Adding new features to a physical exercise web application

Utveckling av nya funktioner till en existerande träningsapplikation

Alan Altonchi
Emil Hising Bertils

Department of Mathematics and Computer Science

Computer Science

C-dissertation 15 hp

Supervisor: Stefan Lindskog

Examiner: Karl-Johan Grinnemo

Date: 2022-06-10

Acknowledgement

Thanks to our supervisors from CGI, Magnus Norman and Liam Rasim, for giving us the opportunity to do our thesis with CGI. Also thanks to our essay supervisor, Stefan Lindskog, for helping us and giving feedback throughout the process of writing the thesis.

Abstract

The objective of this Bachelor thesis project is to improve an existing CGI application referred to as CGIMoving. The aim of CGIMoving is to encourage CGI employees to engage in physical activity. Before this project, anyone on Internet could register for the application, regardless of their affiliation. The primary purpose of this project was to ensure that invitations from administrators are the only means of registration. This was accomplished by removing the registration button from the main page and making sure that the only way to register was through an invitation sent in an email to the user from an administrator. When the application was successfully converted to a so-called invite-only application, further improvements were accomplished. The first additional improvement involved enhancing the visual appeal of the administration page, by ensuring that all colors matched and no text overflowed over button borders. The second improvement was a request made by CGI to notify people through email when a campaign was started. The final improvement consisted of recreating the users list on the administration page so it could be sorted more easily and display whether a user is blocked or not, as well as correcting the broken blocking functionality. All proposed and implemented improvements have been validated and approved by the product owner.

Sammanfattning

Målet med detta examensarbete är att förbättra en redan befindlig CGI applikation som kallas för CGIMoving. Målet med CGIMoving är att uppmuntra CGI arbetare att engagera sig mer i fysiska aktiviteter. Innan vår uppdatering av applikationen kunde vem som helst på Internet registrera sig oavsett om de jobbade för CGI eller inte. Huvudmålet med detta projekt var att se till att det enda sättet att registrera sig var med en inbjudan från en administratör. Lösningen till detta problem var att ta bort "Register Now" knappen från hemsidan, och se till att det enda sättet att registrera sig är att få en inbjudan via epost från en administratör. När applikationen hade gjorts om till en så kallad "invite-only"-applikation genomfördes ytterligare förbättringar. Den första förbättringen handlade om att förbättra visualiseringen av administratörssidan genom att se till att alla färger matchade resten av hemsidan och att text fick plats inom ramarna för knapparna. Den andra förbättringen var en önskan från CGI, som handlade om att registrerade medlemmar skulle bli meddelade via epost när en ny kampanj började. Den sista förbättringen handlade om att göra om användarlistan på administratörssidan så att det blev enklare att sortera den och se till att den visar om användarna är blockerade, samt att rätta till den trasiga blockeringsfunktionen. Alla föreslagna och implementerade förbättringar har validerats och godkänts av produktägaren.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

Maintaining daily physical exercise is important, since it decreases the risk for chronic conditions, such as Accelerated biological aging, type 2 diabetes, coronary heart disease and stroke [1]. Regular physical exercise can also help soften the symptoms of the ongoing pandemic of Covid-19 [2].

A good way to motivate people to exercise regularly is to let them log their training. This lets them see how they are doing and makes it easier to track progress. There are a lot of apps that let their users do this, there are even apps where they instruct the user how to exercise and build personal schedules for them.

The company CGI encourages their employees to exercise because of the benefits it provides. Since not everyone considers the benefits of exercising sufficient, CGI wants to create extra incentive for their employees. The way they have decided to do this is running regular exercise campaigns where the employees can win prizes. To do this CGI needs some way for their employees to log their physical activities. Their solution was to create an in-house web application, called CGIMoving, where the employees can choose between some predefined activities. They then get some lottery tickets when the campaign ends based on the amount of time they have logged. An administrator (admin) then spins a wheel to see which employees wins prizes.

CGI choose to develop their own application for their employees to log their physical exercise. They had many reasons for developing it in-house. The major reason was that they did not find any other applications where they could easily allow their users to register any activity and have the same weight as all the other activities. They wanted to have the same weight on standing to work for 30 minutes as running for the same amount of time. Another reason was that they did not want to pay for the service.

At the start of the project they had a much smaller scope than they have today, they only wanted the users to be able to register activities. When CGI wanted to expand the scope they realised that the students at Karlstad University are capable of solving the implementation of this expansion. The expansion has made it easier for CGI to let students work for them as part the students Bachelor thesis, which they used to have problems with. The problem is that most of the projects that CGI conduct are for customers, which means that the customer must approve that a student is working on that project and most of their customers does not allow this.

This thesis will outline and discuss our work to improve CGIMoving based on the requirements specified and provided by CGI.

1.1 Background

Our Bachelor project is to continue the development of CGIMoving by implementing the new requirement specified by CGI. The application was originally developed in-house by CGI and has since also been the project of two Bachelor theses at Karlstad University, *Improvement of a physical exercise logger* [3] and *Expansion of a physical exercise logger* [4].

In [3], the students worked according to the Scrum methodology, detailed in [5], and managed to implement one feature more than they had planned originally. They implemented the following six features:

1. Create predefined activities for a campaign
2. Implementation of reminders for inactivity
3. Implementation of reminders for unclaimed rewards
4. Edit user registered activities
5. Edit information about users
6. Create events

In [4], the students implemented a new feature where groups can be created and applied to users, dividing them into groups or subgroups. This group had a meeting every week with CGI and managed for the most part to complete their goals.

1.2 Problem

A severe shortcoming in the existing web application is that anyone connected to Internet may create an account and begin to log physical exercises. This provides manual work for the admins who have to keep banning accounts outside of CGI. Our first task is therefore to make sure only invited CGI employees are able to register in the application. Hence, the current problem is that there is currently no way for employees to be invited to the application and that uninvited people have access to the registration for an account.

Our plan to solve these problems is to create a feature for admins to invite people by their email address and check whether the person was invited when they try to register an account. We also plan to make the register page only accessible by invited people, by redirecting uninvited people to the login page if they somehow mange to get to the register page by for example guessing the Uniform Resource Locator (URL) correctly.

Hence, in this thesis we primarily aim to answer the question “How can we make sure only invited people have access to register to the web application”.

1.3 Objective and Goals

The main objective of the work presented in this Bachelor thesis is to design and implement new features for the web application CGIMoving for CGI. CGI decides which features are important for the application and these are prioritised as primary and secondary goals.

The primary goal of the work is to restrict access of CGIMoving for unwanted users. Our plan to accomplish this is by transforming the application into an invite-only platform, as opposed to being open for registration to everyone who can find the app online. To add to this, a feature to view outstanding invitations and revoke them was added upon suggestion by the thesis author.

In addition to the primary goal there is also a three secondary goals, which are prioritised accordingly:

1. Visual improvements
2. Email notification when a new campaign is starting
3. Administer blocked users

Secondary goal 1 is to update the application visually, change buttons so that the text does not go outside of the button for example. There are a lot of places where problems like this exists, mainly on the admin parts of the application though.

Secondary goal 2 is to make the application send out an email to every registered user that a new campaign is starting when an admin creates a campaign.

Secondary goal 3 is to crate a view where admins can see all the blocked users. There will also be a button to pardon a user.

1.4 Ethics

This project does not have any ethical issues since we only change how the application is used, not the intent or purpose of the application. One could argue that logging peoples physical exercise could be an ethical issue, that is not our responsibility though. It is up to the product owner, i.e., CGI in Karlstad, how they choose to use their product. Someone might feel bad about themselves if they are compared to others who are more fit than they are. Although, considering that the application is not a competition, this does not seem that bad. Because of how important physical exercise is for people the benefits of encouraging people to train probably outweigh the risk of it becoming a unwanted competition.

1.5 Method

To implement the proposed improvements, an informal Agile development process [6] has been used. We have used sprints, sprint planning, sprint review, product backlog, and sprint backlog. The sprints have been two weeks long and start with a sprint planning meeting and end with a sprint review meeting. Sprint reviews and sprint planning for the next sprint have been held at the same meeting with the product owner.

The product backlog has been constructed in a discussion with the product owner where we provide a list of improvements to the application, based on both our own opinions and on the backlog CGI had prior to this project, and CGI decides which ones they want and their priority. In the sprint planning meeting we choose which items from the product backlog to include in the sprint backlog, based on what we think we can handle and the priority of the item. During the sprint we try to complete these items and they are then reviewed by CGI when the sprint ends in the sprint review meeting. The implemented features are tested manually during development. If all the items in the sprint are finished before the sprint ends, we choose additional items from the product backlog and start implementing them.

1.6 Stakeholders

The client and project owner is CGI Karlstad, they own the web application and are also the intended users of the application. CGI is an IT consultant company founded in 1976 with offices in multiple countries and have approximately 82 000 employees globally. Their head office is located in Montréal, Canada. In Sweden, CGI has in total 30 offices. The project owner of CGIMoving is located in Karlstad, Sweden.

CGI benefit from the applications improvement since they want to encourage their employees to maintain good physical health, and improving the application will either help to achieve this or make the administrative work easier. They benefit specifically from the upgrade to an invitation based application because their admins of the web application will not have to manually search for and ban unwanted people registering to their application.

1.7 Distribution of Work

Throughout the entirety of the project, the division of work has been nearly equal. Both authors consistently engaged in separate activities. When one author completed their assignment, the other would evaluate it and they would then work together to complete it before going on to their next goals. While working on the actual project and coding, both authors had different responsibilities and worked on various sections of the code before submitting it for review. Each student would then be responsible for writing about their own solutions and code in the thesis, as it would be simpler to describe their own work. Every time a chapter was finished, both authors would evaluate what they had written on a conference call to ensure that everything was complete. There are parts of the thesis that both authors worked on writing together, such as the introduction chapter and the background chapter. Then there are certain chapters that were written by whoever was more in touch with the work that was done and had a clear understanding of the solution presented.

Some things Emil worked on are: the emailing back-end and ensured that emails are sent when users are invited to register on the application. He also worked on the formatting of the emails text-box to ensure that it accepts the email list format provided by CGI, notification emails when campaigns are starting, fixing the blocking users functionality, and a pop-up message when invitations are sent and revoked with an undo button if an invitation was accidentally revoked.

Some things Alan worked on are: the hashing algorithm to generate hashes for the invited users, the browser checking if the hash is valid and redirecting the user to the login or registration page, creating an invitations-sent list on the right-hand side of the invite page that displays a list of all invited emails, the visual update for the admin page to ensure that the text fits properly and the correct colors are used for the buttons, and the conversion of the users list to a better list that can be sorted and displays blocked users.

1.8 Delimitations

One of the delimitation's of this project is that we will not account for the risk that the email account of an invited user has been hacked or that the email is caught by a malicious person via spyware or other malicious means. We will also assume that the invited user is not malicious and will not give away their invitation to someone else.

Another delimitation of the primary goal is that the email-sending Application Programming Interface (API) used in the application can currently only handle up to 100 emails per day. If CGI were to expand this application beyond their office in Karlstad or release it for public use they would need to be able to send far more than 100 emails per day. But since this is not a focus of CGI at the moment we will not account for this limitation.

1.9 Disposition

This thesis is structured as follows. In Chapter 2, it will outline the background of the project, describe what has been done before we started on the application. In Chapter 3, the design choices we made for the improvements are presented. In Chapter 4, the technical implementation of the improvements made to the application is described in detail. In Chapter 5, the results of the improvements made during the project is presented. Additionally, in Chapter 6, we discuss our conclusions from the project work as a whole, and discuss what went well, what did not go well, and give some examples of future work that can be done to the web application. Finally, in Appendix A, all abbreviations and acronyms used in this thesis are explained.

Chapter 2

Background

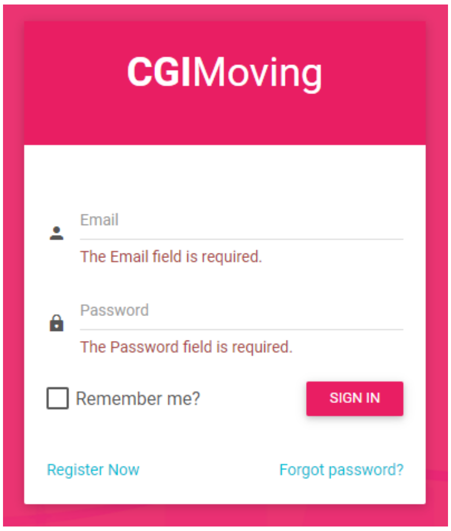
Section 2.1 describes the condition of the product before it was handed over to us for further development, as well as the results of prior students' work on it. Section 2.2 focuses on our area of interest and our key goal as well as why this work needs to be done. Section 2.3 showcases a few functions with similar properties to the primary objective of the project and how they work.

2.1 CGIMoving

The web application CGIMoving was an existing application before this Bachelor thesis started. The development of the application started as an in-house project at CGI and has later been extended in two Bachelor theses before ours. It is accessible on the Internet for everyone to register, but is only intended for CGI employees.

2.1.1 Login and Register

The first thing that the user is presented with when they navigate to the application is the login view, where they can choose to login with an existing account, go to the register view to register a new account or reset the password of their account. This view can be seen in Figure 2.1. If the user chooses to register a new account they are sent to the register view, which can be seen in Figure 2.2.



The Email field is required.

The Password field is required.

Figure 2.1: The login view of CGIMoving.

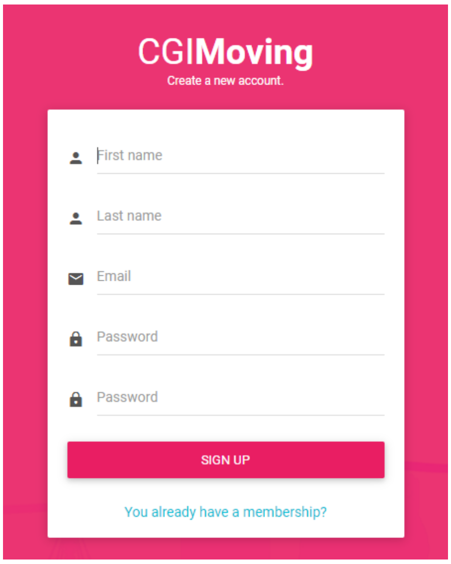


Figure 2.2: The register view of CGIMoving.

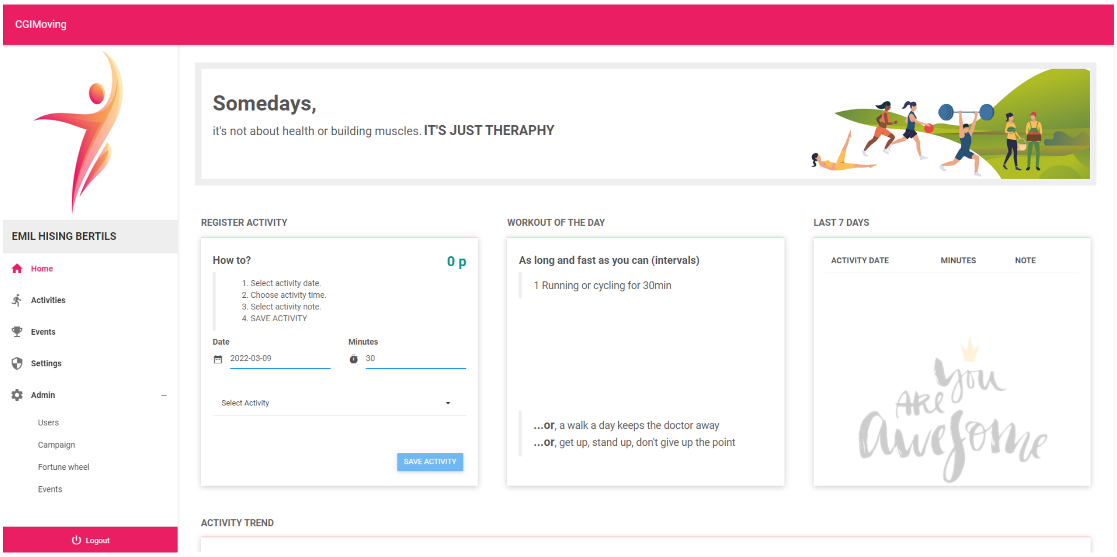
2.1.2 Users

When a user logs into the application they are presented with the home view, see Figure 2.3. In this view, the user can log an exercise session, view "Workout of the day" recommendations, and review their weekly activity log. When a user logs an activity they progress towards different levels, each level reached gives the user a lottery ticket, which provides them with a chance to win a prize at the end of the campaign. An administrator (admin) decides the maximum amount of lottery tickets that a user can earn during the campaign, which is equal to the number of levels the campaign has. If a user scrolls down in the home view, they can see a graph, see Figure 2.4, showing their activity trend. In the activity trend graph, the users' logged exercise is compared to how much exercise is required to reach the campaign levels.

On the left side of Figure 2.3 there is a navigation area where the user can navigate to the different parts of the website. If a user click on Activities in the navigation field they are presented with a table of logged activities. On the Events view, the user can see if there are any planned events that they can participate in. The user is able to edit their account information in the Settings view, there they can also change their password or delete their account. The admin drop-down is only visible if they are an admin of the application.

2.1.3 Admins

An admin can view other admins or users on the Users page under "Admin" in the navigation field, here they can also create a new user profile. In the Campaign view admins have the possibility to change the current campaign or delete it. There they can also add predefined activities that a user can then select from when they log their exercise. When the campaign has ended and it is time to give out prizes an admin can navigate to Fortune Wheel to spin a wheel that decides who wins prizes. Under Event



Somedays,

it's not about health or building muscles. IT'S JUST THERAPHY

How to?

1. Select activity date.
2. Choose activity time.
3. Select activity note.
4. SAVE ACTIVITY

As long and fast as you can (intervals)

1 Running or cycling for 30min

...or, a walk a day keeps the doctor away

...or, get up, stand up, don't give up the point

Figure 2.3: The home view of CGIMoving.

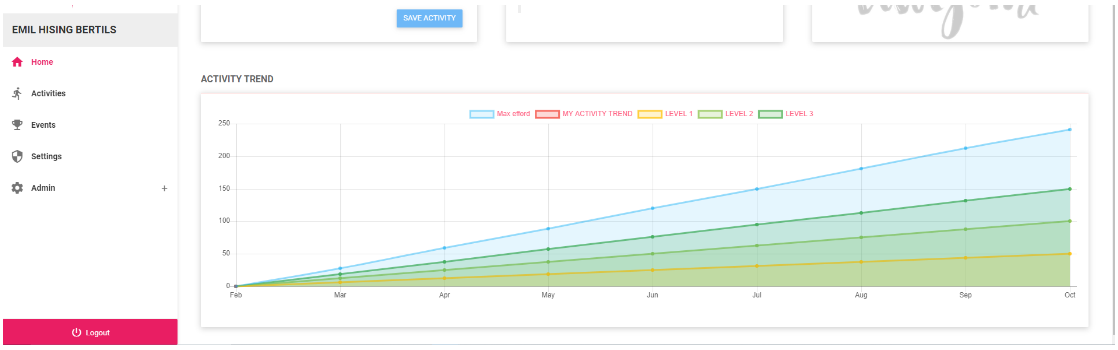


Figure 2.4: The bottom part of the home view of CGIMoving.

an admin is able to create events that users might join to earn extra lottery tickets.

2.2 Similar Functionality

Since the primarily goal of this project is to create an invite-only application, see Section 1.3, the registration web page should only be accessed by a certain authorized individuals. Newsletters have a structure that is very similar to this. In the footer of newsletter emails, there usually is a button that says "Unsubscribe", which when clicked stops the application from sending the user any further emails. The unsubscribe button redirects the user to a web-page, where the website runs functions on the server-side to know who clicked the link and which email to remove from their newsletter list. The same fundamental ideas can be applied to an invite email, it can be determined who clicked the link and if they received a valid invitation to the application, allowing them to register or not. This project is about figuring out how this is done and applying it to the CGIMoving application and convert it into an invite-only website.

Another example is that certain websites have a certain way of dealing with users forgetting their passwords. The user can request a password reset, which then sends them an email with password-reset link that works just as described above, or it contains a code which the user then has to enter manually on a password-reset page to then successfully reset their password. The second approach to this requires more user input for no reason, systems that require user input have a higher chance of failure due to human error [7], which is why it has been decided to combat this problem with the first approach instead.

2.3 The Hash Algorithm

To be able to have an invite-only application, the admins need a way to send unique invitations to users that attackers cannot figure out by brute-forcing the system. Hashes are a way to provide the invitations with this uniqueness.

> "Hashes are the result of a mathematical function that converts a text string (no matter the length) into an encrypted string of a fixed length. For every given piece of data input, algorithms such as Message Digest 5 (MD5) or Secure Hash Algorithm (SHA) fundamentally generate a unique, fixed-length string—the hash value." [8]

Detailed explanations of hash functions are provided in [9, 10, 11].

Since the invitation is erased as soon as the invited user registers on the website, attackers only have a limited time to brute-force the invitation identifier. Every invitation will require a unique identity that is generated in such a way that any potential attackers will find it ineffective to proceed with their brute-forcing techniques. For that reason it was decided to make use of SHA256, since it has no known vulnerabilities that makes it insecure and it has not been broken yet, unlike some other hash algorithms [12]. For example, MD5 remains a popular hash function. Even though it was compromised nearly two decades ago according to [13]:

> "Nowadays it is actually possible to artificially produce MD5 collisions. All you need is time, hardware and the proper software."

This is an excellent paper detailing the vulnerabilities of multiple hash algorithms. In [14], the authors shows how to break MD5 and other hash functions.

Chapter 3

Design

This chapter will discuss the design choices made while fulfilling the stated goals of the project, which can be found in Section 1.3. It will begin by outlining the problem with the current behaviour of the application and then move on to describe our solution to this problem. This will be discussed separately for each of the features of the project.

In Section 3.1, the design choices made to solve the primary goal, i.e., *how to stop random people from registering to the application*, is described. Section 3.2 will describe the minor visual updates done to the application. The decisions on how to solve the second secondary goal of the project, i.e., *notify users of a new campaign starting*, will be discussed in Section 3.3. Finally, Section 3.4 will focus on how to let the admins of the application handle blocked users.

The CGIMoving application is based on two design patterns. These are *Model-View-Controller* pattern (MVC), further described in [15], and the *Repository* pattern, further described in [16]. These patterns will be used in our solutions as well.

3.1 Invite-only Application

This section will be divided into two sub-sections: first, it will detail the solution to the problem of random users accessing the application, and second, it will cover the newly introduced capabilities for administering invitations.

3.1.1 Redesign to an Invite-Only Application

The issue with the application's current implementation is that random users have begun to register and it takes time for admins to block all of these unwanted users. Our response to this issue is to make the application an invite-only application. An invite-only application is one that requires users to receive an invitation to join, rather than just finding the application's URL and registering an account. This method introduces several new issues, which are detailed below.

1. How will the user get the invitation?

2. How to prevent not invited people form registering to the application?

3. How to make sure the invitation is only used once?

To resolve issue one, a view in which an admin can enter an email address and send an invitation by email was implemented. The email contains a link to the application's registration page, where the recipient can create an account. The input field is a multi-line text field in which the admin can enter one or more email addresses that will receive their own application invitation. There is a notification informing the admin that the invitations have been issued if the emails have been entered correctly, or an error with information on what went wrong if the emails have not been sent.

Our answer to issue two is to remove the link to the register page from the login view, as shown in Figure 2.1, and to replace it with an invitation-link that contains a code that is validated upon visiting the register view. This prevents anyone without a valid code from accessing the registration view. This code is generated using the *SHA256* hash algorithm, as detailed in Section 2.3. This is to ensure that the code is extremely difficult to guess and also to ensure that each invitation has a pseudo uniqueness.

Our response to issue three is twofold. To begin, upon registration, the invitation-specific code is removed, ensuring that the invitation can only be used once. Second, the code is validated once more during registration to prevent it from being shared before to registration and subsequently used many times.

3.1.2 Administration of Outstanding Invitations

Following the invite-only change, it was discovered that the only option to view and delete outstanding invitations was to check directly in the database, which was considered a concern. The product owner agreed and desired a resolution. Our answer to this problem is to provide a dynamic list that allows an admin to view all pending invitations and withdraw them if desired. When an admin revokes an invitation, the database entry for that invitation is deleted. Additionally, a notification will be displayed to reflect that the invitation has been revoked, and a button will be provided with the notification to undo the change in the event that the admin accidentally hit *revoke*.

3.2 Visual Updates

After completing the implementation of the solution that fulfills the primary goal, the work on improving the website's minor aspects could begin. The user page contains certain buttons that are not styled correctly and text that overflows off the button if it is too long. This can be corrected by resizing the buttons or modifying the text content. Certain buttons are colored incorrectly and do not fit the website's theme. This can be remedied by reusing previously used colors and avoiding the introduction of new colors on commonly used buttons.

3.3 Notification on Campaign Starting

There is currently no way, in the application, for the admins to notify the users that they have started a new campaign. Which means that either the users wont get notified or that an admin has to notify them manually outside of the application. To make sure that the users are notified of a new campaign without increasing the work for the admins, the product owner wanted a feature to automatically inform users of a new campaign.

When a campaign is about to begin, every user of the application should be notified by email. The email should use the same design style as the invite email form the invite-only feature. The email should notify the user that there is a new campaign starting and contain information of when it is scheduled to start, which is when the user will be able to start registering their activities. The email should also contain a link, directing the user to the “Home” page of CGIMoving where they can log their activities.

3.4 Administer Blocked Users

The product owner wanted to make it easier for the admins of the application to determine whether a user is blocked or not. To do this, the list of all users will be updated. A new column will be added, showing whether the user is blocked or not. The admins will also be able to order the list alphabetically, ascending or descending, on a desired column to help filter through all the users.

Chapter 4

Implementation

4.1 Invite-only Application

This section will discuss the implementation of the functionality that allows users to join solely via invitation. It will detail the functionalities that were implemented and demonstrate how the application appears after our implementations have been finalized.

4.1.1 Redesign of the Login/Register User Interface

Before the work on the back-end code for the invite-only features could begin, access to the registration page for non-invited users had to be restricted. As seen in Figure 2.1, the Login page with a “Register” button at the bottom is the first thing customers see when they visit the website. As shown in Figure 4.1, the "Register" button was removed from the login panel.

Visitors can access the registration page using the `cgimoving.com/Identity/Account/Register` URL without having to click the register button. To address this, it would have to be ensured that the application searches for a hash in the URL and that the hash is kept in the database containing invitation hashes. Only if the hash is valid will

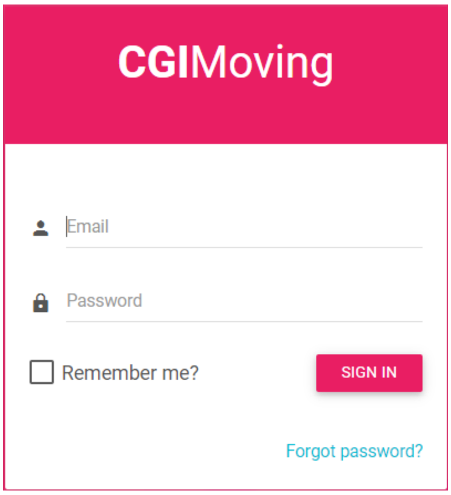


Figure 4.1: The new login view without the register button.

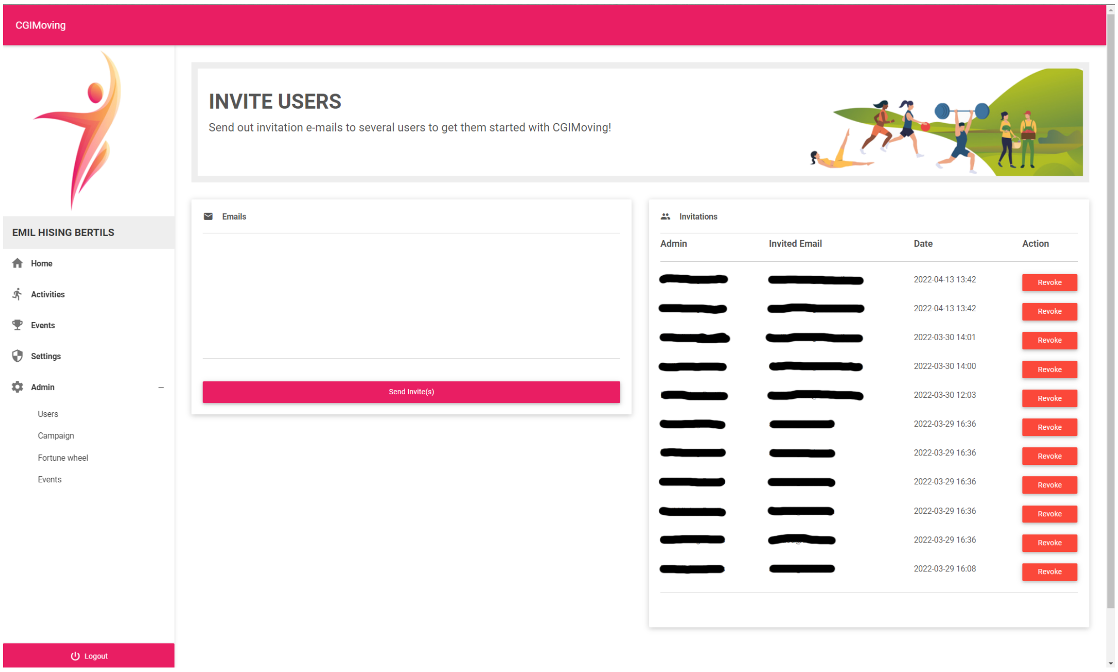
the user be able to access the registration page. If the hash is not found in the database, the user is forwarded to the login page.

Multiple users will be able to access the registration page if they visit the same registration URL with the same hash. This means they can all register with the same hash, which is a problem. When a user completes registration by submitting the registration form, the website rechecks the hash in the database before removing it permanently. This prohibits numerous individuals from using the same hash.

4.1.2 Invitation Page

Next, admins required a way to send email invitations to new users. A new link to the “Invite Users” page on the “Users” page was added. The plan was to send many emails simultaneously using a format provided by CGI.

The invite users page, which can be seen in Figure 4.2, has two sections, one to the left and one to the right of the page. The section on the left contains a text area where an admin can enter email addresses of users they want to invite to the application. It is possible to enter the text in different formats since the application will format the text and remove anything that is not an email address. The right section contains a list of all outstanding invitations and gives the admin the possibility of revoke the invitation, and also delete the hash entry from the database.



INVITE USERS

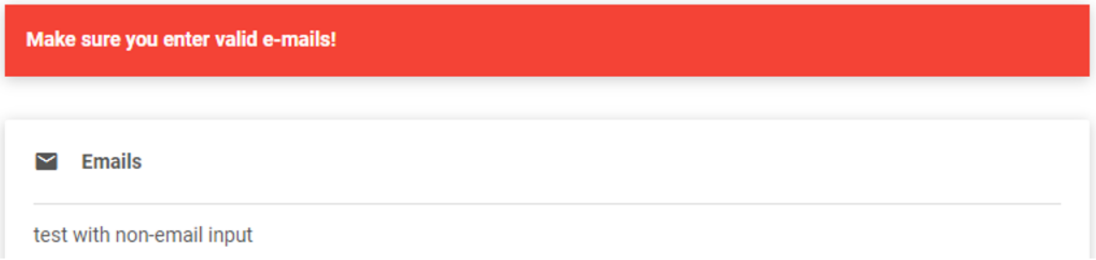
Send out invitation e-mails to several users to get them started with CGIMoving!

Admin	Invited Email	Date	Action
[redacted]	[redacted]	2022-04-13 13:42	Revoke
[redacted]	[redacted]	2022-04-13 13:42	Revoke
[redacted]	[redacted]	2022-03-30 14:01	Revoke
[redacted]	[redacted]	2022-03-30 14:00	Revoke
[redacted]	[redacted]	2022-03-30 12:03	Revoke
[redacted]	[redacted]	2022-03-29 16:36	Revoke
[redacted]	[redacted]	2022-03-29 16:36	Revoke
[redacted]	[redacted]	2022-03-29 16:36	Revoke
[redacted]	[redacted]	2022-03-29 16:36	Revoke
[redacted]	[redacted]	2022-03-29 16:36	Revoke
[redacted]	[redacted]	2022-03-29 16:08	Revoke

Figure 4.2: The invite users view.

The left panel is used to send invites through email to the email addresses supplied in the text field when an administrator presses the *Send Invite(s)* button. As shown in Figure 4.3 and Figure 4.4, a notification pop-up informs the administrator of the outcome of the process. The notification is seen above the leftmost card.

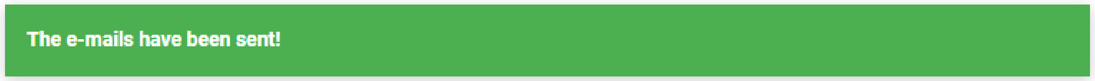
When the administrator presses the button to send the invitations, an email is drafted. Figure 4.5 depicts the email. The email has the “Register!” button, which is a link to the application’s registration page with the unique hash obtained during the composition of the email. When a user hits the link, they are allowed to register for a CGIMoving account using any email address of their choosing.



Make sure you enter valid e-mails!

test with non-email input

Figure 4.3: Wrong input notification.



The e-mails have been sent!

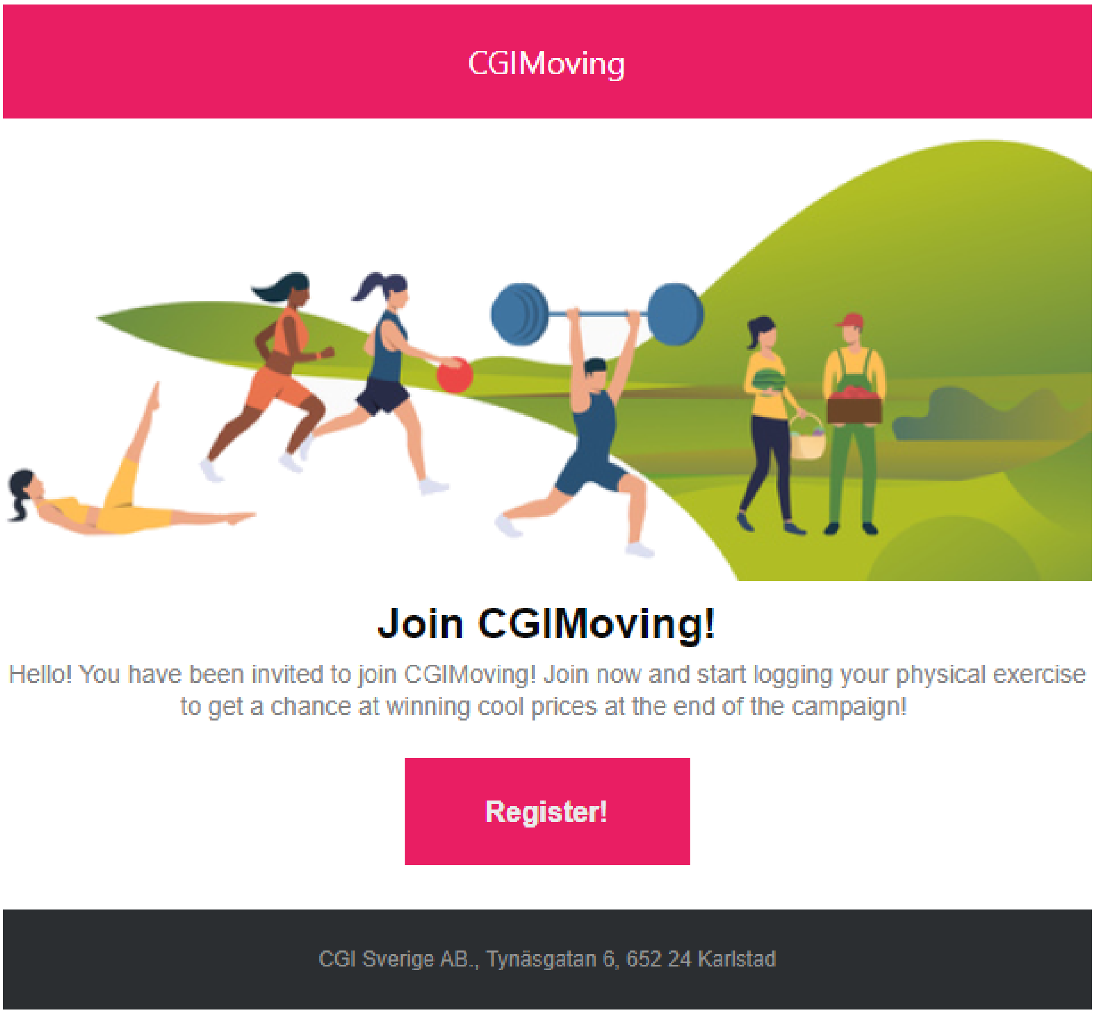
Figure 4.4: Notification that the emails were sent.

4.1.3 Creating Hashes

To construct a fully functional invitation system, hashes attached to URLs provided to the invited user could be used, and then verified by the database to determine whether or not the hash is valid. When an administrator clicks the “Send Invite(s)” button on the invite user page, separate hashes are generated for each invited user.

As mentioned above, the SHA256 algorithm was used to generate the hashes, and the back-end has been modified to ensure that the input to the hash is unique for each user, even if a lot of emails are sent simultaneously. Once a hash has been generated, it is appended to the `cgimoving.com/Identity/Account/Register?hash=` URL and an email invitation is sent.

If the hash is correct, when a user clicks on the registration link in the email, they are routed to the registration page. If the hash is invalid, the user will be redirected to the login page. This is the result of code that extracts the hash from the URL and verifies that it exists in the hash database, and displays a different page depending on the outcome of the result.



CGIMoving

Join CGIMoving!

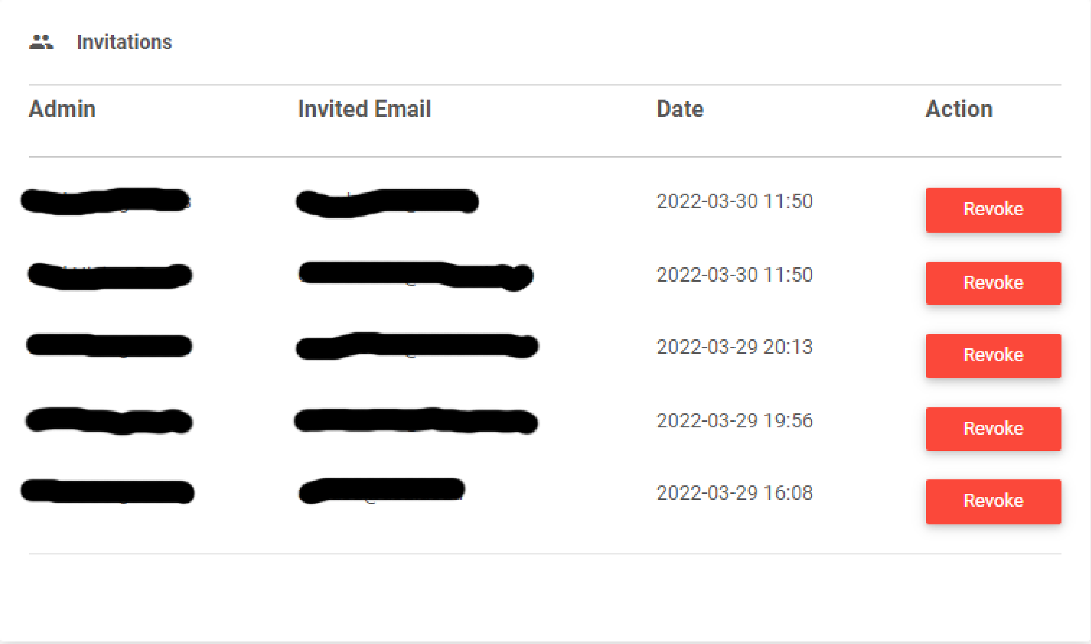
Hello! You have been invited to join CGIMoving! Join now and start logging your physical exercise to get a chance at winning cool prices at the end of the campaign!

CGI Sverige AB., Tynäsgatan 6, 652 24 Karlstad

Figure 4.5: Email invitation to join CGIMoving.

4.1.4 Adding the Invited Users List

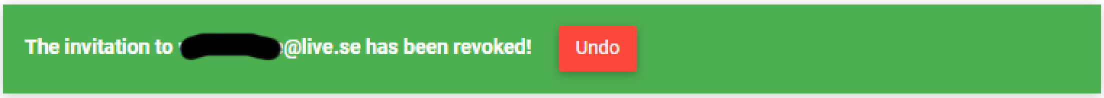
After converting the website to an invite-only application, the scope was expanded to add more quality of life aspects to the feature. It was suggested to CGI that a “Invitations” panel should be added to the right side of the invite users page, and they agreed. This panel displays an ordered list of all invitations sent by all administrators, as shown in Figure 4.6. It displays the admin’s name, the invited user’s email address, and the date. An admin can also revoke an invitation from this page. When revoking an invitation, a success message is displayed, and the administrator has a few seconds to undo their action before the hash is removed forever. The notification is depicted in Figure 4.7.



Invitations

Admin	Invited Email	Date	Action
[redacted]	[redacted]	2022-03-30 11:50	Revoke
[redacted]	[redacted]	2022-03-30 11:50	Revoke
[redacted]	[redacted]	2022-03-29 20:13	Revoke
[redacted]	[redacted]	2022-03-29 19:56	Revoke
[redacted]	[redacted]	2022-03-29 16:08	Revoke

Figure 4.6: List of outstanding invites.

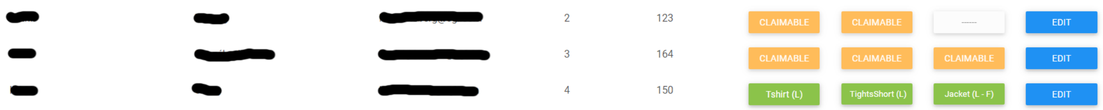


The invitation to [redacted]@live.se has been revoked! Undo

Figure 4.7: Notification of a revoked invitation.

4.2 Visual Updates

This section describes the visual update implemented in the admin's view. During the creation of the invite functionality, we were not satisfied with a few other minor issues in the application. There is a list of all current application users on the user page. This list contained buttons with excessively long text, so that the text extended beyond the button to the right. Certain texts were shortened where the meaning could be maintained. In other instances, the text font size was reduced to fit within the button. In one instance, the wording “Not Selected” was changed to “Claimable”. This is illustrated in Figure 4.8. The color of the “EDIT” button was changed from brown to blue to make the button’s function more apparent.



[redacted]	[redacted]	[redacted]	2	123	CLAIMABLE	CLAIMABLE	------	EDIT
[redacted]	[redacted]	[redacted]	3	164	CLAIMABLE	CLAIMABLE	CLAIMABLE	EDIT
[redacted]	[redacted]	[redacted]	4	150	Tshirt (L)	TightsShort (L)	Jacket (L - F)	EDIT

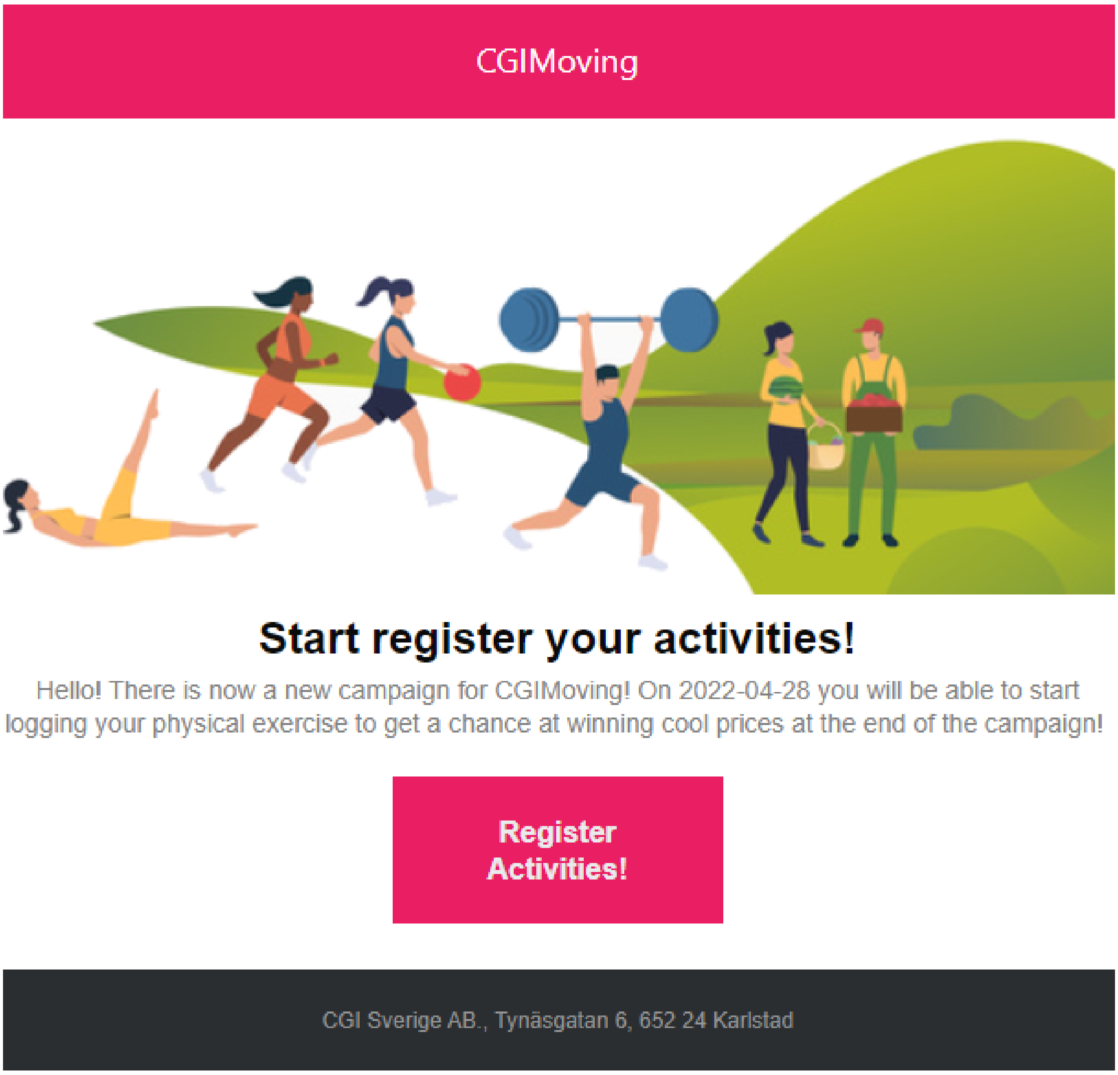
Figure 4.8: List of current users.

4.3 Notification of a Campaign Starting

This section describes the feature that notifies all users when a new campaign begins. This feature works by sending email notifications to all the application users anytime an administrator launches a new campaign. The email’s information varies based on when the campaign is scheduled to begin. The email offers a link to CGIMoving where recipients can register for the campaign. The email can be seen in Figure 4.9. If the campaign is scheduled to begin at a later date, the email will also include the campaign’s launch date.

4.4 Administer Blocked Users

This section describes the update to the list of users that is available to the admins, to help them manage blocked users. To help the admins to see which users are blocked, a new column, named “STATUS”, was added. The column shows an icon if that user is



CGIMoving

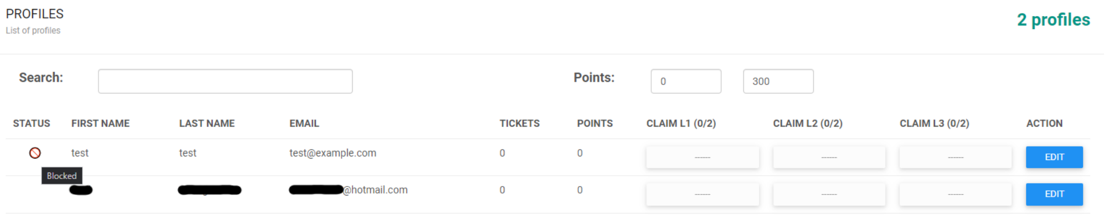
Start register your activities!

Hello! There is now a new campaign for CGIMoving! On 2022-04-28 you will be able to start logging your physical exercise to get a chance at winning cool prices at the end of the campaign!

CGI Sverige AB., Tynäsgatan 6, 652 24 Karlstad

Figure 4.9: Email notification that a new campaign has started.

blocked and hovering over it displays the text “Blocked”, as can be seen in Figure 4.10. To make it possible to see all the blocked users more easily, the columns were made sortable. Clicking on a column header orders the list alphabetically, first ascended and then descended if clicked again, based on the contents of that column. The button to block a user was changed to have a orange color instead of the light green it had before, the “UNBLOCK” button already had the same orange color. This helps with signaling the severity of the action to the admin.



PROFILES

List of profiles

2 profiles

Search:

Points: 0 300

STATUS	FIRST NAME	LAST NAME	EMAIL	TICKETS	POINTS	CLAIM L1 (0/2)	CLAIM L2 (0/2)	CLAIM L3 (0/2)	ACTION
🚫 Blocked	test	test	test@example.com	0	0	------	------	------	EDIT
	[redacted]	[redacted]	[redacted]@hotmail.com	0	0	------	------	------	EDIT

Figure 4.10: List of all users of the application.

Chapter 5

Results

This chapter will discuss the result of this Bachelor thesis project. It will briefly describe the result and discuss whether or not the goals were fulfilled. It will then present an evaluation of the result.

The project had one primary goal and three secondary goals, see Section 1.3. The primary goal was to convert CGIMoving to an invite-only application was completed and is discussed in Section 5.1. All three secondary goals were completed. This chapter will discuss each completed goal separately.

5.1 Invite-only Application

The primary goal of this project was to convert CGIMoving to an invite-only application and as such was prioritised highest of the requirements specified by the product owner. The product owner specified that they wanted the application to be shifted from open registration to something where they had more control over who was able to use the application. To achieve this CGIMoving was changed into an invite-only application.

5.1.1 Results

To change CGIMoving into an invite-only application, a new view was added to the existing application. In this view an admin can send email invitations to people with a registration link, which will take the user to the registration view. The registration view is only accessible with a hash generated uniquely for each invite. The admin can also view all the outstanding invitations and revoke them if desired. All of these features combined makes sure that the admins of the application have very good control over who gets to use the application, which in turn means that the goal is fulfilled.

5.1.2 Evaluation

This functionality is clearly an improvement of CGIMoving, since the admins now have a lot more control over the users than before. The implementation is secure from unwanted users as long as hashes of outstanding invitations are not leaked and that invited people do not give away their one-use-only registration code to someone else. It is easy to invite a single user as well as multiple users at once. To simplify the process for the product owner, an admin can input the product owners generated standard output of email addresses of their employees and the application understands what parts are the email addresses and filter out the rest. The control over outstanding invitations is simple and a revoke button is provided. The notification messages makes it easy to understand if the action went well or not and gives indications as to what went wrong in case of an error.

5.2 Visual Updates

The visual updates were the goal with the second highest priority specified by the product owner and as such became the first secondary goal of the project. The goal was to update some visual aspects of the application since some buttons had the text running outside of the button.

5.2.1 Result

The text of buttons on the admin view *Users* were changed to fit inside of the button and still keep the same meaning as the previous text. The color of the edit buttons in the same view was changed from brown to blue to look more pleasing and better convey what they do. A banner was also added to the fortune wheel view so that it resembles the other views a bit more.

5.2.2 Evaluation

The result of this change was appreciated by the product owner, even if the change is rather small. No information was lost and it is much easier to understand what the buttons do, mainly due to the fact that a user now can read the complete text on the buttons.

5.3 Notification on Campaign Starting

Notification on a campaign start was the goal with the third highest priority in the project. For this goal, an email notification was to be implemented, informing the users of the application that there is a new campaign starting. The email contains information about when the user will be able to register activities for the campaign and a link that they could follow to start registering their activities.

5.3.1 Result

When an admin creates a new campaign, an email is sent to each user of CGIMoving. The email informs the user that a new campaign has been started and provides the user with information about when they will be able to register their activities. It also contains a button that will direct the user to a page on CGIMoving where they can register their activities after they have signed in to their account. The email is designed in the same style as the invite email from the invite-only feature.

5.3.2 Evaluation

The implemented feature includes all of the requirements specified for the goal. It improves CGIMoving since all users will be aware that they can start using the application as soon as possible and not through word of mouth or manual work from the admins. Since the email is in the same style as other emails from the application, the user will be familiar with the design and gather the information more quickly.

5.4 Administer Blocked Users

The fourth highest priority goal was to improve administration of blocked users. For this feature the product owner wanted to be able to see which users were blocked at a glance while looking at the user list. They also wanted to be able to sort the list of all users based on a specific column.

5.4.1 Result

There is now a new column in the user list showing the status of the user. The status can at the moment only be empty or blocked, which is shown with an icon. Hovering over the icon with the mouse shows a little box with the text “Blocked”. The block button now has the same orange color as the unblock button has, instead of light green. Clicking on the column header will sort the rows in ascending alphabetical order based on the column's content, clicking the header again will sort in descending alphabetical order.

5.4.2 Evaluation

All specified requirements were met and the product owner was satisfied with the result. Previously the only ways of knowing whether a user was blocked or not was to click on the “Edit” button for that specific user or check the database manually. Having the information at a glance reduces the work required by the admin and provides them with a better sense of how many of the users are blocked. Being able to sort the list on the columns is useful to easily get a list of all the blocked users without having to have multiple lists or typing keywords into a search field and can also be used for other things, such as ranking the users by lottery tickets earned for example.

Chapter 6

Conclusions

This chapter is organized as follows, Section 6.1 will compare how well the project went with our work plan presented in Chapter 1. It will then discuss the benefits and drawbacks, if there are any, of our implemented features. Lastly it will discuss our experience from the project. Section 6.2 will outline future work that could further improve the application. Finally, in Section 6.3 some concluding remarks are provided.

6.1 Discussion

In the following subsections, a discussion of the work process, implemented features, and our experiences from the project is provided.

6.1.1 Work Process

At the start of the project we had decided to work in two weeks sprints, which we did not manage to follow in the beginning due to our limited experience with the work structure and inexperience with the programming language and the design patterns used in the application. After about a month into the project, we were able to realise the two weeks sprint plan and followed it going forward. During the project we maintained communication with the product owner and they were present at the sprint planning and review meetings. To have the product owner present at the sprint planning made it much easier to understand the requirements for each feature since we could ask questions about them even before we started working on implementing them.

6.1.2 Implemented Features

The main benefit of our implementations is to reduce the administrative work required for the application or at least make the work easier for admins where work could not be automated. One example where the work is considerably reduced is that the admins will no longer have to spend time banning unwanted users from the application, which was considered a problem by the product owner. The work is made easier by the ease of inviting multiple users to register for the application at once. The main drawback of our implementation is that the admins will have to manually invite users to the application, instead of the users being able to register on their own. However, inviting users only needs to be done once per employee, and since there are more unwelcome users than new employees, inviting new employees requires less effort than continuously banning unwelcome users. Since the positives outweigh the negatives, we consider our implementation of the feature to be an improvement.

6.1.3 Experiences

During this project we have got to experience a lot of new things, ranging from completely new, to us, design patterns in a full fledged application. This means that we have encountered new problems we have had to solve, which, in turn, have led us learning new things. We have expanded our knowledge of the programming language used in the application as well as learnt how to code using the two design patterns “MVC” and “Repository”. Since the project was an improvement of an existing application, we have learnt new things about both back-end and front-end programming. We have also learnt how to work with a database and how to generate a hashes as well as how to create a good looking web page through the use of HyperText Markup Language (HTML) and Cascading Style Sheets (CSS).

6.2 Future Work

There are a lot of features that could be added to improve the current application in the future. In this section, the features we believe should be prioritized and why they would be beneficial to the application will be outlined.

6.2.1 Multiple Campaigns

One proposed feature is the possibility to have multiple campaigns ongoing at the same time, which would make it possible for CGI to run local campaigns for each office. Having local campaigns would be beneficial for a few reasons. It would be easier to manage and give out rewards at the end of the campaign since most, if not all, users will be within close proximity of each other. It would also make choosing motivating rewards easier, since there will be fewer opinions on what is a good reward. It would make it easier for the users to view events happening in the campaign since all events will be created by their office and the event list will not be filled with events from offices form the other side of the world, assuming all offices would use the event feature.

6.2.2 Expiring Invites

Another feature that would improve CGIMoving is that invites to the application expire after a specified period of time. To do this, a “Expire after X day” section could be added to the invite users page, allowing admins to specify the number of days before an invitation expires. This could be beneficial since it would make sure there is not a bunch of unused invites stored in the database, making the database smaller and reducing the risk of a malicious unwanted user guessing a valid hash.

6.2.3 Settings for Email Notifications

Some users might not want to participate in the current campaign and may consider the email notifications about a new campaign to be spam. Because of this, it would be an advantage if there was a setting where these users could turn off email notifications. There could also be settings that decides to which degree users gets notified trough email, to cater to a wider user base.

6.2.4 News

A news feature is something that the product owner already wants for their application. This feature would function in a similar way as the event feature except it would not have a start and end times and dates. It could be used to give the users information about for example what rewards they can win, who won last nights bowling competition or how many participated in the last event. It could share the “Event” page and have a summary of the latest news on the “Home” page.

6.3 Concluding Remarks

The primary goal of this project was to implement an invite-only feature as well as removing the normal registration process in order to convert CGIMoving into an invite-only application. We managed to implement this feature in a way that satisfied the requirements specified by CGI. We also implemented yet another feature in conjunction to this that makes the administration of outstanding invites much easier. There were also three secondary goals in this project, of which we completed all three. These implementations also satisfied the requirements specified by CGI.

Bibliography

[1] Frank W Booth, Christian K Roberts, and Matthew J Laye. Lack of exercise is a major cause of chronic diseases. *Comprehensive physiology*, 2(2):1143, 2012.

[2] J Brailovskaia, XC Zhang, D Cai, S Lu, ZH Gao, and J Margraf. The benefits of physical activity and positive mental health for reducing the burden of covid-19: Validation from a cross-sectional and longitudinal investigation in china and germany. *International Journal of Mental Health and Addiction*, pages 1–14, 2021.

[3] Arturs Locmelis and Mathias Thorén. Improvement of a physical exercise logger, 2020.

[4] Arvid Gördén and Isabelle Rybank. Utökning av en träningsapplikation. (Swedish) [expansion of a physical exercise logger], 2021.

[5] Henrik Kniberg. *Scrum and XP from the Trenches*, volume 2. C4Media, 2015.

[6] Robert C Martin, James Newkirk, and Robert S Koss. *Agile software development: principles, patterns, and practices*, volume 2. Prentice Hall Upper Saddle River, NJ, 2003.

[7] IEvangelist. Security and user input | Microsoft Docs. `https://docs.microsoft.com/en-us/dotnet/standard/security/security-and-user-input`, 2021. Accessed on 15/04/2022.

[8] Dora Tudor. What is hashing and how does it work? `https://heimdalsecurity.com/blog/what-is-hashing/`, 2021. Accessed on 15/04/2022.

[9] Dieter Gollmann. *Computer Security, 2nd edition*. John Wiley & Sons, 2006.

[10] Matt Bishop. *Introduction to Computer Security*. Addison-Wesley, 2005.

[11] William Stallings. *Cryptography and Network Security, Principles and Practices, 4th edition*. Prentice Hall, 2006.

[12] Peter Callaghan. Why you should use SHA-256 in evidence authentication. `https://blog.pagefreezer.com/sha-256-benefits-evidence-authentication`, 2022. Accessed on 15/04/2022.

[13] Gorka Ramirez. MD5: The broken algorithm. `https://www.avira.com/en/blog/md5-the-broken-algorithm`, 2015. Accessed on 15/04/2022.

[14] Xiaoyun Wang and Hongbo Yu. How to break MD5 and other hash functions. In Ronald Cramer, editor, *Advances in Cryptology – EUROCRYPT 2005*, pages 19–35, Berlin, Heidelberg, 2005. Springer Berlin Heidelberg.

[15] Steve Smith. Overview of ASP.NET Core MVC, 2022. Accessed on 09/06/2022.

[16] Tom Dykstra. Implementing the Repository and Unit of Work Patterns in an ASP.NET MVC Application, 2021. Accessed on 09/06/2022.

Appendices

Appendix A

Abbreviations and Acronyms

Table A.1: All the abbreviations and acronyms used in the thesis.

Admin	Administrator
API	Application Programming Interface
CSS	Cascading Style Sheets
HTML	HyperText Markup Language
MD5	Message-Digest 5
MVC	Model-View-Controller
SHA	Secure Hash Algorithms
SHA256	SHA with 256 bits output
URL	Uniform Resource Locator