Part A. User requirements.

System requirements:

1. System allows users to Register or to Login.
2. System should be able to provide to the users to create Companies, Projects, Users and Tickets.
3. Software provides assigning Projects to Companies, Users to Project and Tickets to Users functionality.
4. Software allows users to change the state of the Tickets.
5. System allows to create a board to show up Tickets.
6. Allow users to produce reports that shows productivity of users and development processes.
7. Allows to predict the number of projects and users that company will need in future, based on historic data.
8. System allows users to Logout or to Delete account.

Non-functional requirements:

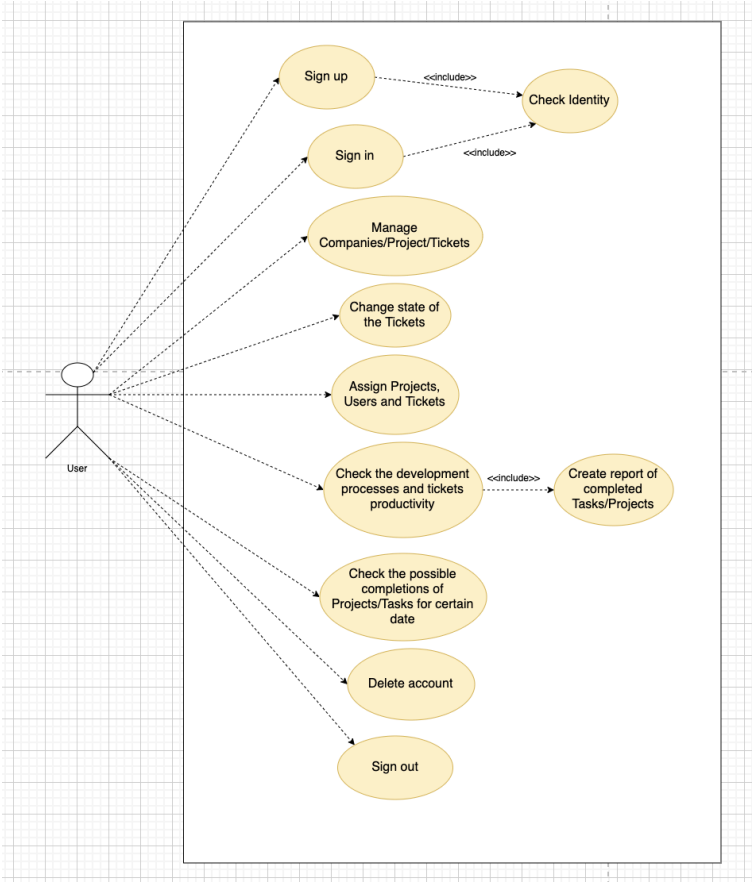
1. The software must run on Windows operating system.
2. The system must be implemented using C# programming language and Blazer.

Actors:

1. User

Part B. Use Cases.

Use case model/diagram



Use case descriptions

Sign Up

Use case name	Sign Up
ID	UC-001
Description	A new user is being registered to the application by adding his details and setting up a password
Primary actor	User (new user)
Supporting actor	-
Stakeholders and interests	-
Pre-conditions	1. User have to see/skip the onboarding; 2. User has to be new; 3. Application must be connected to the database system in order to save/store User's data;
Post-conditions	1. The unique new User's account is created; 2. User's data is saved; 3. Main page is presented;
Trigger	New User wants to register into the system, clicks sigh up button after adding his data
Main success scenarion	1. User selects New account; 2. User adds his name, username and password; 3. User clicks the sign up button; 4. Loading indicator is started; 5. System checks the spelling and special symbols; **include::Check Identity** 6. New account is created; 7. Loading indicator is terminated; 8. Main page is presented;
Variations	-

Sing In

Use case name	Sign In
ID	UC-002
Description	Already registered user is getting access to the application and data by verifying himself
Primary actor	User (new user)
Supporting actor	-
Stakeholders and interests	-

Use case name	Sign In
Pre-conditions	1. User data has to be stored in the database; 2. Application must be connected to the database system in order to verify User's data;
Post-conditions	1. User's account is verified; 2. Main page is presented;
Trigger	User wishes to login into the System;
Main success scenarion	1. User adds his username and password; 2. User clicks the sign in button; 3. Loading indicator is started; 4. System checks the spelling and special symbols; **include::Check Identity** 5. Loading indicator is terminated; 6. System checks the Database if the User has any Companies created; 1. If no, System executes the Create Company Use case with an option go to the main page; 2. If yes, System presents the main page; 7. Main page is presented;
Variations	-

Check Identity

Use case name	Check Identity
ID	UC-003
Description	User's details need to be verified as accurate
Primary actor	Database System
Supporting actor	-
Stakeholders and interests	-
Pre-conditions	User details must be provided to the System and the to the Database
Post-conditions	Details are returned as verified or checked
Trigger	User wishes to login into the System
Main success scenarion	1. User's details are received; 2. User's account is verified; 3. Details are returned as verified or checked;
Variations	-

Manage Companies, Projects and Tickets

Use case name	Manage Companies, Projects and Tickets
ID	UC-004
Description	User can perform CRUD operations for Company/Project/Ticket
Primary actor	User
Supporting actor	-
Stakeholders and interests	-
Pre-conditions	User must have been signed in to the System
Post-conditions	System executes CRUD operations related to Company/Project/Ticket
Trigger	User wishes to perform CRUD operations relative to the Company/Project/Ticket
Main success scenarion	1. User's selects CRUD operation button towards Company/Project/Ticket; 2. System shows specific page for selected option; 3. User manages details of Company/Project/Ticket; 4. User presses CRUD button; 5. System checks if the all blanks are filled(CU); 6. System sends data to the Database system; 7. Database system returns successful result after executing CRUD operation; 8. System alerts User by showing appropriate page, information;
Variations	-

Change the state of the Ticket

Use case name	Change the state of the Ticket
ID	UC-005
Description	User can updated the state of the Ticket to To Do, Planned, In Process, Done
Primary actor	User
Supporting actor	-
Stakeholders and interests	-
Pre-conditions	1. User must have been signed in; 2. User must have been created or assigned to at least one Ticket;
Post-conditions	Ticket status/state is updated
Trigger	User wishes to update the state of the Ticket

Use case name	Change the state of the Ticket
Main success scenarion	1. User selects the Ticket; 2. User selects the state for the Ticket to update; 3. System sends Update operation with the new state to the Database system; 4. Database system updates the state and returns the successful result; 5. System alerts User about update; 6. System updates the UI;
Variations	-

Assign Projects, Users and Tickets

Use case name	Assign Projects, Users and Tickets
ID	UC-006
Description	User can assign Projects to Companies, Users to Projects and Tickets to Users
Primary actor	User
Supporting actor	-
Stakeholders and interests	-

Pre-conditions	1. User must have been signed in;
	2. User must have been:
	  1. created a Company to assign Project into it;
	  2. created/assigned to the Project to assign Users into it;
	  3. created/assigned a Ticket to assign User into it;

Post-conditions	Project is assigned to the Company or User is assigned to the Project or Ticket is assigned to the User.
Trigger	User wishes to assign User to the Project or Project to the Company or Ticket to the User

Use case name	Assign Projects, Users and Tickets

Main success scenarion	(ex. with ticket assigning to the User);
	1. User selects the Ticket
	  1. User clicks assign User;
	  2. System requests the Database to show the list of all users;
	  3. Database return successful list of users;
	  4. System shows list of users to the User;
	  5. User selects the user from the list;
	  6. System updates the Ticket's assignee and adds the Ticket to the selected user's "tickets list";
	  7. System allows selected user to manage the Ticked;
	2. User selects user in Registered users list;
	  1. System shows selected user's profile;
	  2. User clicks "Assign ticket to this user" button;
	  3. System shows all User's assigned or created tickets;
	  4. perform step 1.6 and 1.7;

Variations	-

Check the development processes and tickets productivity

Use case name	Check the development processes and tickets productivity
ID	UC-007
Description	The report page of the completed projects and tasks is produced for the User
Primary actor	User
Supporting actor	-
Stakeholders and interests	-
Pre-conditions	1. User must have been signed in; 2. User must have been assigned/created a Project; 3. User must have been assigned/created a Ticket/Task;
Post-conditions	System presents the productivity report page to the User
Trigger	User wishes to see the productivity of the Projects and Tasks
Main success scenarion	1. User clicks the Productivity Report button; 2. User selects specific date range (default range is 7 days); 3. System asks Database system if there any completed Projects or Tasks; 4. Database system returns completed Projects and Tasks; 5. System displays results separately to the user;
Variations	-

Check the possible completions of Projects/Tasks for certain date

Use case name	Check the possible completions of Projects/Tasks for certain date
ID	UC-008
Description	The user is provided by a *possible* completed projects/tasks matching the selected date filter
Primary actor	User
Supporting actor	-
Stakeholders and interests	-
Pre-conditions	1. User must have been signed in; 2. User must have been assigned/created a Project; 3. User must have been assigned/created a Ticket/Task; 4. User must have been defined the complexity for Project/Task;
Post-conditions	The board with prediction data is illustrated to the User
Trigger	User wishes to see the predictions of the Projects and Tasks
Main success scenarion	1. User clicks the Predictions button; 2. System searches for any Project/Task; 3. System shows the default predictions for each Project/Task which is 7 days range; 4. User selects specific date range; 5. Loading indicator is started; 6. System filters the projects/tasks that's predicted finishing date is matching the selected by User range; 7. Loading indicator is terminated (during calculation User can perform another activities *concurrently*); 8. System presents results to the User;
Variations	-

Delete account

Use case name	Delete account
ID	UC-009
Description	User can delete his Application account which will remove his data from the database forever
Primary actor	User
Supporting actor	-
Stakeholders and interests	-
Pre-conditions	User must have been signed up/in
Post-conditions	1. System removes User's data from database; 2. System presents the sign up page to the user;

Use case name	Delete account
Trigger	User clicks the delete account button
Main success scenarion	1. User clicks the delete account button; 2. System alerts the User that account deletion will deleted all created only Companies/Projects/Tickets; 3. User presses Next button; 4. System alerts the User that he will be removed from all assigned to him Tickets/Projects; 5. User presses next button; 6. System performs all deletion operations; 7. System show the sign up page to the User;
Variations	-

Sign out

Use case name	Sign out
ID	UC-010
Description	User can logout from the Application and system will retain his data for the next login
Primary actor	User
Supporting actor	-
Stakeholders and interests	-
Pre-conditions	User must have been signed in
Post-conditions	User is sign out and presented the Sign up page by the Appliaction
Trigger	User clicks the sign out button
Main success scenarion	1. User clicks the sign out button; 2. System presents sign up page to the User;
Variations	-

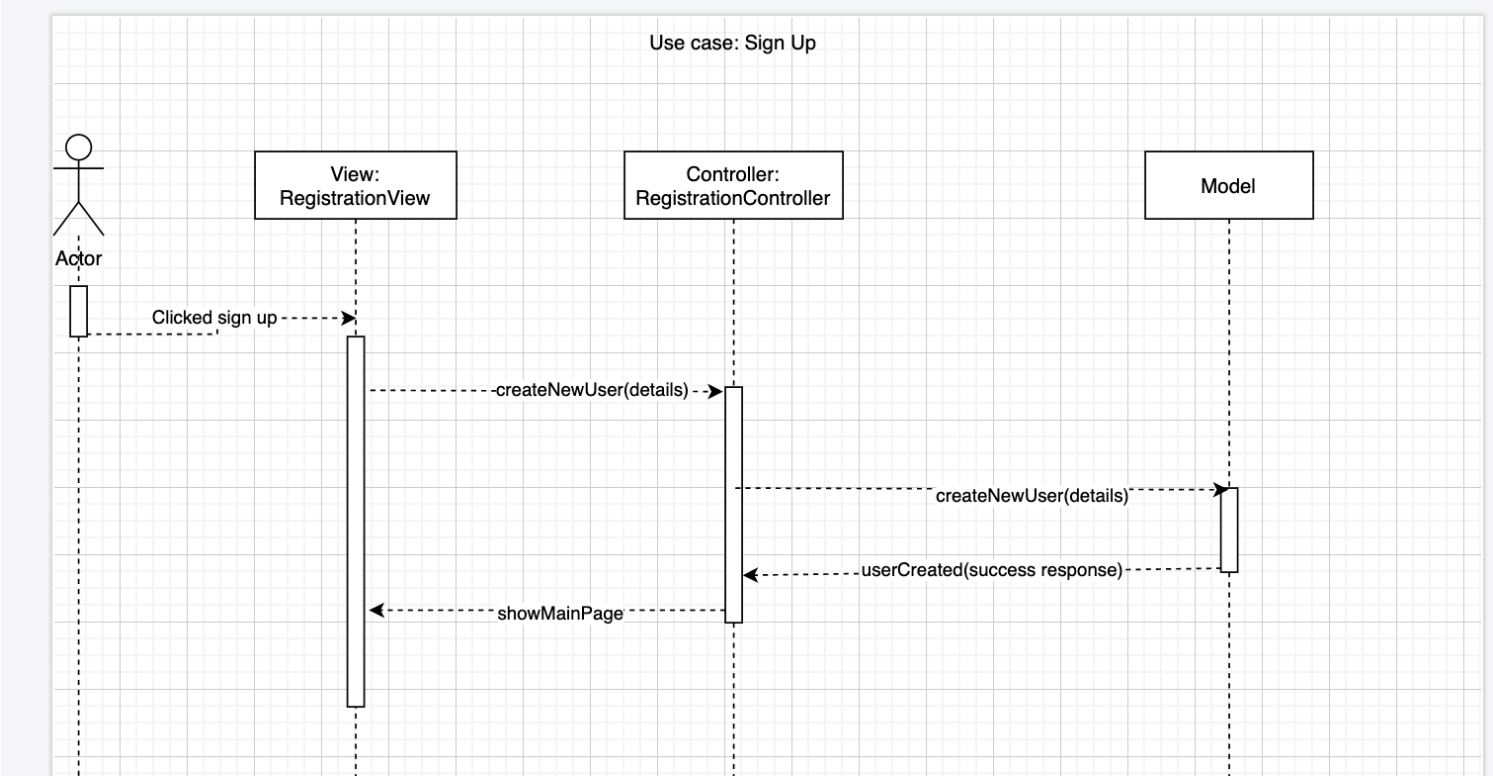
Part E. Activity.

Class, responsibility and collaboration (CRC)

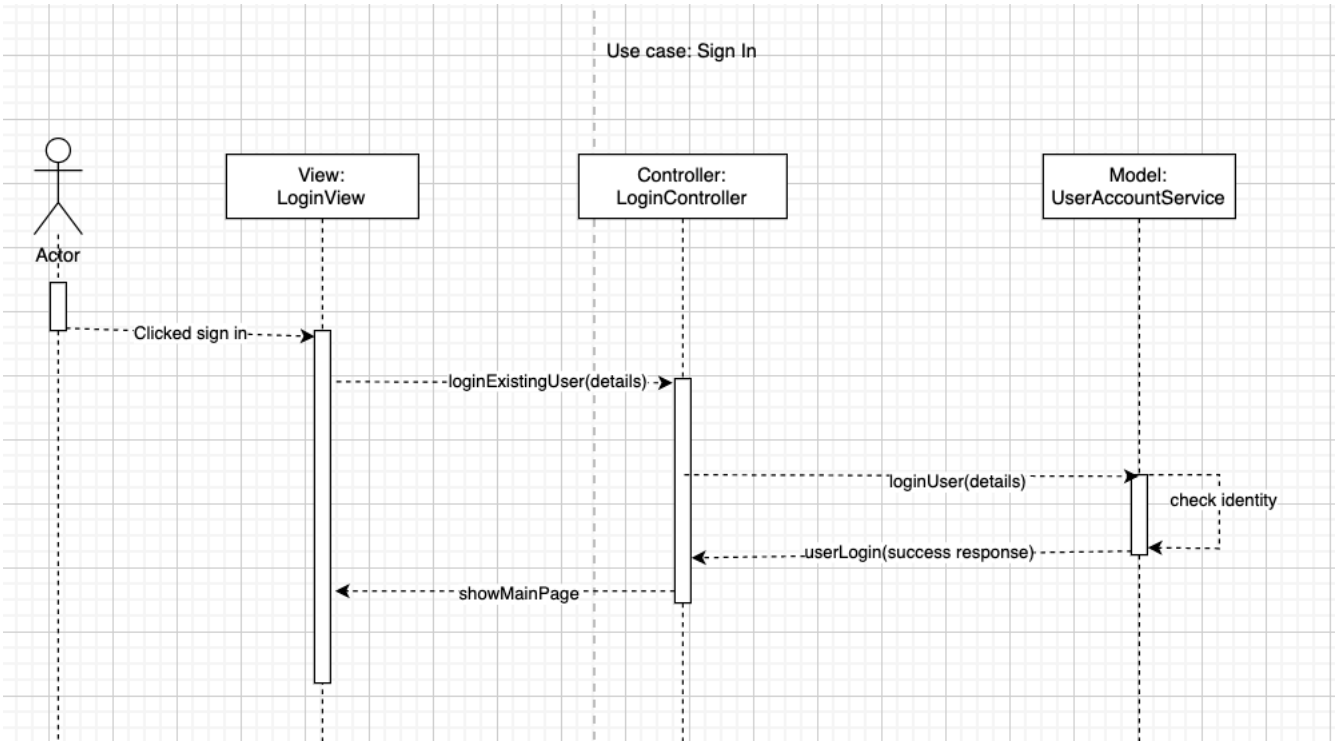
Class name	Type	Responsibilities	Collaborations
User	model	- store data related to it - change its data - provide its data	Project, Company, Ticket
Project	Model	- store data related to it - change its data - provide its data	User
Company	Model	- store data related to it - change its data - provide its data	User
Ticket	Model	- store data related to it - change its data - provide its data	User
UserAccountService	Model	- Highest level class from Model layer providing user account management	UserLoginController User
ComapanyService	Model	- Highest level class from Model layer providing company management	CompanyController Company
ProductivityReportService	model	- Produces productivity report	ProductivityController ProjectService

Part D. Collaboration.

Use Case: Sign Up sequence diagram

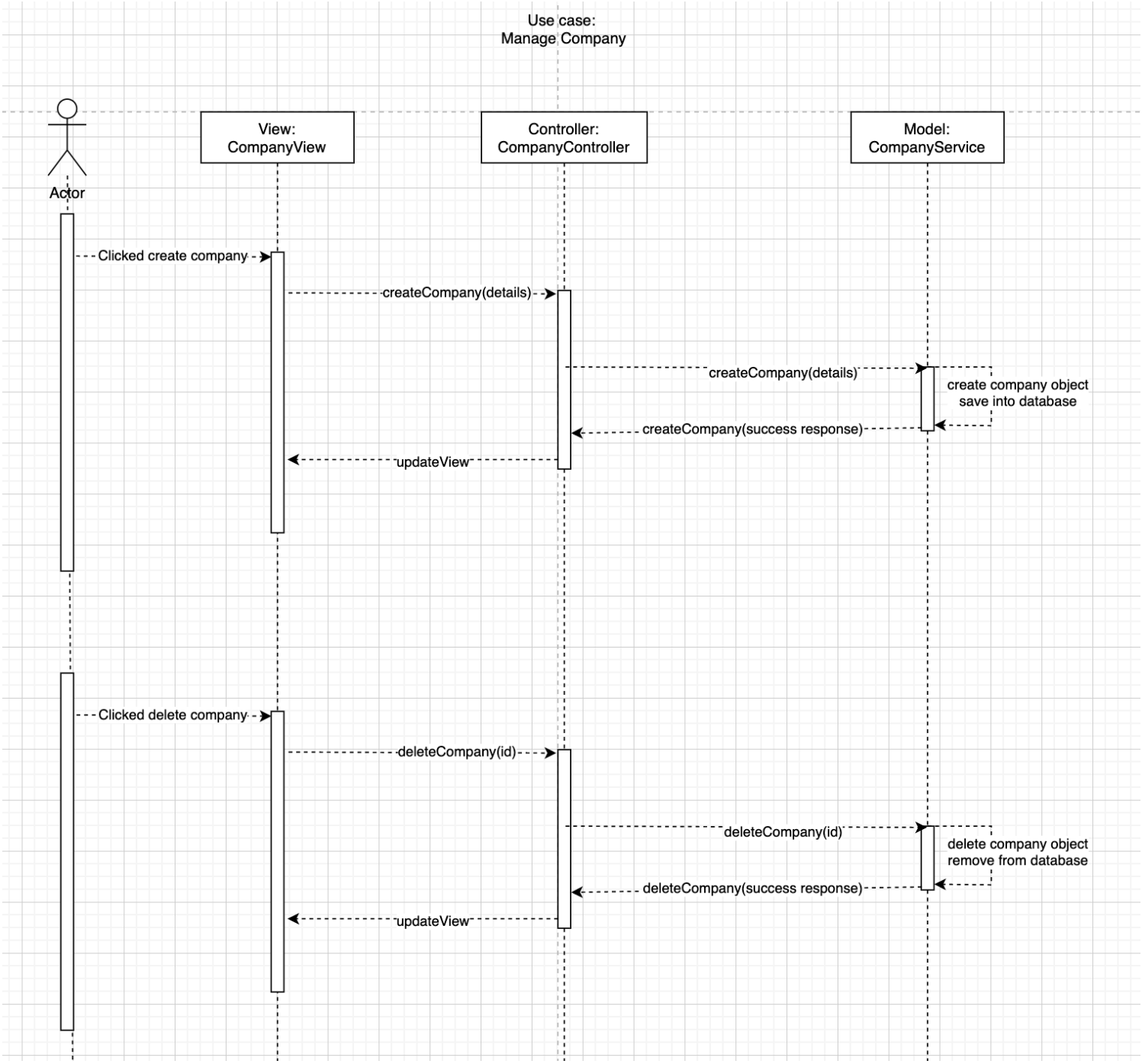


Use Case: Sign In + Check identity sequence diagram

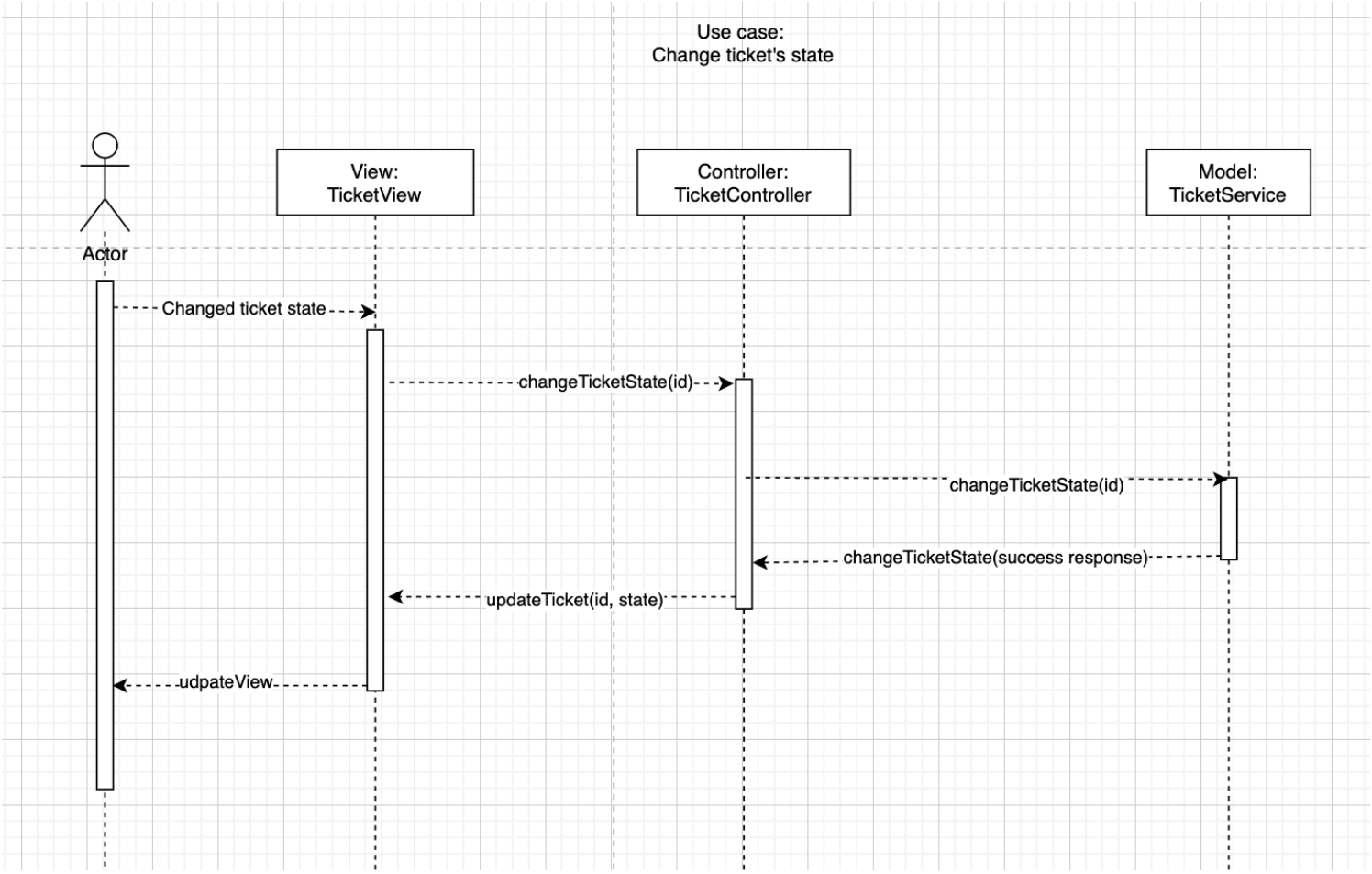


Use Case: Manage Companies/Projects/Tickets sequence diagram

This is only managing the Companies sequence diagram. Other entity management would be in similar manner

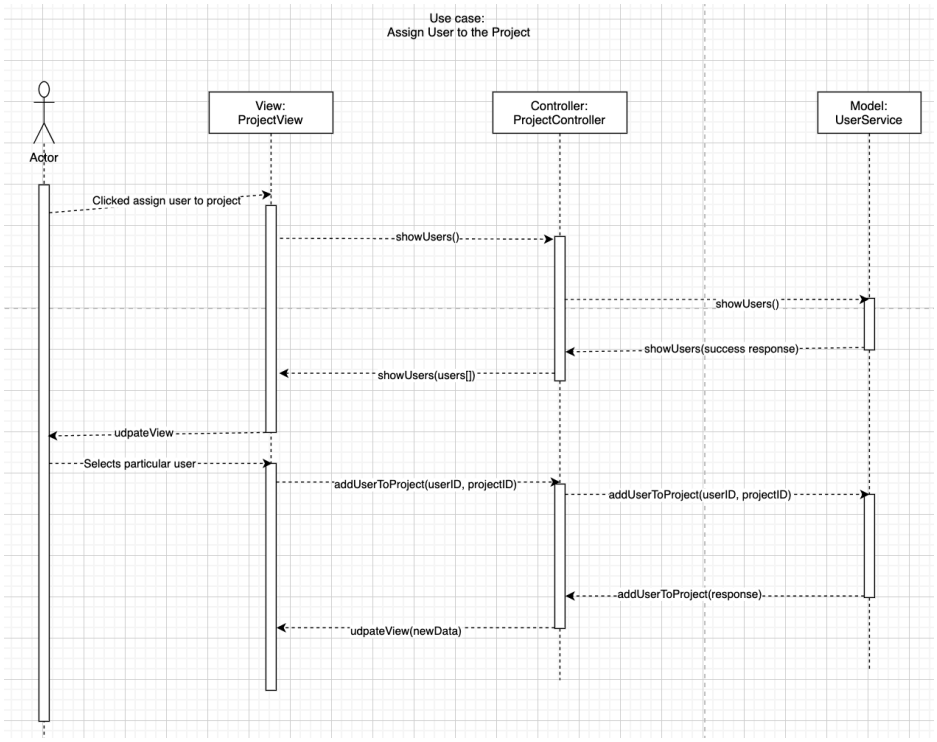


Use case: Change state of the ticket sequence diagram

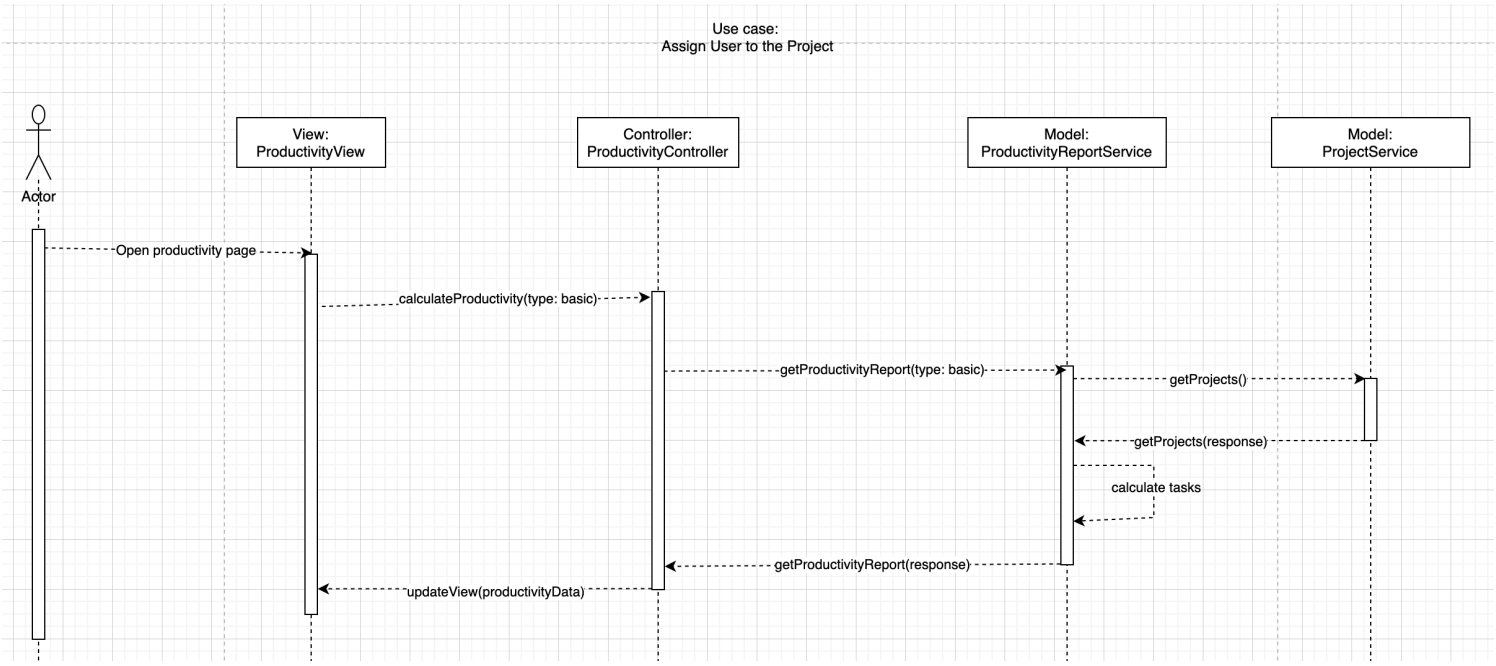


Use case: Assign Projects, Users and Tickets sequence diagram

In this example sequence diagram only for assign User to Project is shown. Other assigning functionality occurs in the same way.

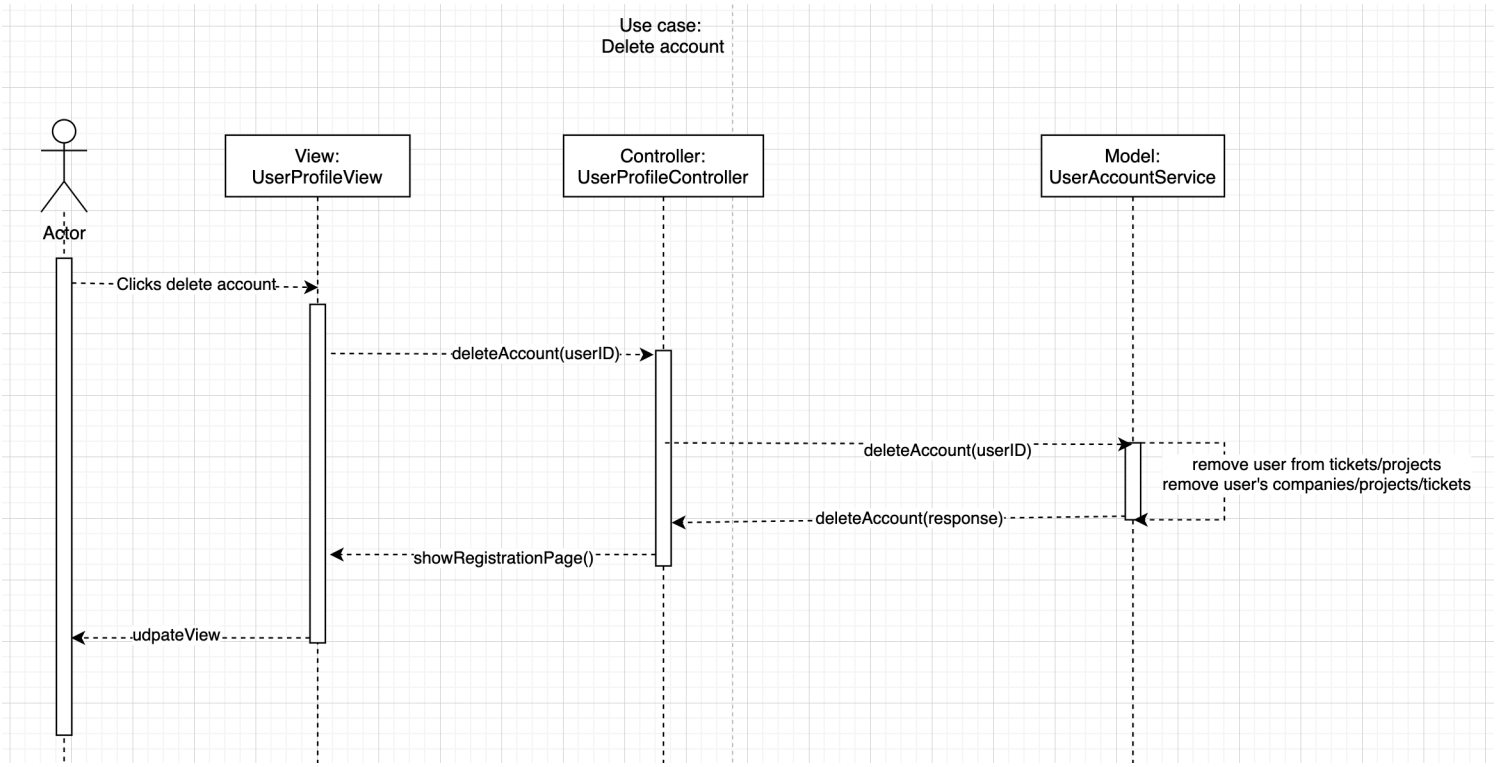


Use case: Check development progress and user productivity

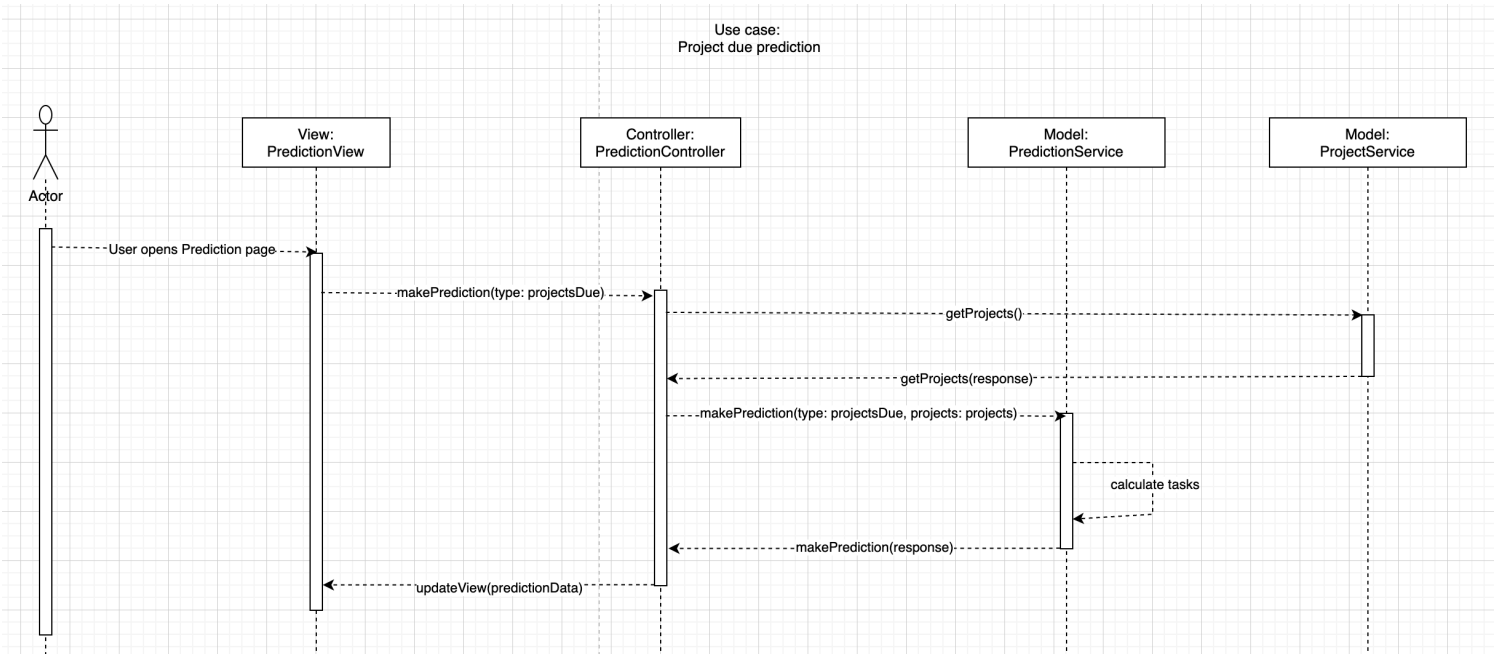


Use case: Delete account sequence diagram

Note: **Sign out** use case sequence diagram is 90% same except deleting any data. That is way the diagram will not provided in order to avoid repeated design within document.



Use case: Check the possible completions of Projects/Tasks for certain date sequence diagram



Part E. Activity.

This activity diagram is only for Predicting the days needed for the Project to be finished by using amount of task, their complexity.