Crs Soft: Cross Rational Solutions

Summer Internship Project

**Social Kickstarter Platform**

*Project short-name: PalRaiser*

Analysis Report

Sarp Ulaş Kaya

Supervisors: Aysu Çağışlar, Beste Özalp, Tuğçe Özer

Jul 2, 2021

**Contents**

[1 Introduction 3](#_Toc76298800)

[2 Proposed System 4](#_Toc76298801)

[2.1 Overview 4](#_Toc76298802)

[2.2 Functional Requirements 5](#_Toc76298803)

[2.2.1 Login and Register 5](#_Toc76298804)

[2.2.2 Projects 6](#_Toc76298805)

[2.2.3 Social Features 8](#_Toc76298806)

[2.3 Non-functional Requirements 9](#_Toc76298807)

[2.3.1 Performance Characteristics 9](#_Toc76298808)

[2.3.2 User Interface 9](#_Toc76298809)

[2.3.3 Error Handling and Extreme Conditions 10](#_Toc76298810)

[2.3.4 Quality Issues 10](#_Toc76298811)

[2.3.5 Security Issues 10](#_Toc76298812)

[2.3.6 Supportability 11](#_Toc76298813)

[2.4 Pseudo Requirements 11](#_Toc76298814)

[3 System Models 12](#_Toc76298815)

[3.1 Use Case Diagram 12](#_Toc76298818)

[3.2 MVC Class Diagram 13](#_Toc76298819)

[3.3 Entity-Relationship Diagram 14](#_Toc76298820)

[3.4 User Interface 15](#_Toc76298821)

Analysis Report

*Project short-name: PalRaiser*

# Introduction

PalRaiser is a crowdfunding website where users can publish their projects to receive financial support from other users of the website by providing information on what their project is and why it needs funding. Additionally, the publisher of the project can post updates about the project, and the users that view the project can also create discussion topics to ask their questions, state their opinions or give suggestions about the project. These features are functionally similar to those of popular crowdfunding platforms such as Kickstarter and GoFundMe. The ‘Raiser’ in the project’s name derives from the fundraiser campaigns for projects published on such sites.

In addition to crowdfunding, the platform can also be used for simple social interactions with other users of the site such as following their accounts to see and be notified of their activities, share their thoughts on the platform in terms of posts, rate other users’ posts and projects, and commenting under other users’ posts and discussion topics. These features are functionally similar to those of popular social media platforms such as Twitter and Instagram. The ‘Pal’ in the project’s name derives from these social features.

# Proposed System

## Overview

Each user is greeted by the login scree where they have to enter the e-mail address and password they chose while creating their accounts. If they do not have an account yet, they can go to the registration page to sign up where they are asked to choose an e-mail address to be linked to their account (has to be unique), a password, their credit card information which is to be used whenever they decide to fund a project on the site, and various trivial information such as their birthday and gender.

Once the user is logged in, they have the option to publish one of their projects on the site. To do so, they have to enter its name, summary, detailed explanation and its type, i.e., video game, social media platform, virtual assistant etc. Besides publishing a project of their own, a user also has the option to browse through other projects on the site. They can search for projects by their name, their type or their publisher’s name. Upon choosing a project, the user is redirected to the project’s page where they will be able to see its name, publisher, summary, how much has been raised for it so far, various fundraiser milestone goals with brief explanations of what they will mean for the project’s development, and an option to donate. Upon scrolling down, they can also see tabs for a detailed explanation of what the project is, the project’s update history, and a forum where users can have discussions about the project. Everything on these pages other than the forum entries can only be edited by the publisher of the project, while the forum entries can also be edited or deleted by those who have posted them.

Besides searching for projects, the users can also search for other users by their names and send follow requests to them. Each user’s information, followers, posts and projects are shown on their profile. All of the users are able to view them in full detail by clicking on them, but they can only be edited or deleted by the owner of the profile. The users can also create new posts or publish new projects from here.

Aside from being able to see the other users’ activities on their profiles separately, the user can also access an “Activity” tab to see the activities of every user they are following. In addition to their posts and projects, they can also get notified of whenever one of their projects hits a milestone or gets an update. Either from here or from the user profiles, the user also has to option to rate or comment on other user’s posts, and their comments can also be rated and replied to by other users. The user can rate projects as well, and they also have the ability to report a project. Since scams can unfortunately be commonplace in crowdfunding platforms, the users of PalRaiser have the option to report suspicious projects as cons or scams, but they also have to state their reasoning as to why they think the project is a scam upon doing so. If a project gets reported this way, any user will then be warned upon entering the page with a message stating that X number of users have reported the project as a scam. The user then has the option to view these reports and decide whether they still wish to support the project despite the risks.

## 2.2 Functional Requirements

### 2.2.1 Login and Register

* The system should hide the characters of the user’s password.
* The system should not allow more than one account to be linked with the same e-mail.
* The system should ask for the user to confirm their password upon registration, in other words, ask for them to type the same password in two separate fields and check if they are correct.
* The system should redirect the user to the website’s main page after they log in or register.

### 2.2.2 Projects

* The system should allow the user to publish a project by entering its project’s title, publisher, summary, detailed explanation and fundraiser goals.
* The system should allow the user to search for projects using multiple keys. For instance, the user should be able to search for projects by not only by their name or by their publisher solely, but also by using both at once.
* The system should allow the user to see the project’s title, publisher, summary, detailed explanation, fundraiser goals, ratings, reports and update log.
* The system should warn the user with a message if the project has been reported.
* The system should allow the user to rate the project.
* The system should not allow the user to give the same rating to a project more than twice. They should not be able to like a project twice. Same for dislikes.
* The system should allow the user to change their rating for a project, i.e., change a like to a dislike and vice versa.
* The system should allow the user to remove their rating for a project.
* The system should allow the user to report a project as scam and ask for why they are reporting it.
* The system should not allow the user to report the same project more than once.
* The system should allow the user to remove their report for a project.
* The system should allow the user to fund the project by entering how much they want to raise.
* The system should allow the user to fund a project more than once.
* The system should allow the user to create discussion topics under a project.
* The system should allow the user to create more than one discussion topic under a project.
* The system should allow the user to reply to discussion topics.
* The system should allow the user to reply to the same discussion topic more than once.
* The system should allow the user to edit or remove their discussion topics and their replies under discussion topics.
* The system should allow the user to change information about the project or remove it from the website if they are the publisher.
* After choosing to donate to a project and specifying the amount they wish to donate, the user should be informed by a message that they are about to make a transaction using the credit card they had registered to support the project and asked if they really wished to proceed. The card info may also be shown to remind them but with appropriate secrecy, i.e., only the last 4 digits of their card’s number being shown.

### 2.2.3 Social Features

* The system should allow the users to search for other users by their names.
* The system should allow the user to send a follow request to another user.
* The system should allow the user to accept a follow request from another user.
* The system should not allow the user to follow another user more than once.
* The system should allow the user to see another user’s basic (name, account creation date) and trivial (birthday, gender) information, projects and posts on their profile.
* The system should allow the user to edit their own information, publish a new project or delete a project they have published, and create a post or delete/edit one they have created via their profile.
* The system should allow the user to use another user’s profile to rate and comment on their posts or go to the page of one of their projects.
* The system should allow the user to comment on a user’s post more than once.
* The system should allow the user to edit or remove their comments.
* The system should allow the user to rate other users’ comments to posts.
* The system should not allow the user to give the same rating to a post or a comment more than once.
* The system should allow the user to change or remove their rating on a post or a comment.
* The system should allow the user to see the activity of the users they are following on the Activity page.
* The system should allow the user to unfollow other users.

## 2.3 Non-functional Requirements

### 2.3.1 Performance Characteristics

* Response time of the system should not exceed 1 second.
* There shouldn’t be any data loss between the server-client communications.
* The pages should not take multiple seconds to load as long as the user has an ideal and stable internet connection.

### 2.3.2 User Interface

* Everything in the user interface should be easily readable. Grammar errors should be avoided and contrasting colors should be used for texts or icons and their backgrounds.
* There shouldn’t be any non-functional elements on the user interface that may falsely give the impression that they are functional pieces, i.e., there shouldn’t be any solely decorative elements that look like buttons or text fields.
* The aesthetics of the website should not be neglected as it may turn off potential users from the website, but should also not be overdone. It should be clean and polished but not distracting.

### 2.3.3 Error Handling and Extreme Conditions

* The System should display appropriate error messages whenever the user doesn’t follow what they are supposed to do. For instance, an error message should tell them that there is already an account linked with the e-mail they are trying to register with if they enter an e-mail that is linked to another account while registering.
* After displaying an error message, the system should reload the page or keep it as is appropriately. A page or a functionality should not fail to work or load properly when the user doesn’t follow their intended design.

### 2.3.4 Quality Issues

* There shouldn’t be any downtime in the website’s services.
* There should be appropriate character limits for the text fields to be filled.
* Form fields should be complementary to the type of information that is asked, i.e., if an integer value is asked, then only numerical characters should be available for the user to enter.
* Mandatory and optional fields to fill out in forms should be managed appropriately.

### 2.3.5 Security Issues

* A user shouldn’t be able to see the activity of a user they’re not following on the Activity page.
* A user shouldn’t be able to change any information regarding a project published or a post created by another user.
* A user shouldn’t be able to change another user’s account information.
* A user shouldn’t be able to see another user’s password or credit card information.

### 2.3.6 Supportability

* The website should not be supported exclusively by a specific internet browser or a group of internet browsers. It should be available for any internet browser to support.
* The website’s availability to a user should not depend on the local software or hardware of their device as long as it is a device that is able to browse the internet.

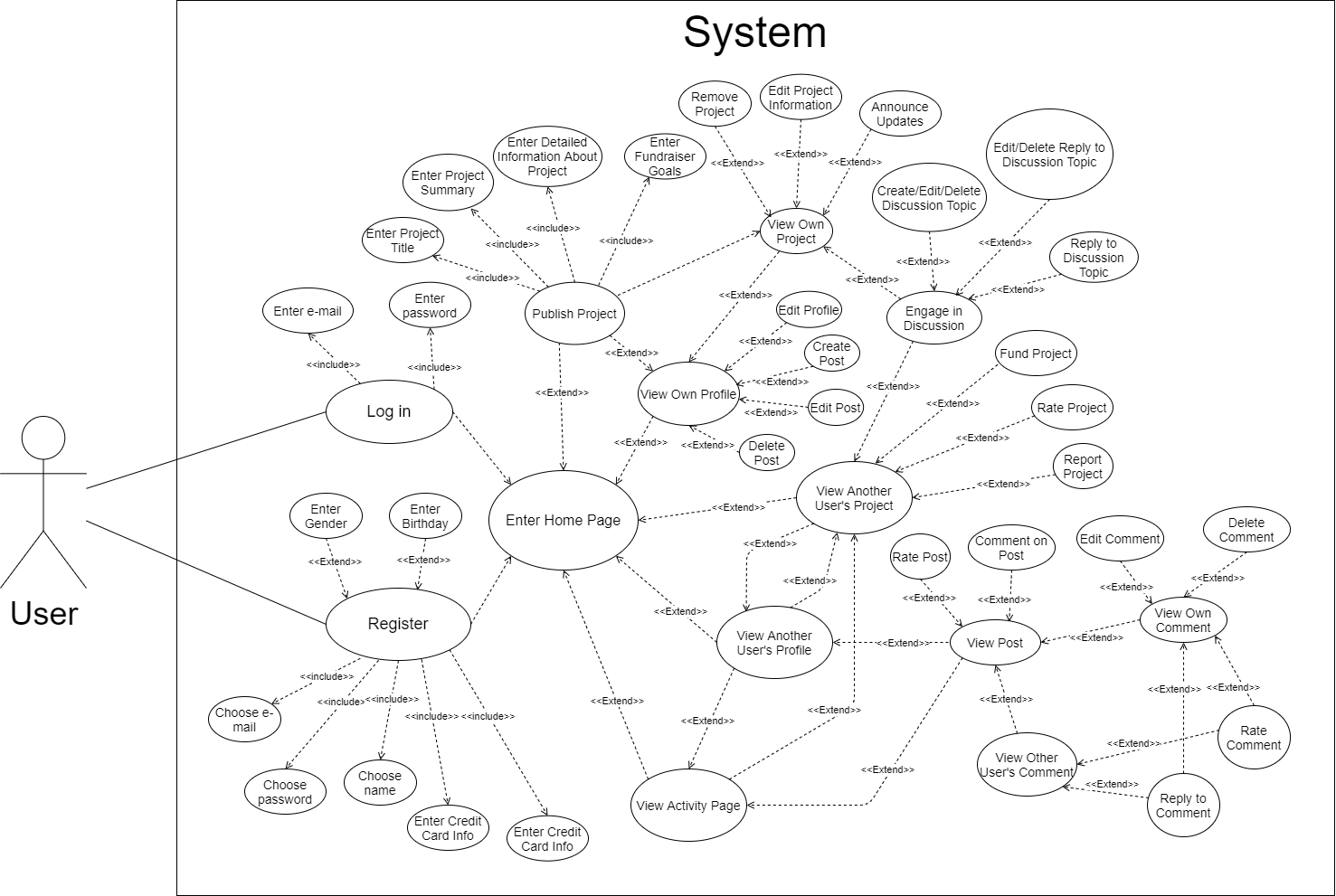
## Pseudo Requirements

* The project will be coded in C#
* The ASP.NET Core 3.1 open-source web framework will be used to design the website
* The database will be set up using Microsoft SQL Server
* GitHub will be used for version control during development.

# System Models

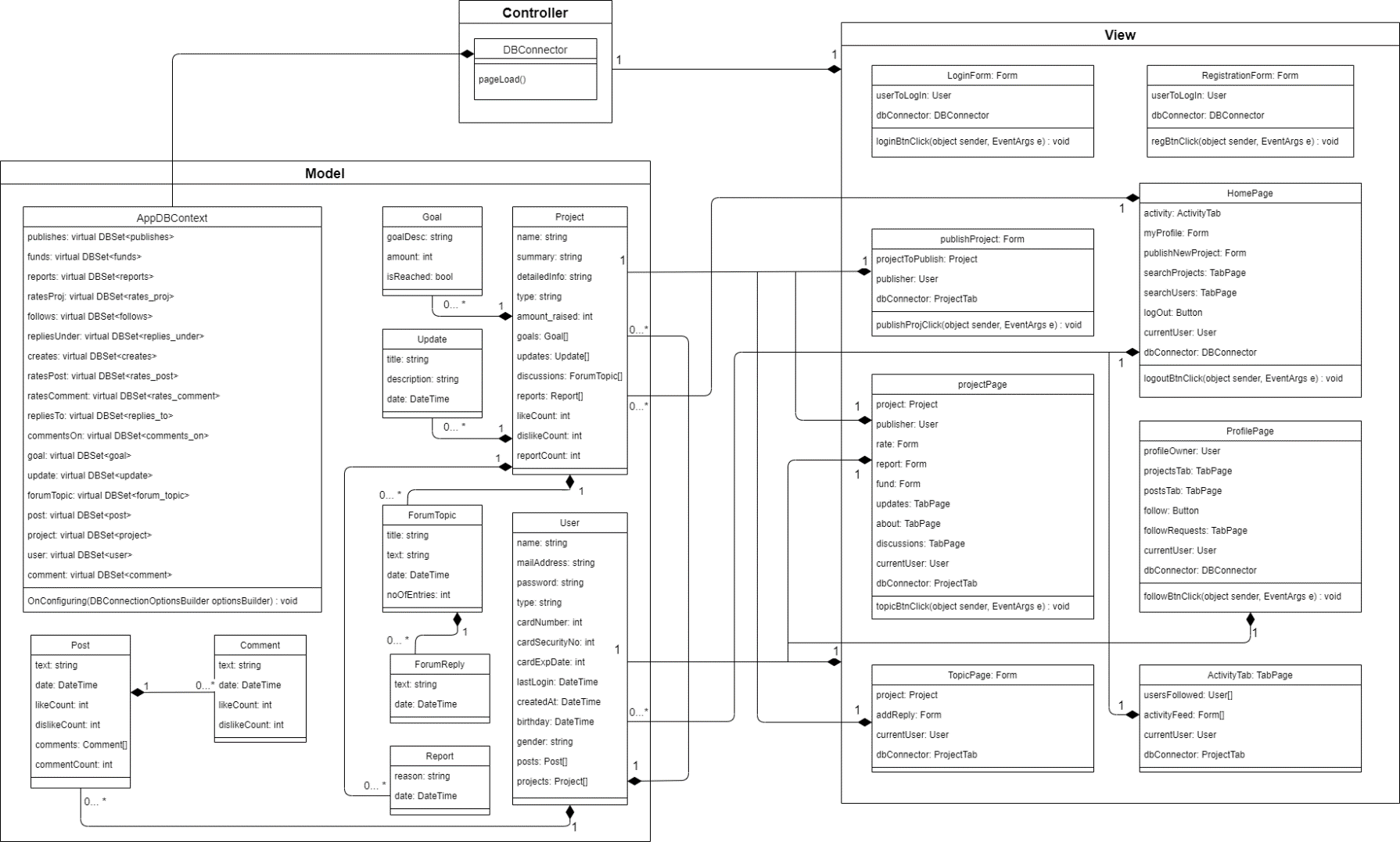


## Use Case Diagram



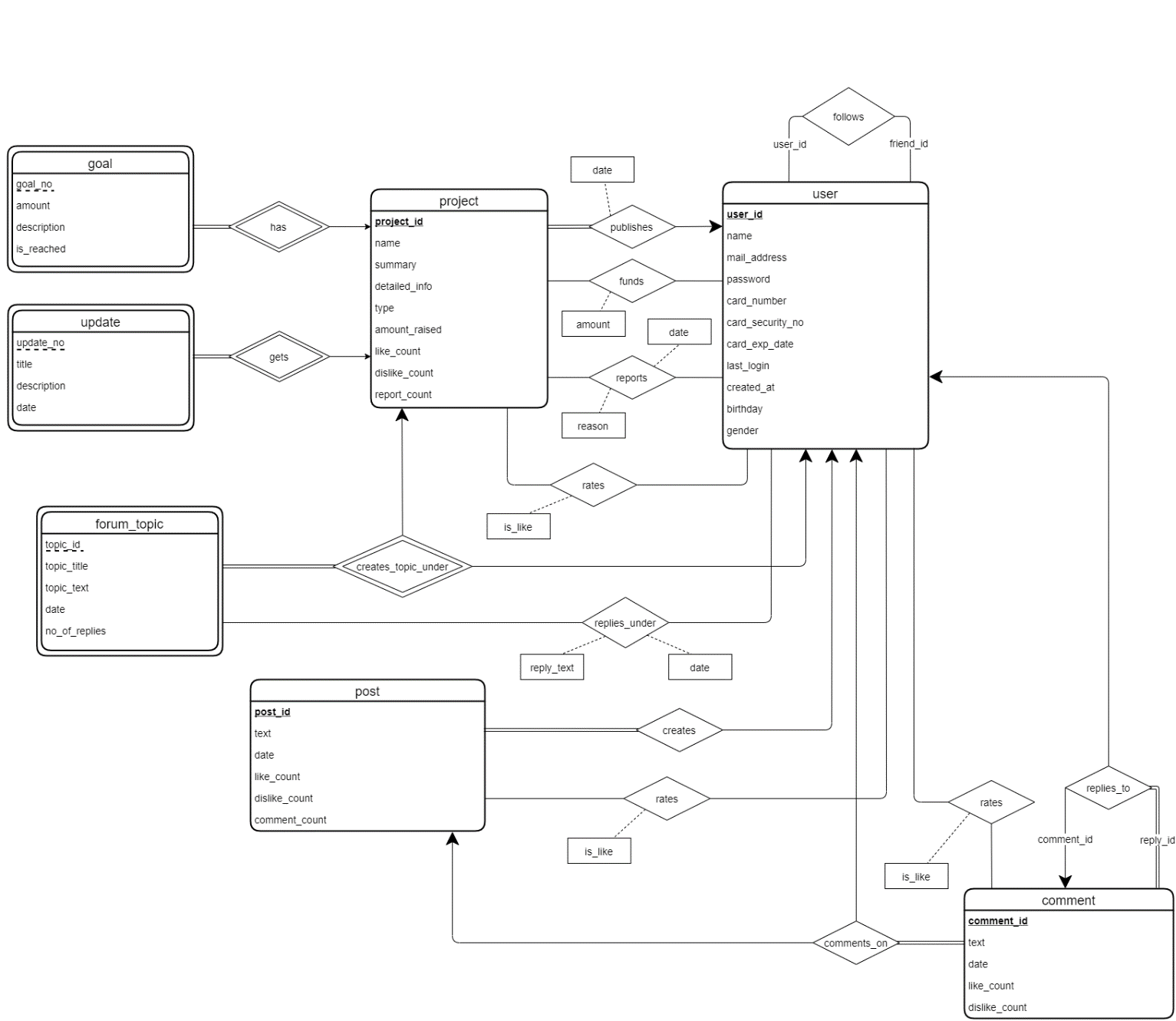
*Figure 1: Use-Case Diagram*

## MVC Class Diagram



*Figure 2: Class Diagram*

## Entity-Relationship Diagram

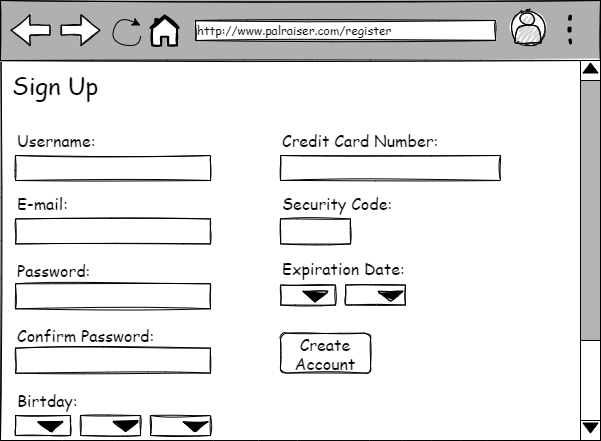


*Figure 3: Entity-Relationship Diagram*

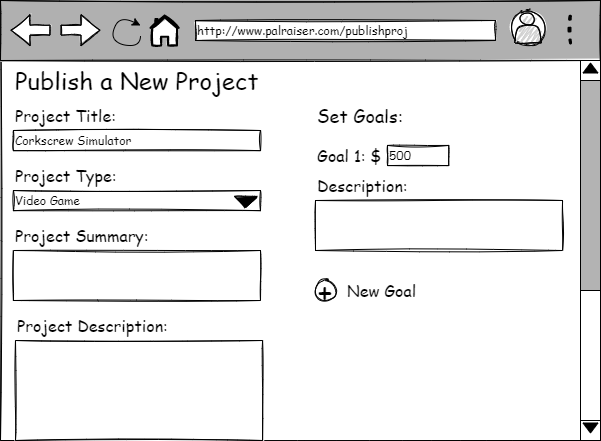
## User Interface



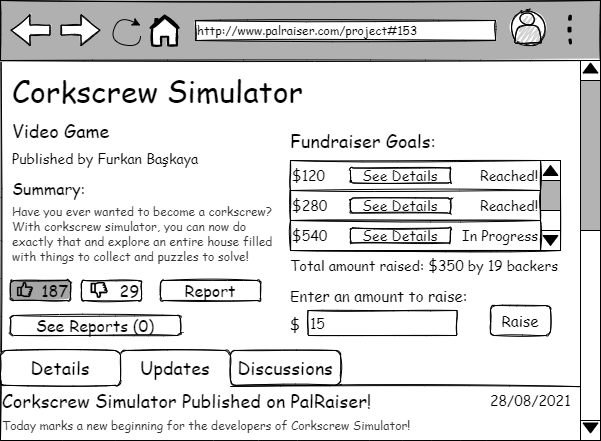
*Figure 4: Login Screen*



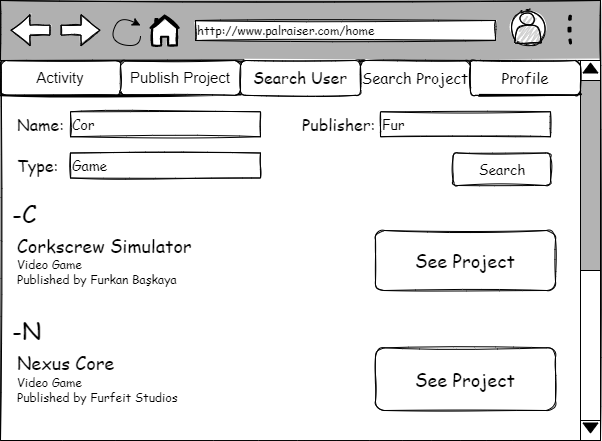
*Figure 5: Registration Screen*



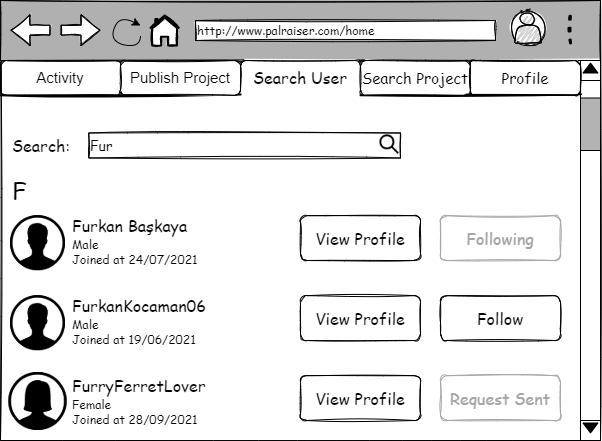
*Figure 6: Project Publishing Screen*



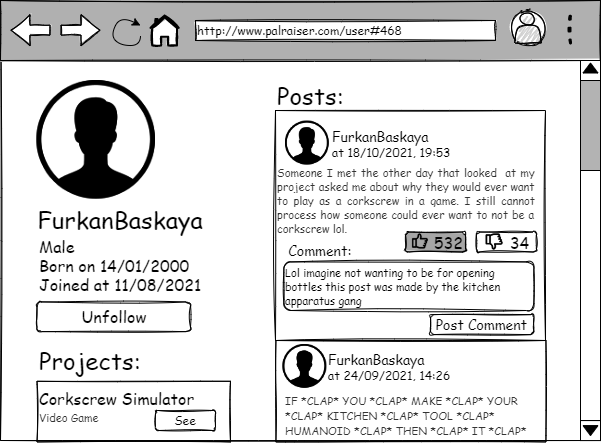
*Figure 7: Project Page*



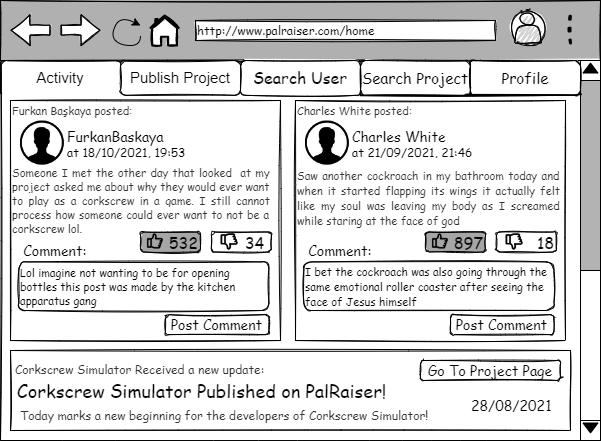
*Figure 8: Project Search Tab*



*Figure 9: User Search Tab*



*Figure 10: User Profile*



*Figure 11: Activity Tab*