

Part A - Requirements

R1 The software should allow companies to be created

R1.1 The software should allow projects to be created

R1.2 The software should allow users to be created

R1.3 The software should allow tickets to be created

R2 The software allows the users to read and update the following:

R2.1 Companies

R2.2 Projects

R2.3 Other Users

R2.4 Tickets and state of tickets

R3 Projects must be able to be assigned to companies

R3.1 Users must be able to be assigned to projects

R3.2 Tickets must be able to be assigned to users

R3.3 The state of the tickets can be changed by the user

R4 The software allows the user to delete companies

R4.1 The software allows the user to delete projects

R4.2 The software allows the user to delete other users

R4.3 The software allows the user to delete tickets

R5 Software can produce reports

R5.1 The software should display the productivity of users within the report

R5.2 The software should show the progress of development within the report

R6 Users can build a board that displays tickets

R7 The software will allow a user to predict the number of projects that a company will work on and the appropriate number of people needed to build these projects based on previous data

R7.1 The user can produce a prediction for a specific date

NON FUNCTIONAL REQUIREMENTS

NF1 Application will run on Windows and not any other operating system

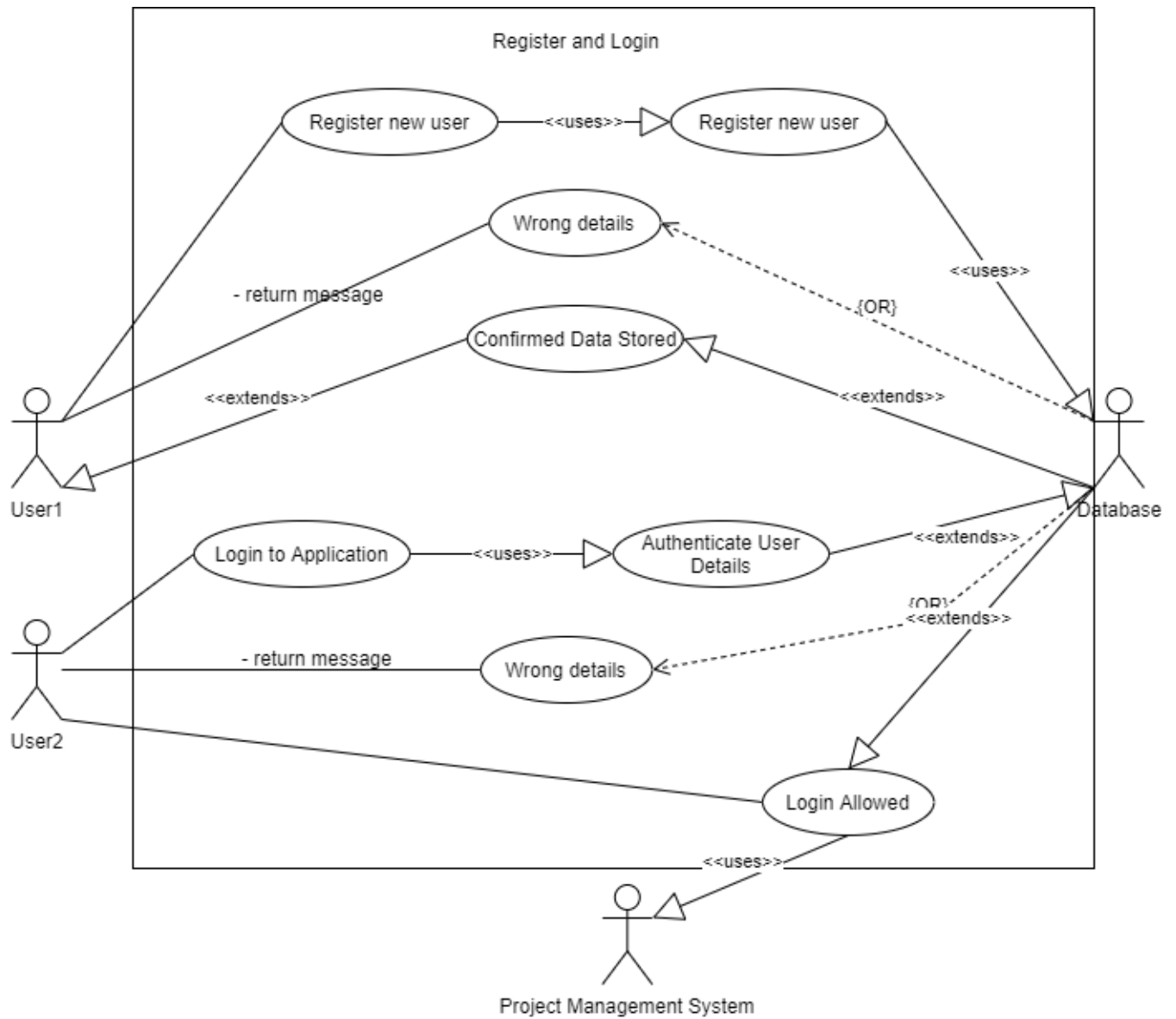
NF2 Application should be a programmatic dynamic interface

NF3 Application data will be saved in a persistent form such as inside a database

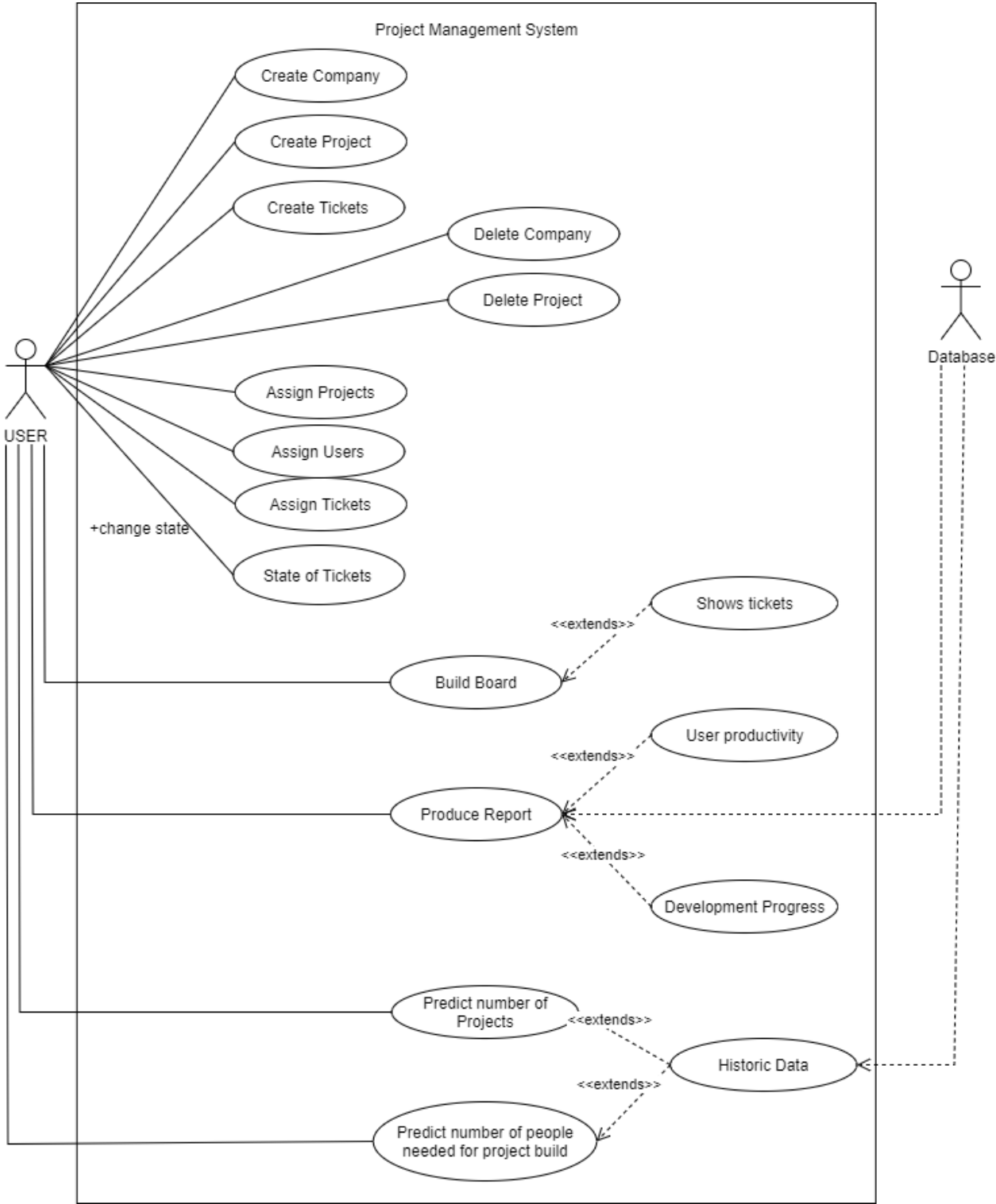
NF4 Application can read and write the data and activity is threaded

NF5 Application is built as a .NET C# application

Part B - Use case Diagrams



Part B - Use case Diagrams II



PART B - USE CASES

Use Case: Register

Id: UC-- 001

Description: Users can register a set of login details unique to them

Primary Actor: User

Supporting Actors: Database

Stakeholders and Interests: No stakeholders and interest

Pre-- Conditions: Not an existing user

Post Conditions: Upon successful registration, the user can use these details to log into the system

Trigger: The user presses the register button

Main Success Scenario

1. The user presses the register button
2. Users enter information to register.
3. The system validates the input data from the user.
4. User details are now stored in the database system.
5. Users can log in using details they used to register

Variations: No variations

Use Case: Login

Id: UC-- 002

Description: Users use the details they registered with and enter them into the login screen

Primary Actor: User

Supporting Actors: Database

Stakeholders and Interests: No stakeholders and interest

Pre-- Conditions: Not an existing user

Post Conditions: Upon successful registration, the user can use these details to log into the system

Trigger: The user presses the register button

Main Success Scenario

1. The user enters their login information
2. The system checks and authenticates user details
3. Access is granted

Variations: No variations

Use Case: Create Company

Id: UC-- 003

Description: This use case will allow the user to create a company.

Primary Actor: User

Supporting Actors: Database

Stakeholders and Interests: No stakeholders and interest

Pre- Conditions: Must be logged in to create a company

Post Conditions: A new company is added

Trigger: The user wants to create a company

Main Success Scenario

1. User selects an option to create company
2. The user enters company details. System checks if the input is valid
3. System updates the database by adding a new company
4. The system displays a new company

Variation: Users can add more companies with an add button as well

Use Case: Create Project

Id: UC-- 004

Description: This use case will allow the user to create a project.

Primary Actor: User

Supporting Actors: Database

Stakeholders and Interests: No stakeholders and interest

Pre- Conditions: Must be logged in to create a project

Post Conditions: A new project is created

Trigger: The user wants to create a project

Main Success Scenario

1. User selects an option to create a project
2. The user enters the project information. System checks if the input is valid
3. System updates and adds a new project
4. The system displays a new project

Variation: User can add more projects with an add button as well

Use Case: Create Tickets

Id: UC-- 005

Description: This use case will allow the user to create a ticket

Primary Actor: User

Supporting Actors: Database

Stakeholders and Interests: No stakeholders and interest

Pre-- Conditions: Must be logged in, a company must be made prior, the ticket can only be created within a project

Post Conditions: A new ticket is added to the project

Trigger: The user wants to create a ticket/task within a project

Main Success Scenario

1. User selects an option to create a ticket
2. The user enters the ticket information. System checks if the input is valid
3. System updates the database of the project and adds the ticket to the chosen project
4. The system displays a new ticket in the project

Variation: Users can add more tickets with an add button as well

Use Case: Assign Project

Id: UC-- 006

Description: This use case will allow the user to assign a project to a chosen company

Primary Actor: User

Supporting Actors: Database

Stakeholders and Interests: Companies - will want to know the specifications of the project and its scope

Pre-- Conditions: Must be logged in, a company must be made prior, a project must already exist

Post Conditions: A project is assigned to a company

Trigger: The user wants to assign a project

Main Success Scenario

1. User selects an option assign project
2. The system displays all free projects
3. User picks a project then picks the company to assign it to
4. The system checks if it's valid then updates the database
5. The system now displays the project within the company

Variation: Users can drag projects into the company

Use Case: Assign Tickets

Id: UC-- 007

Description: This use case will allow the user to assign a ticket to another user

Primary Actor: User

Supporting Actors: Database

Stakeholders and Interests: Other Users - they will want to know the tickets set for them

Pre- Conditions: Must be logged in, a company must be made prior, a project and ticket must already exist

Post Conditions: A ticket is assigned to a user

Trigger: The user wants to assign a ticketed task to another user

Main Success Scenario

1. User selects an option to assign tickets
2. The system displays all created tickets within the project
3. User picks a ticket
4. The system displays users within the company of the assigned project
5. User picks another user
6. The system assigns tickets to the selected user

Variation: None

Use Case: Assign Users

Id: UC-- 008

Description: This use case will allow the user to assign a user to a project

Primary Actor: User

Supporting Actors: Database

Stakeholders and Interests: Other Users - they will want to know the project they will be participating in

Pre- Conditions: Must be logged in, a company must be made prior and a project must already exist

Post Conditions: A user is assigned to the project

Trigger: The user wants to assign a user to a project

Main Success Scenario

1. User selects an option to assign a user to a project
2. The system displays users within the company
3. User picks a user
4. The system assigns the user to the project
5. The system saves the information in the database

Variation: None

Use Case: Change ticket state

Id: UC-- 009

Description: The use case will allow the user to change the ticket state

Primary Actor: User

Supporting Actors: Database

Stakeholders and Interests: None

Pre-- Conditions: Must be logged in, a company must be made prior, a project and ticket must already exist

Post Conditions: A ticket state is changed

Trigger: The user wants to change the ticket state

Main Success Scenario

1. User selects an option to change ticket state
2. The system displays all different states to change tickets to
3. User picks a ticket state
4. The system updates the ticket state
5. The system displays the new ticket state

Variation: None

Use Case: Build a ticket board

Id: UC-- 010

Description: The use case will allow the user to build a board

Primary Actor: User

Supporting Actors: Database

Stakeholders and Interests: Users keeps track of progress and list tasks to be done clearly

Pre-- Conditions: Must be logged in, a company must be made prior and a project must already exist

Post Conditions: A ticket board is created

Trigger: The user wants to create a ticket board

Main Success Scenario

1. User selects an option to create a board within the project
2. The system creates and displays an empty board for the project

Variation: None

Use Case: Produce a report

Id: UC-- 011

Description: The use case will allow the user to produce a report that shows the user's productivity and development progress

Primary Actor: User

Supporting Actors: Database

Stakeholders and Interests: Companies - keeps track of individual user performance for any KPI's

Pre- Conditions: Must be logged in, a company must be made prior and a project must already exist

Post Conditions: A report is generated

Trigger: The user wants to create a report based on a given option

Main Success Scenario

1. User selects an option to create a report for the project
2. The system checks if the user is within the company
3. Once validated the system displays to the user, and the user can select different options to show different metrics on the system (user productivity and project development progress) Once the user selects the options the system will calculate and fetch the necessary data to display the information in the form of a report

Variation: None

Use Case: Create a Prediction

Id: UC-- 20

Description: The use case will allow the user to produce a report that will predict the number of projects that each company will have, and the number of people needed to build these projects for any future date, based on historic data.

Primary Actor: User

Supporting Actors: Database

Stakeholders and Interests: Companies - to have a clearer vision on future expansions and infrastructure to support the number of projects and people they will potentially have

Pre- Conditions: Must be logged in, a company must be made prior and a project must already exist.

The company must also have done a few projects in a given time frame

Post Conditions: A report is generated

Trigger: The user wants to create a prediction report based on a given option

Main Success Scenario

1. User selects an option to create a prediction report for the company
2. The system checks if the user is within the company
3. Once the user is validated, the system will fetch the necessary data
4. The system will then calculate using the algorithm provided
5. The system will generate a report based on its findings and display to the user

Variation: None

Use Case: Delete Company

Id: UC-- 012

Description: The use case will allow the user to delete a company

Primary Actor: User

Supporting Actors: Database

Stakeholders and Interests: None

Pre- Conditions: A company must already exist, User must be logged in

Post Conditions: A company is removed from the database

Trigger: The user wants to remove a company

Main Success Scenario

1. User selects the option to delete a company
2. The system displays the company and a confirmation
3. User selects confirmation and the system removes the company from the database

Variation: None

Use Case: Delete Project

Id: UC-- 013

Description: The use case will allow the user to delete a project in a company

Primary Actor: User

Supporting Actors: Database

Stakeholders and Interests: None

Pre- Conditions: User must be logged in, A company must already exist and at least one project is ongoing at the company and no tickets are within the project

Post Conditions: A project is deleted

Trigger: The user wants to remove a project

Main Success Scenario

1. User selects the option to delete a project
2. The system displays the selected project and a confirmation
3. User selects confirmation and removes the project from the company
4. System updates its database

Variation: Project can be saved but simply unassigned from the company as opposed to being deleted

CRC

Model

Class	Responsibilities	Collaboration
UserLogin	<ul style="list-style-type: none"> - Holds user login details after registration 	<ul style="list-style-type: none"> - login - register - <i>LoginPageView</i> - <i>RegisterPageView</i>
Users	<ul style="list-style-type: none"> - Holds user personal information such as the company they belong to and how many tasks a user has done 	<ul style="list-style-type: none"> - <i>CompanyProfileView</i> - <i>ReportGeneratorView</i> - register
Projects	<ul style="list-style-type: none"> - Holds project details such as progression, number of tickets in the project, project overview, project due date 	<ul style="list-style-type: none"> - <i>MainDashboardView</i> - <i>CompanyProfileView</i> - <i>ReportGeneratorView</i> - <i>PredictionGeneratorView</i> - createProject - createTickets
Tickets	<ul style="list-style-type: none"> - Holds ticket tasks and states such as task due date and whether its not started, in progress or completed 	<ul style="list-style-type: none"> - <i>BoardView</i> - <i>oneTicket</i> - createTickets - oneTicket
Company	<ul style="list-style-type: none"> - Holds company information such as number of projects currently ongoing, completed and number of people within 	<ul style="list-style-type: none"> - <i>CompanyProfileView</i> - <i>PredictionGeneratorView</i> - createCompany - createProject

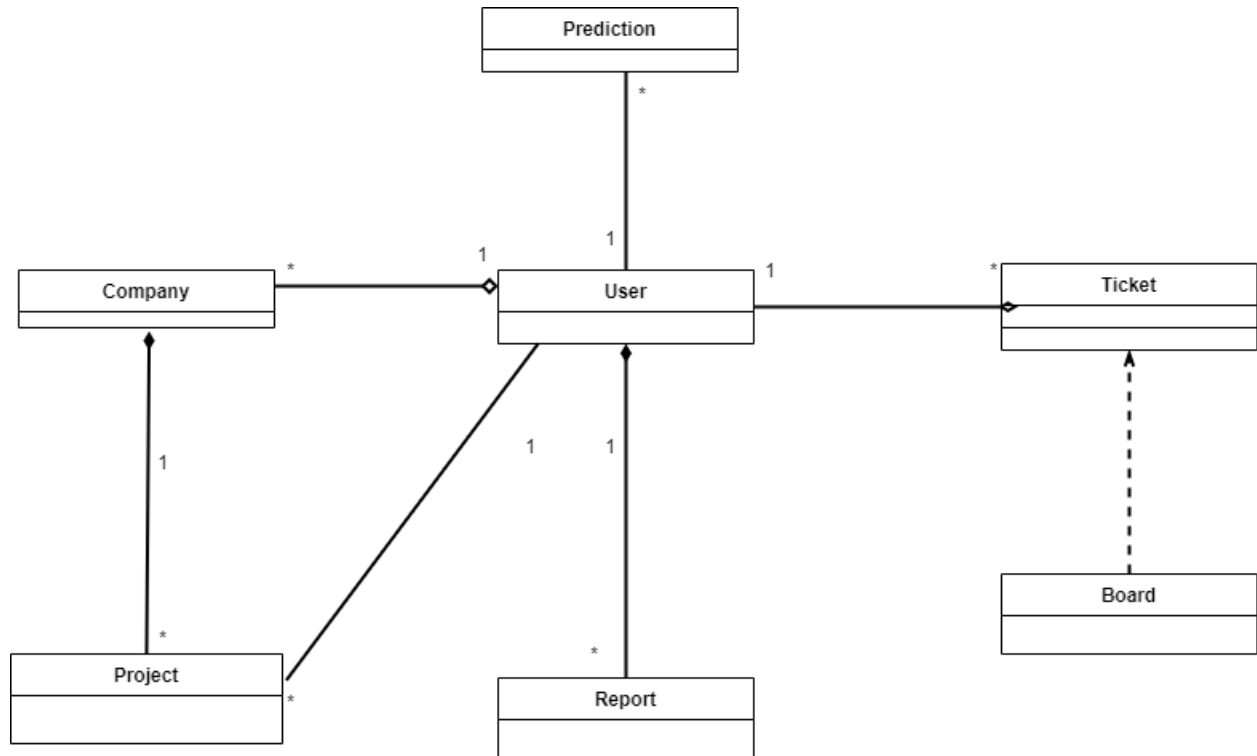
<i>View</i>		
Class	Responsibilities	Collaboration
<i>LoginPageView</i>	- View for entering user login details	- UserLogin - login
<i>RegisterPageView</i>	- View for registering user details	- register - UserLogin
<i>MainDashboardView</i>	- View for showing general information and a menu of options ie creating company or projects	- Projects - Company - createCompany - createProject - deleteCompany - deleteProject - assignProject
<i>CompanyProfileView</i>	- View for showing company details. projects summaries and predictions	- Company - Projects - Users - createProject - deleteProject - predictionProjects - predictionPeople
<i>ProjectView</i>	- The view that shows project details in full - Board creation - Reports - Current Projects - Users belonging to company	- Projects - Users - buildBoard - reportgenerateProductivity - reportgenerateProgress
<i>BoardView</i>	- View showing overall tickets and what states they are in	- Tickets - ticketState - assignTickets
<i>oneTicket</i>	- View showing a single ticket in full detail	- Tickets - ticketState - assignTickets
<i>ReportGeneratorView</i>	- View for showing the details of progress and/or productivity	- Project - User - reportGenerateProductivity - reportGenerateProgress
<i>PredictionGeneratorView</i>	- View for showing prediction details	- Project - Company

		<ul style="list-style-type: none"> - predictionProjects - predictionPeople
--	--	----------------------------------------------------------------------------------------------------

Controller		
Class	Responsibilities	Collaboration
login	<ul style="list-style-type: none"> - Takes user input details 	<ul style="list-style-type: none"> - UserLogin - User - <i>LoginPageView</i>
register	<ul style="list-style-type: none"> - Takes user input details 	<ul style="list-style-type: none"> - UserLogin - User - <i>RegisterPageView</i>
createCompany	<ul style="list-style-type: none"> - Input to create a new company 	<ul style="list-style-type: none"> - Company - <i>MainDashboardView</i>
createProject	<ul style="list-style-type: none"> - Input for creating a project 	<ul style="list-style-type: none"> - Projects - <i>MainDashboardView</i>
createTickets	<ul style="list-style-type: none"> - Input for ticket creation 	<ul style="list-style-type: none"> - Tickets
deleteCompany	<ul style="list-style-type: none"> - Input for company deletion 	<ul style="list-style-type: none"> - Company - <i>MainDashboardView</i>
deleteProject	<ul style="list-style-type: none"> - Input for project deletion 	<ul style="list-style-type: none"> - Projects - <i>MainDashboardView</i>
assignProjects	<ul style="list-style-type: none"> - Input for assigning projects 	<ul style="list-style-type: none"> - Projects - <i>ProjectView</i>
assignUsers	<ul style="list-style-type: none"> - Input for assigning users to projects 	<ul style="list-style-type: none"> - User - <i>ProjectView</i>
assignTickets	<ul style="list-style-type: none"> - Input to assign tickets to user 	<ul style="list-style-type: none"> - Tickets - <i>ProjectView</i> - <i>BoardView</i>
ticketState	<ul style="list-style-type: none"> - Change the ticket state at the user's requests 	<ul style="list-style-type: none"> - Tickets - <i>BoardView</i>
buildBoard	<ul style="list-style-type: none"> - Generates a board for a project 	<ul style="list-style-type: none"> - Tickets - Project - <i>ProjectView</i>
reportGenerateProductivity	<ul style="list-style-type: none"> - Generates a report for user 	<ul style="list-style-type: none"> - Company

	productivity	<ul style="list-style-type: none"> - User - <i>ProjectView</i> - <i>CompanyProfileView</i>
reportGenerateProgress	<ul style="list-style-type: none"> - Generates a report for project progress 	<ul style="list-style-type: none"> - Company - Projects - <i>ProjectView</i> - <i>CompanyProfileView</i>
predictionProjects	<ul style="list-style-type: none"> - Generates a prediction for number of projects 	<ul style="list-style-type: none"> - Company - <i>CompanyProfileView</i>
predictionPeople	<ul style="list-style-type: none"> - Creates a prediction for number of people needed for predictionProjects 	<ul style="list-style-type: none"> - Company - <i>CompanyProfileView</i>

Domain Model



Activity

