user to the Remove User Screen. Clicking on the My account link will take the user back to the Main Screen.

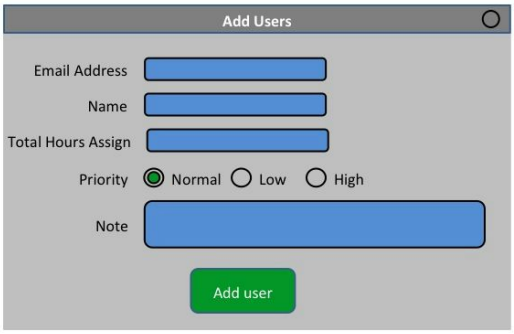
Modify Project Screen

By changing the information, the user is able to edit the project. Clicking the save button will update the project information and take the user back to the Project Scree.



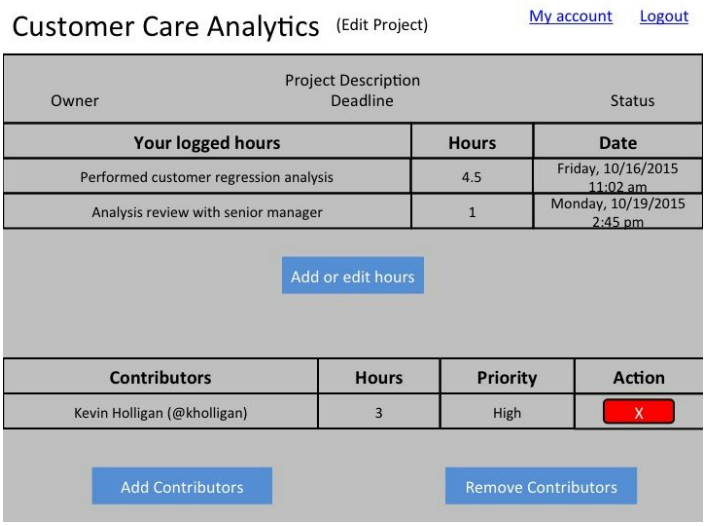
Add Hours Screen

By filling out the necessary information, the user is able to record the hours that they worked on for that project. Clicking on the Save button will record the hours and take the user back to the Project Screen.



Add Users Screen

A manager for a project can add a user to their project by filling out the necessary information. Clicking on the Add user button will add that user to the project and take the user back to the Project Screen.



Remove User Screen

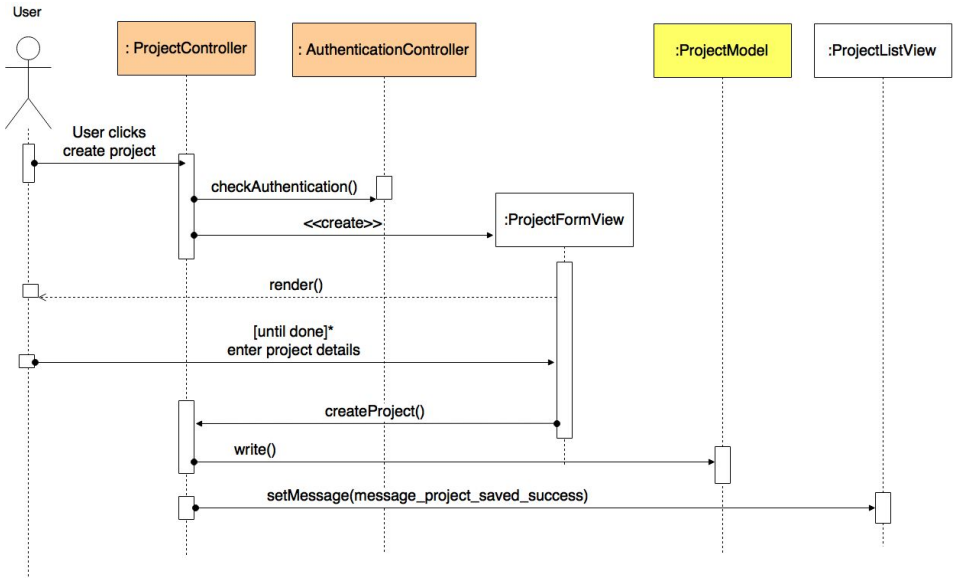
The manager of the project can remove a user by clicking on the red button, next to the user's information. Clicking on the red button will remove the user and take the user back to the Project Screen.

7. User Interactions

at least 3 interactions descriptions and sequence diagrams b/w user and app without conditional constructions and if-else/use multiple diagrams instead. For each interaction, describe how your system will support it.

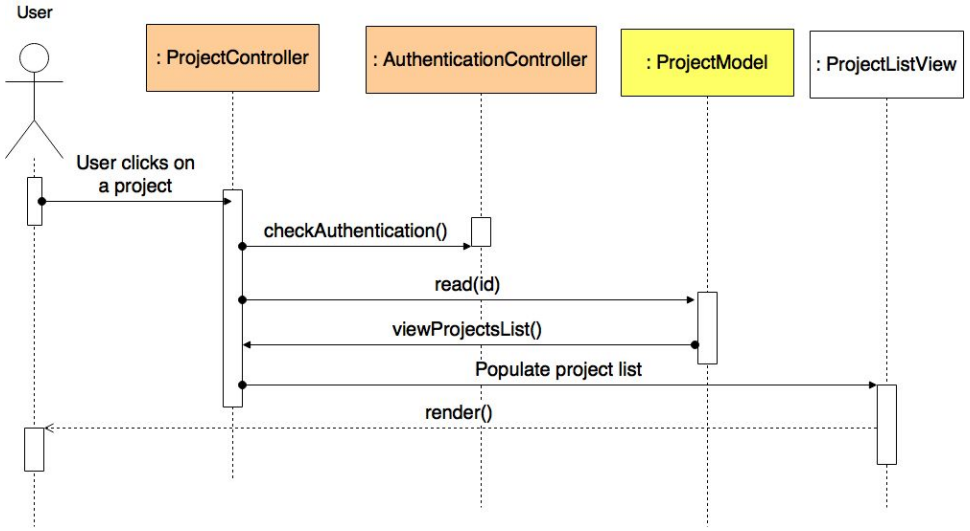
Create Project

The system will create the project with the necessary information provided by the user. It will also add the project to the list of projects so that user can view them.



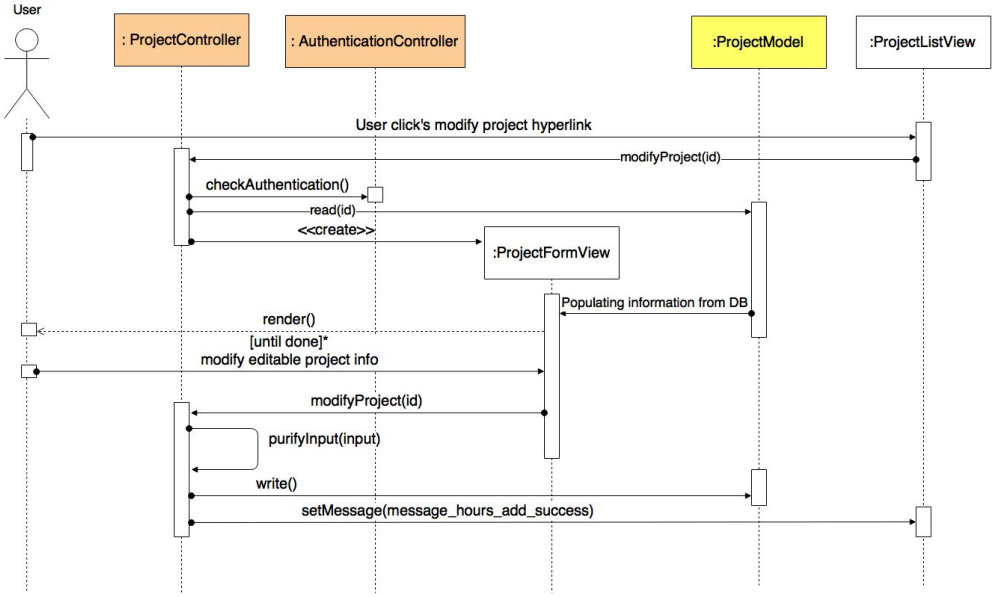
View Project

The system will keep the project updated and show a list of projects that the user is in.



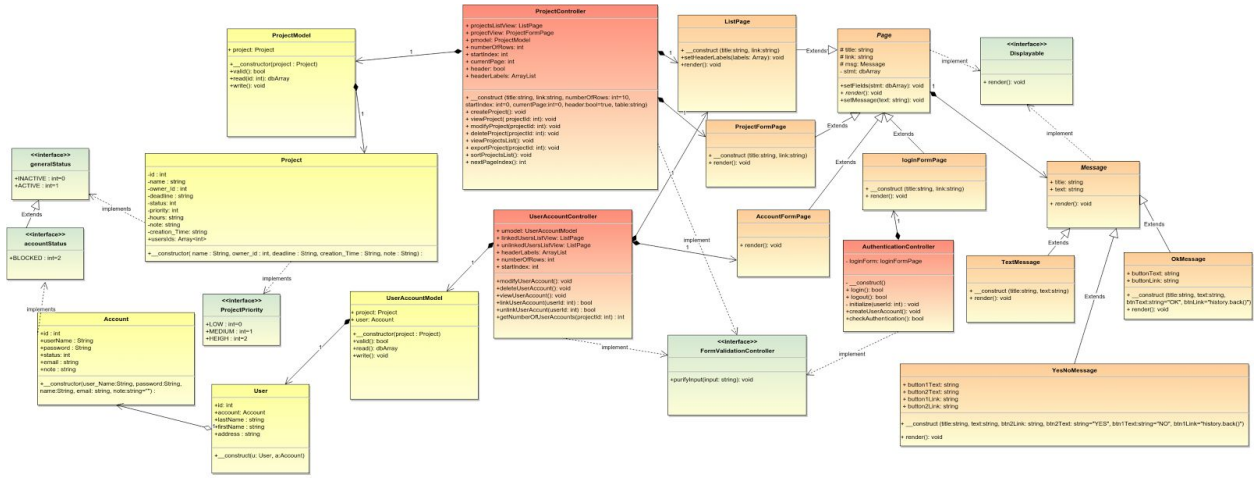
Enter Hours

The system will update the user's hours ba



8. Class Diagram

Draw a class Diagram for the entire project and its relationships



(more detailed view on github)