Virtual reward system

# Introduction

The goal is to create simple easy extendable online system which will provide the users with the ability to send virtual gifts or rewards to their co-workers. Gift, badges or simple line of text with gratitude will be given to colleague based on their involvement and skills.

# Gifts

Gift can be created by the user who wants to send them. List of predefined images will be available to the user to choose from. All users can create and send gifts.

## **Example**

Name of the gift (from the database table Badges/Gifts)



Thank you for helping me this SQL query! (Reason from the database table User\_Badges)

Tags: Tag1, Tag2, Tag3, Tag4 (Table Badges\_Gifts\_Tags)

# Badges

Badges will be simple set of predefined data (icon, text, explanation). Only administrators and managers will able to submit new badges in the system. Badges should be skill related!

## Earning badge

Badges might be given from team lead or co-worker who is confident that the user has such skills.

Badges might be given to users when they complete a task not related to his/hers usual activities.

## **Example**

Name of the badge (from the database table Badges/Gifts)



CSS ninja (Reason from the database table User\_Badges)

Tags: Tag1, Tag2, Tag3, Tag4 (Table Badges\_Gifts\_Tags)

# Goals

When looking at earned badges and gifts by users, colleague or team lead can easily determine which programmer has certain skills and ask him/her for assistance.

# Requirements

## Use KODAR CRM database to obtain information about users.

1. Make accounts with the available information (name, username, email, etc.) from the CRM.

## UI build with Twitter Bootstrap.

1. Use Twitter Bootstrap to add style to the application- http://getbootstrap.com/

## Responsive design. – Use Twitter Bootstrap

.



## Use free for commercial use icons.

Look at the following links.

1. <http://www.iconarchive.com/commercialfree.html>
2. <http://www.softicons.com/commercial-icons>
3. <https://www.iconfinder.com/free_icons>

## Send email to users when they receive badge or gift card.

If an employee gets a badge, send an email which informs them about the badge. Get the email from the CRM system, use the name of the employee from the CRM too.

## Build with web services (using ASP.NET or JAVA)

# Database

## TABLE USERS

|  |  |  |
| --- | --- | --- |
| **NAME** | **TYPE** | **DESCIPTION** |
| ID | int | Primary key/Mandatory |
| USERNAME | nvarchar(30) | Username for login in the system, get it from KODAR CRM |
| PASSWORD | nvarchar(30) | Password for login in the system, Encryption |
| NAME | nvarchar(50) | Name of the user, get it from the KODAR CRM. |
| EMAIL | nvarchar (50) | User email, get it from the KODAR CRM. |
| USER\_ROLE | int | User roles. |

The system must support 4 roles (Administrator (can add gifts and badges from administration panel), Manager (can give badges/gifts to programmers, testers, etc.), Programmer (can give/upload gifts to colleagues), Tester (can give/upload gifts to colleagues).

The fields “USERNAME”, “NAME”, “EMAIL” must be taken from the KCRM database. With every user login, these fields must be refreshed.

## TABLE BADGES\_GIFTS

|  |  |  |
| --- | --- | --- |
| **NAME** | **TYPE** | **DESCIPTION** |
| ID | int | Primary key/Mandatory |
| NAME | nvarchar(30) | Name of the gift |
| FLAG\_TYPE | bit | If it is 0 type is badge, if it is 1 type is gift. |
| IMAGE\_ID | int | foreign key (Images) |
| USER\_ID | int | foreign key(users), this field must show which user has uploaded the (badge/gift) |

The FLAG\_TYPE field can be 0 or 1. If it is 0 the type is badge, if it is 1 the type is gift. This field depends on which radio button is clicked in the user interface

## TABLE TAGS (skills that can be attached to a badge/gift)

|  |  |  |
| --- | --- | --- |
| **NAME** | **TYPE** | **DESCIPTION** |
| ID | int | Primary key/Mandatory |
| NAME | nvarchar(50) | Tag |

## TABLE BADGE\_TAGS

|  |  |  |
| --- | --- | --- |
| **NAME** | **TYPE** | **DESCRIPTION** |
| ID | int | primary key/Mandatory |
| TAG\_ID | int | Fkey/TAGS/ Mandatory |
| BADGE\_GIFT\_ID | int | Fkey/BADGES\_GIFTS/ Mandatory |

Every badge/gift can have multiple tags attached to it.

* Badge for good backend skills – SQL and ASP tags;
* Badge for good frontend skills – HTML and CSS tags.

## TABLE IMAGES

|  |  |  |
| --- | --- | --- |
| **NAME** | **TYPE** | **DESCIPTION** |
| ID | int | Primary key/Mandatory |
| NAME | nvarchar(255) | Name of the, Image |
| IMG | varbinary(MAX) | Badge icon, image |

## TABLE USER\_BADGES

|  |  |  |
| --- | --- | --- |
| **NAME** | **TYPE** | **DESCIPTION** |
| ID | int | Primary key/ mandatory |
| SENDER\_ID | int | foreign key (Users), represents who gives the badge/gift |
| RECIPIENT\_ID | int | foreign key (Users), represents who gets the badge/gift |
| BADGE\_GIFT\_ID | int | foreign key(BADGES\_GIFTS) |
| GIVE\_DATE | datetime2 | When a certain user has gotten a badge |
| REASON | nvarchar(2000) | Reason for giving a badge, plain text. |

The SENDER\_ID field contains the ID of the user who gives the Badge/Gift.

The RECIPIENT\_ID field contains the ID of the user who gets the Badge/Gift.

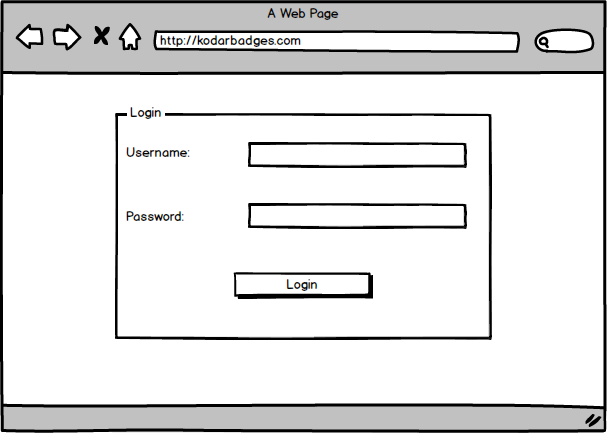
The GIVE\_DATE field will contain the date when a certain badge/gift has been given.

The REASON field will contain plain text with the reason for giving a badge/gift.

# Screens and functionalities

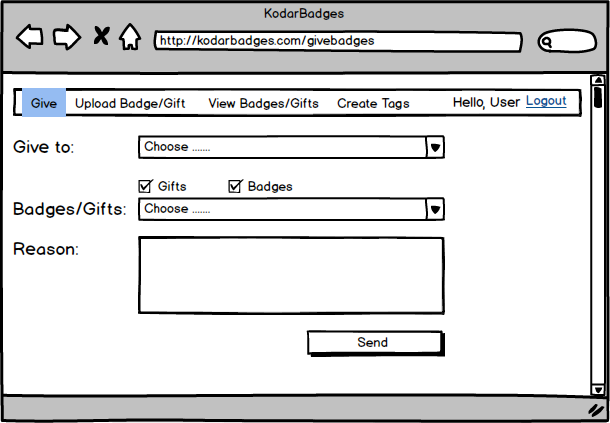
## Login screen

To enter the system, a user must first login via username and password. When the user clicks on the “Login” system, if the username and the password match, he will be allowed in the system.



## Main Screen

### Manager View



When the manager logs in the system, a screen with the main functionality of the application appears.

At first, a recipient for the badge/gift must be chosen from the dropdown “Give to:”. It will contain the names of all employees (taken from the KODAR CRM).

After a recipient is chosen, the sender must choose, badge or gift from the dropdown. The information can be filtered through the checkboxes “Gifts” and “Badges”.

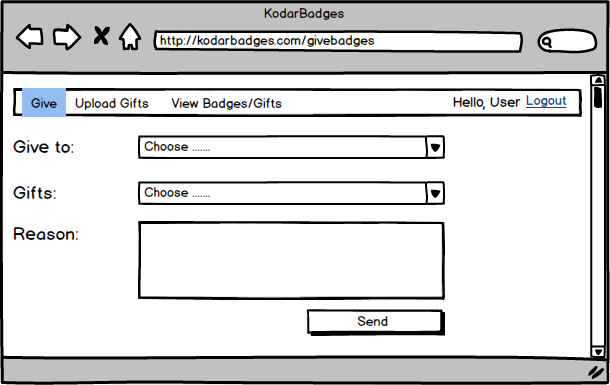
In the textfield “Reason” the user can enter a reason for giving a certain badge/gift.

Mandatory fields: “Give to”, “Badges/Gifts”, “Reason”.

When the button “Send” is clicked, the data is saved in the database, and an email is sent to the person who gets the badge.

### Employee View

The logic is the same as the “Manager View”, but the employees can send only gifts



### Email structure

The data which comes from the database is displayed as follows: ***<data that comes from the database>***

The structure of the email must be as follows:

**From:** [*noreply@mail.com*](mailto:noreply@mail.com) *(hardcoded)*

**To:** ***<***[***employee@kodar.net***](mailto:employee@kodar.net)***> (Email from the Users table)***

**Theme**: Badges (hardcoded)

**Text:**

*Congratulations* ***<Nerd Nerdinson> (Name form the Users table),*** *you have earned a* ***<CSS Badge/ A gift> (Name from the Badges/Gifts table).***

***< Image of the badge/gift, from the database>***

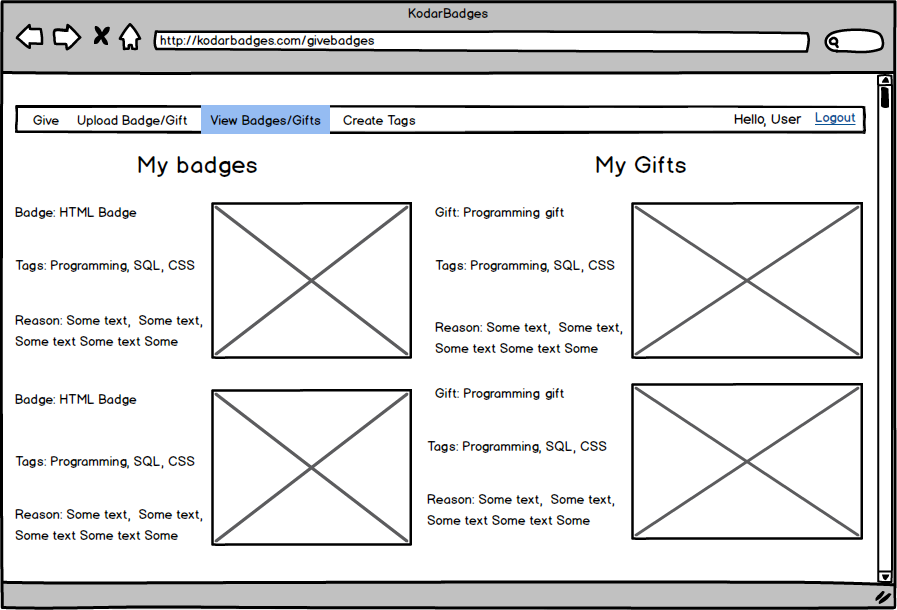
***<Tags>,<Tags>,<Tags>.***

***< Continue with the good work.> (Reason from the textfield)***

*You got the badge from* ***<Boss Bossinson (Name form the Users table >***

## View Screen

In the “View” screen, the user will be able to see all his badges and gifts.



Badge/Gift: ***<name of the badge/gift>***

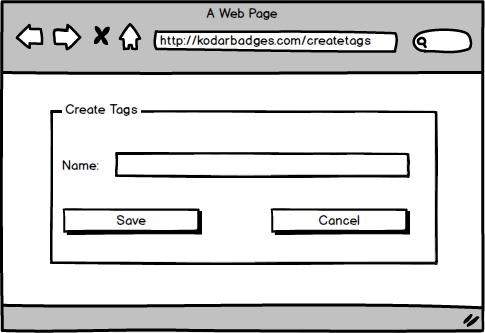
Tags : ***<tags>***

Reason: ***<reason for the badge>***

Image: ***<image of the badge/gift>***

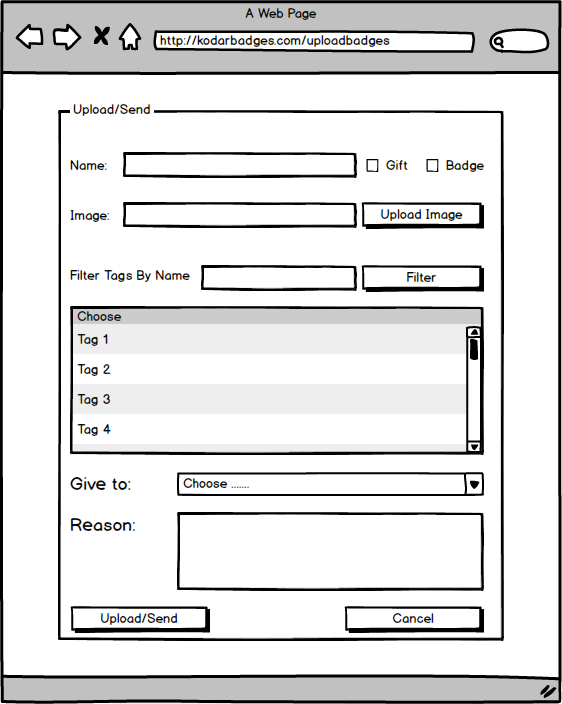
## Create Tags (Manager Account)

Managers can create tags. To create a tag, a name must be specified and then the “Save” button must be clicked.



## Upload Screen – Manager View

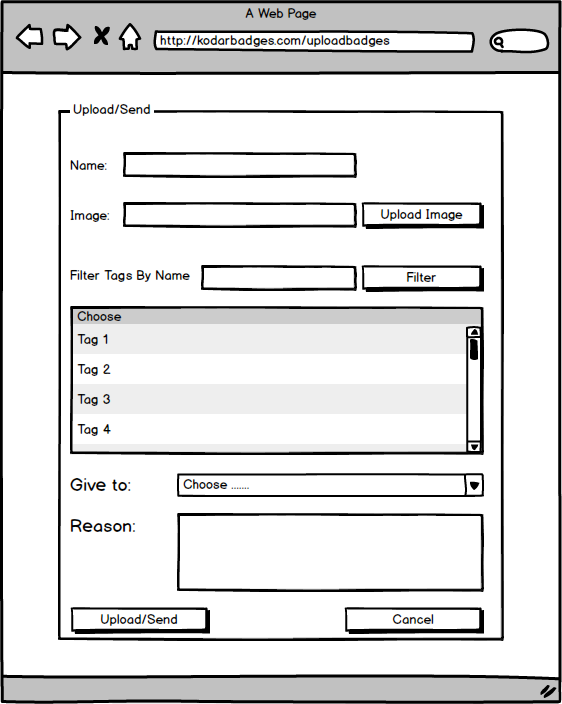
## Managers can upload badges and gifts in the system by clicking the “Upload” element from the menu. The badges and the gifts, which are uploaded by the administrator, will be visible for all users. The badges and the gifts which are uploaded by the other users will be visible only for them.



The user must enter a name of the badge/gift, then choose if he is creating a badge or a gift from the checkboxes and then he must upload an image (the badge, the gift).Tags can be chosen from a table. When the “Upload” button is clicked, the information about the badge/gift is saved in the database. The tags can be filtered by tag name. The will be a functionality which allows the manager to send the badge/gift, as he uploads it. It is not mandatory (if the “Give to” and the “Reason” fields are not filled in the badge/gift is only saved in the database).

**Upload Screen – Employee View**

The logic is the same as the “Manager View” but the employee can upload/send only gifts.



# Similar projects

<http://www.openbadges.org/>