Bilkent University

Department of Computer Engineering

Object-Oriented Software Engineering Project

CS 319 Project: Tempo

Final Report

Project Group: 1.G

Member Names: Mert Saraç, Burak Erkılıç, Kaan Kıranbay, A. A. M. Jubaeid Hasan Chowdhury

Course Instructor: Eray Tüzün

Final Report
May 5, 2018

Table of Contents

1. Introduction

At this point, we have finished all parts of the system. However, it must be expressed that since we have limited time for this project and we tried to add too many features, we have to exclude two features. These features are two non-pivotal features to our application: chat and habit tracker. Like we mentioned in our design report, efficiency of the most common features is more important to the users than adding more none-pivotal functionalities and we decided to follow that ideal. On the other hand, we have an authorization page that works with database. User can easily register by entering his/her email, username, name and password or login to existing account. After accessing their account, users can easily create events and see them on the calendar. Also, if it is personal event, the user can easily change its timing by dragging it with mouse on the calendar. To remove an event, the user should use "-" button which works as a delete event button. By clicking "=" button which means menu button, the user can see his/her informations and preferences. Also, the user can easily change these informations. In the calendar page, there are add friend and remove friend buttons. By using these, the user can manipulate his/her friend list easily.

2. Design Changes

Firstly, like it was mentioned, chat system and habit tracker feature were excluded from the system. As our project is implemented in a object-oriented manner with good design patterns, we can easily add this features in later version if the user demands.

Secondly, to make it more easily accessible, add friends and remove friends buttons are moved to calendar page from the profile page. In this way, the user can access most of the common actions in the calendar page like add friend, remove friend, add event, remove event, change an events location. In design, we have 2 new classes that are storage and user. Storage class is added after final report of first iteration. Storage class contains numerous arraylists that holds different informations from database like friends. User class contains his/her username, password and profileID and it is connected with database. The user class is used to authenticate during log in.

Thirdly, in our design, Event class was a abstract class and every other personal event classes were subclass of it. However, as there were a lot event type classes this design wastes a lot of unwanted space in the database. In order to eliminate it, we combined them into a single concrete event class that can be used as any of the personal event type depending on the attribute. This is a good demonstration of model transformation by collapsing objects.

Fouthly, weekly view of the calendar is also excluded due to limited time and it is not vital for the system. In the new design, we have just monthly view of the calendar that provides more control over events since they are all can be seen in the calendar. In addition, weekly view is not effective when user wants to change timing of the event in the calendar. For example, if the user wants to move an event to other week, he/she needs to open monthly view which is an unpleasant situation.

3. Lessons Learnt

We have learned countless number of things throughout the development process of this project. Some of the most important ones are listed below.

First, we figured out how to manage time for group meetings and how to work together so that all of our works up to date for everyone. Since all of the group members have other lectures and things to do, it was really hard to arrange meetings even in weekends. Let meetings alone, communication among group members was also hard due to some technical and personal problems. We tried to solve this issue by having meeting on online chatting application such as Discord, Skype etc. and also by working on Google doc and Version Control Softwares like GitHub to keep our works instantly synced. We also tried to schedule the meetings in the make up hour for this course or in the weekends.

Second lesson is that a lot of design transformation happens when you implement the system for the first time. In the design report of the project, we failed to understand the advantage and disadvantage of some our designs which become more clear when implemented it. In the design report we were more strict about sticking to object oriented design principles and patterns. However, during implementation we learned that taking some ‘shortcuts’ can make life a lot easier and make the system a lot more efficient. For example, the case about Event class which was mentioned in Design Change’s second part. It looked more object-oriented to have seperate sub-class for each event types. However, during implementation we realized that combining them together would reduce complexity and save database space while keeping all the functionalities. Another is that we wanted to have separate view and controller for each of our functionalities. However, we realized that user wants easy access to the functionalities he/she uses more commonly than using separate tabs or views to access them. Hence, we tried to add as much of the common functionalities in the main calendar view such as to-do-list, friend-list, add friend, logout etc. without cluttering it.

Third, we got adapted to the process of continuous build. At first we faced issues regarding synchronization of codes between different group members which lead to wasting of our time. For example, at one time, one of group member worked on the responsive layouts for pages. However, due to changes in design made by other group members, this process was gone to waste and this group member had to do these change again for the new design. To minimize it, we all started to use Intellij IDEA which allows easy use of VCS such as GitHub and features of "Update Project" and "Commit", it becomes easy to work with a harmony. Each member of our group needs to update project before starting to work and run it again to check for errors. Also he needs to review the updated code and give feedbacks. This continuous build helped us enormously especially for our motivation as our system is always improving and always works! We first build the skeleton of the system such as database connection and calendar view so that we don't need any drivers or stubs then we continuously added, tested and integrated all other features.

Finally the most shocking thing learned from this project is that software engineers doesn't leap into coding from day one. All the reports, meetings, class hours, notes helped us to deeply understand our application even before starting the implementation. Even more shocking thing is that these reports are actually better to analyze and understand our project than actual implementation because they hide all the unnecessary details and represent informations in a more human understandable way that can be changed readily. Hence, we think we understood the concept of modelling far more deeply than before doing this project.

4. User Guide

4.1 System Requirements and Installation

Our application has very simple system requirements. In order to use Tempo, the customer needs to have minimum JDK 1.8 installed. Another requirement is a stable internet connection. Since Tempo interacts with the database, not having stable internet connection will cause problems with calendar updating and Smart Events which is an interactive event with friends and needs internet connection to request or accept it. However, internet with a low bandwidth is fine since the application transfers only low quantity of data with the database.

4.2 How to Use

4.2.1 Opening the Application

The Tempo Application can be opened with the help of .exe file. .exe file of the application is available at GitHub.

Alternative way to using it, loading Java IDEs such as NetBeans, Eclipse or Intellij IDEA. Since we used Intellij IDEA in implementation process of the system, we recommend Intellij IDEA.

.

4.2.2 Registering and Logging-in to the System

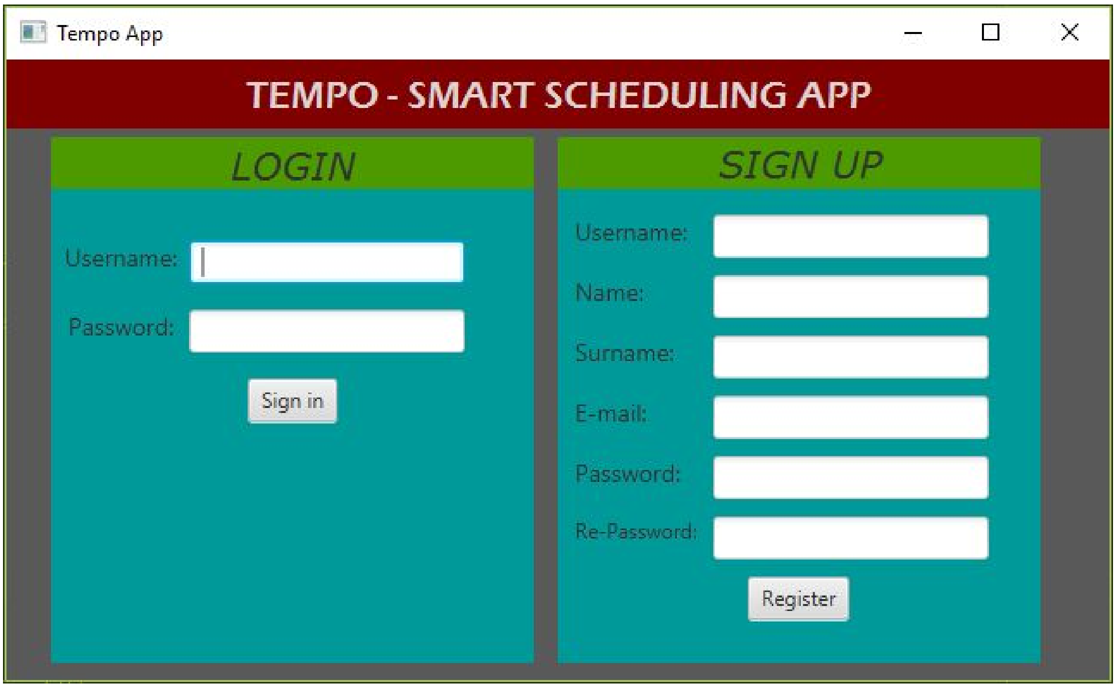


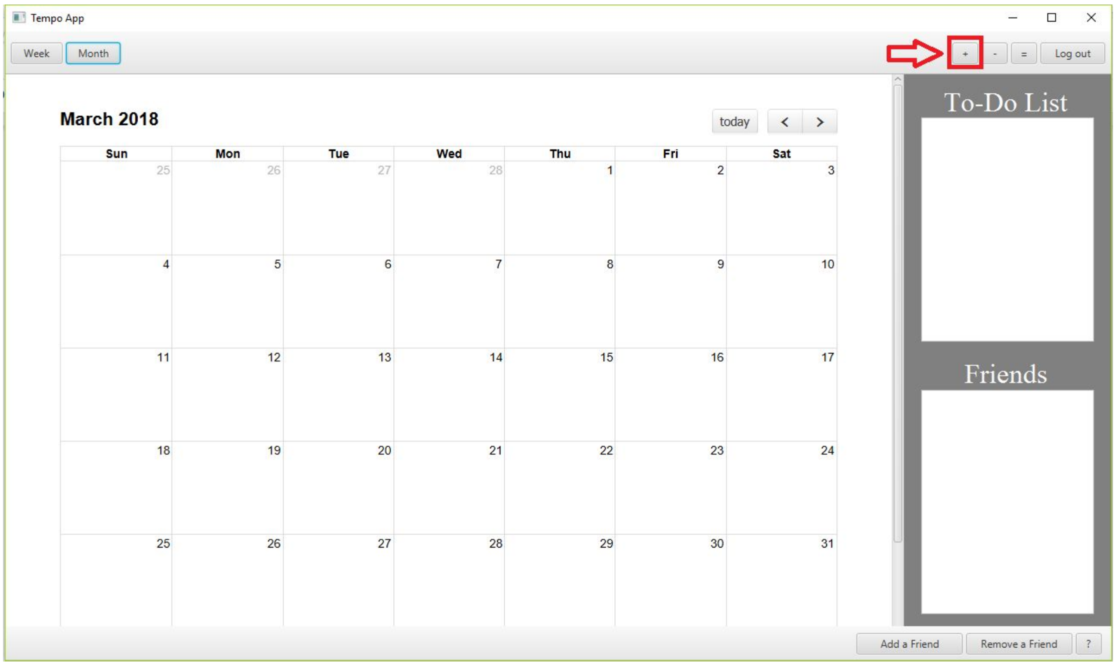
Figure 1 - Authorization Page

After opening the application, the user will see an authorization page. If the user already has a profile in the system, the user can sign in and access to the system by filling “Username” and “Password” in the left side of the authorization page by using a keyboard and then clicking on “Sign in” button.

If the user doesn’t have a profile in the system, the user can sign up to the system by filling the “Username”, “Name”, “Surname”, “E-mail”, “Password”, “Re-Password” in the right side of the authorization page by using a keyboard and then clicking on “Register” button.

While logging in, the user will get a message if the user’s “Username” or “Password” is wrong. Also, while registering, the user will get a message if the user’s “Username” is already taken or the given passwords are not the same.

4.2.3 Adding Events to Calendar



March 2018

Sun	Mon	Tue	Wed	Thu	Fri	Sat
25	26	27	28	1	2	3
4	5	6	7	8	9	10
11	12	13	14	15	16	17
18	19	20	21	22	23	24
25	26	27	28	29	30	31

Figure 2 - Main Page - Add Event Button Highlighted

If the user is logged in to the system, the user will be able to add events from the Main Page. On the right top of the Main Page there are 4 buttons. Clicking the leftmost one will open a pop-up window for adding events.

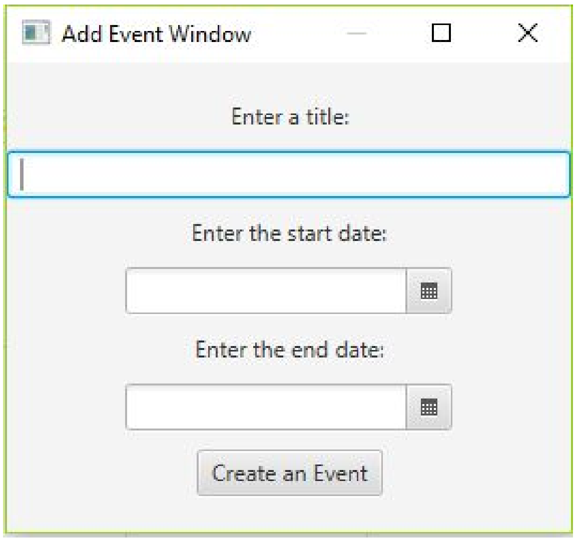


Figure 3 - Add Event Window

After pop-up window opens, the user will be able to give a title to the event and start and end dates for the event. Clicking “Create an Event” will save the event to the System

4.2.4 Removing Events from Calendar

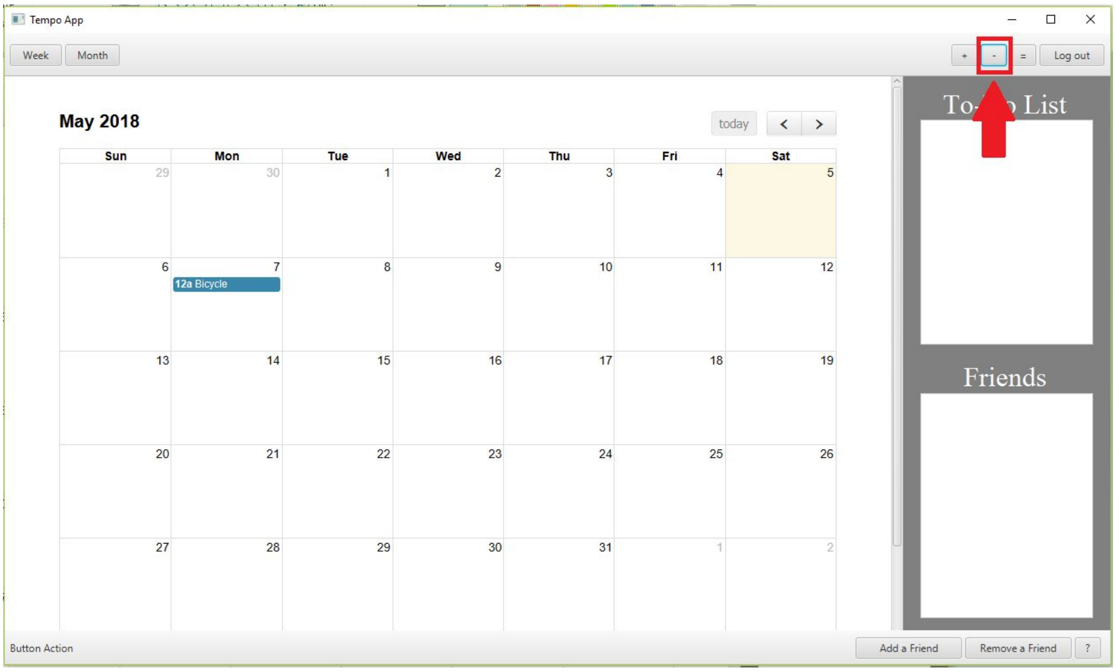
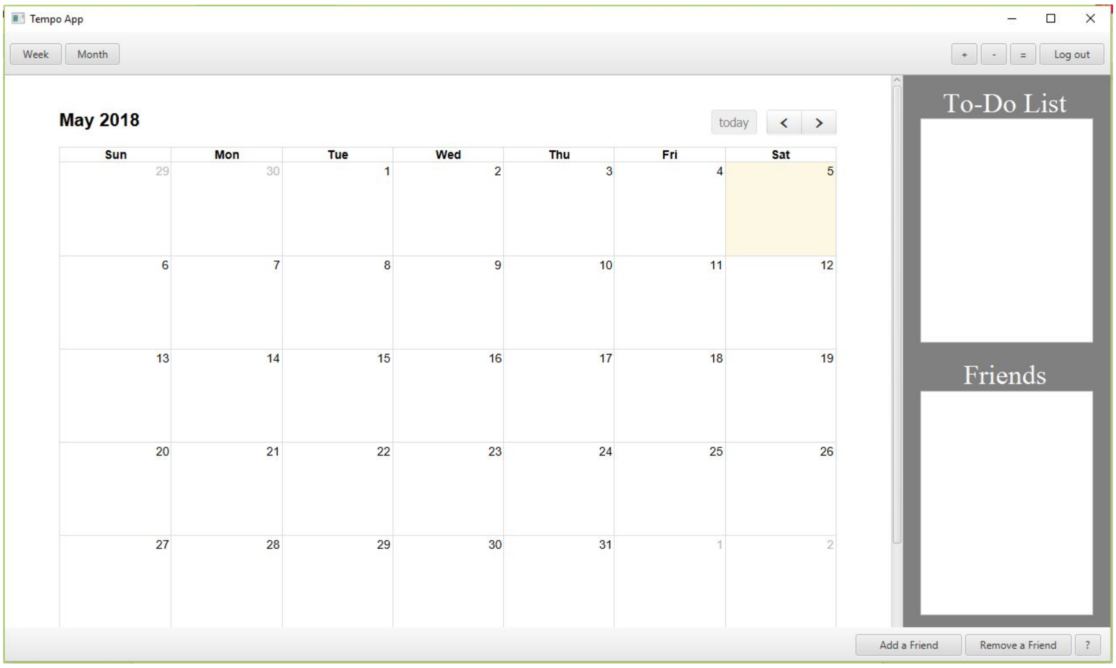


Figure 4 - Main Page - Remove Event Button Highlighted

If the user is logged in to the system, the user will be able to remove events from the Main Page. On the right top of the Main Page there are 4 buttons. After clicking on an event(from calendar or To-Do List), user will click the second from the left to remove the selected event.



May 2018

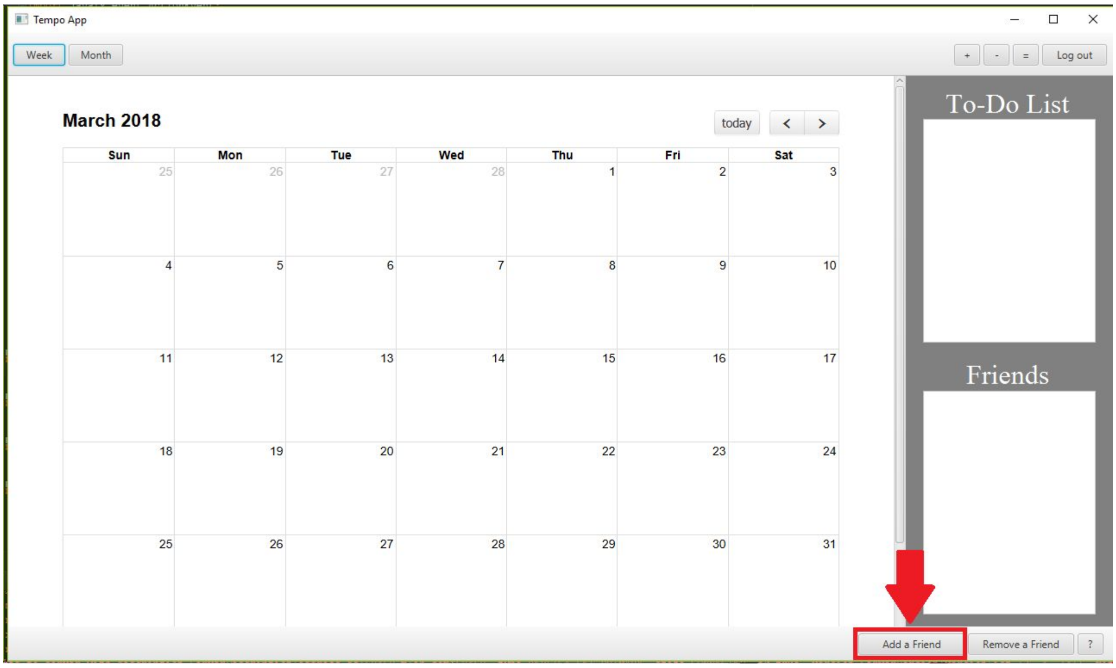
Sun	Mon	Tue	Wed	Thu	Fri	Sat
29	30	1	2	3	4	5
6	7	8	9	10	11	12
13	14	15	16	17	18	19
20	21	22	23	24	25	26
27	28	29	30	31	1	2

Figure 5 - Main Page - Event Removed

After user chooses an event and clicks on the remove event button, the event will be removed from the system.

4.2.5 Adding or Deleting Friends

- Adding a Friend



March 2018

Sun	Mon	Tue	Wed	Thu	Fri	Sat
25	26	27	28	1	2	3
4	5	6	7	8	9	10
11	12	13	14	15	16	17
18	19	20	21	22	23	24
25	26	27	28	29	30	31

Figure 6 - Main Page - Add a Friend Button Highlighted

If the user is logged in to the system, the user will be able to add friends from the Main Page. On the right bottom of the Main Page there are 3 buttons. If user clicks on “Add a Friend” button, a new pop-up window will open.

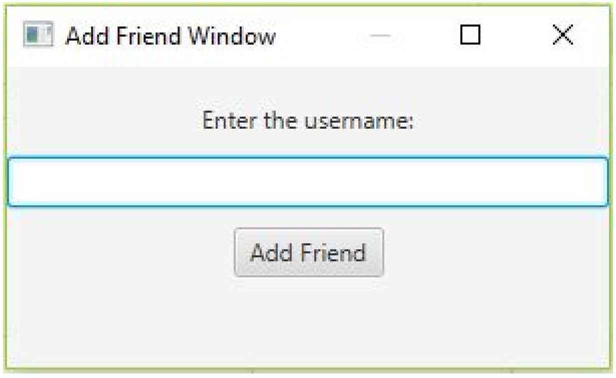
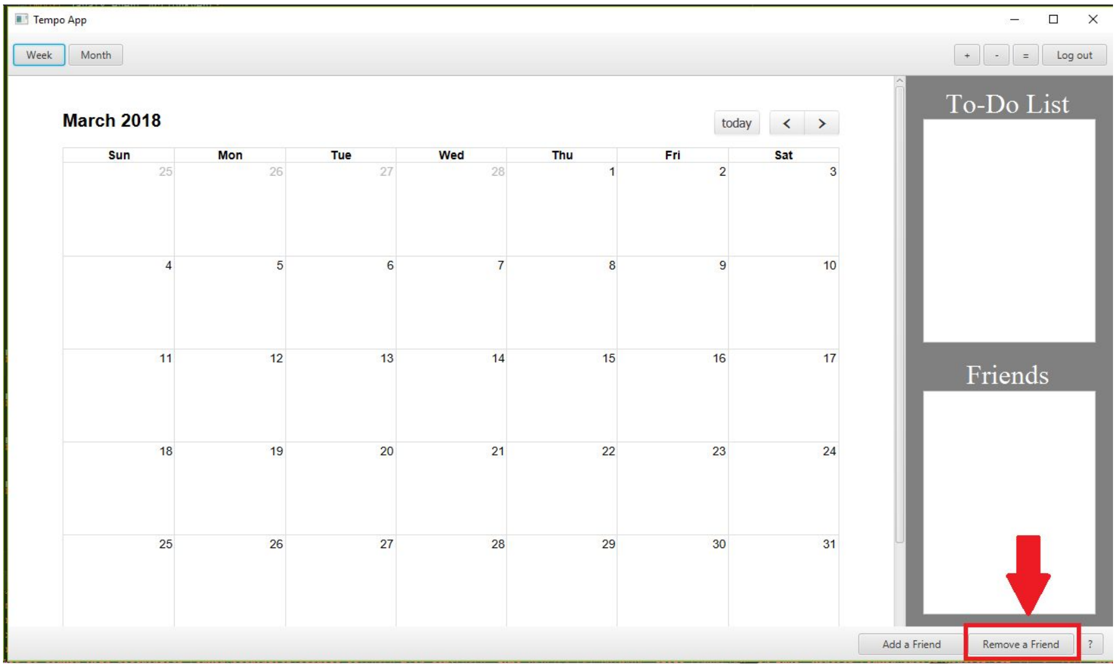


Figure 7 - Add Friend Window

The user has to give exact username for the person that the user wants to add as a friend. After clicking “Add Friend” button, the user will add the person that he wants to add. If not, then the user will be warned about the situation that this username does not exists.

-Remove a Friend



March 2018

Sun	Mon	Tue	Wed	Thu	Fri	Sat
25	26	27	28	1	2	3
4	5	6	7	8	9	10
11	12	13	14	15	16	17
18	19	20	21	22	23	24
25	26	27	28	29	30	31

Figure 8 - Main Page - Remove a Friend Button Highlighted

If the user is logged in to the system, the user will be able to remove friends from the Main Page. On the right bottom of the Main Page there are 3 buttons. If user clicks on "Remove a Friend" button, a new pop-up window will open.

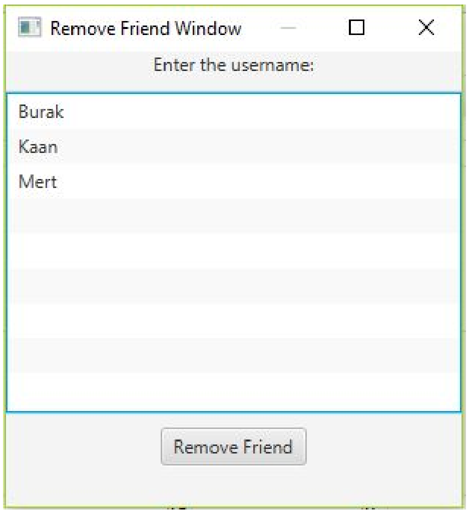


Figure 9 - Remove Friend Window

User will choose a friend then click on "Remove Friend" to remove the friend.

4.2.6 Changing Profile Settings

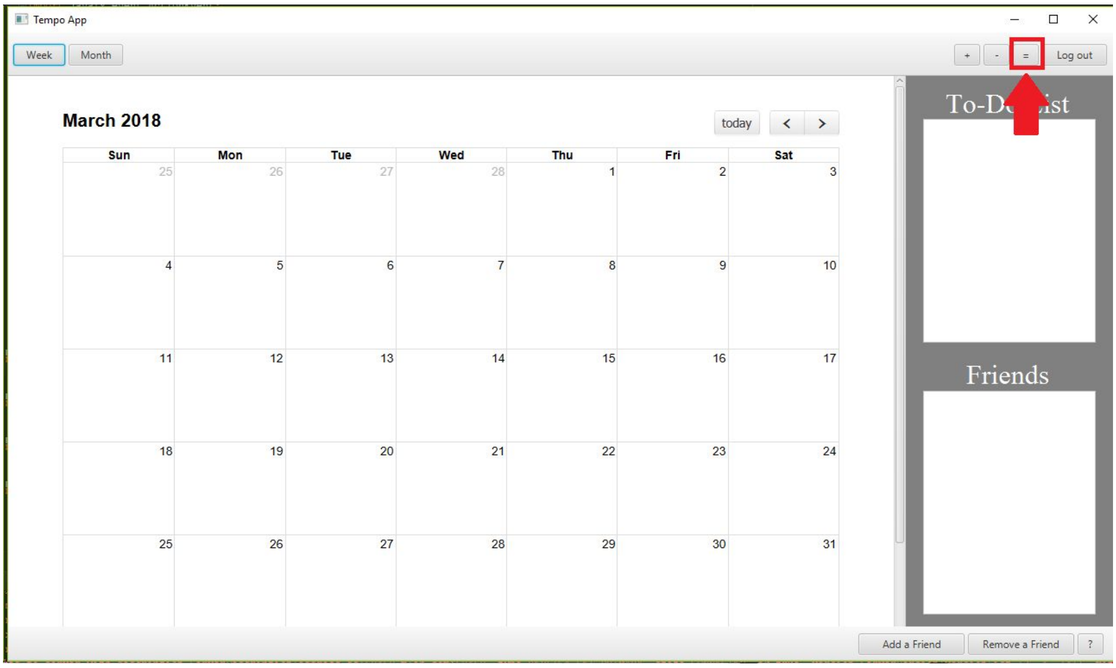


Figure 10 - Main Page - Profile Page Button Highlighted

If the user is logged in to the system, the user will be able to open his profile settings from the Main Page. On the right top of the Main Page there are 4 buttons. Clicking the third from the left will open a pop-up window for adding events.

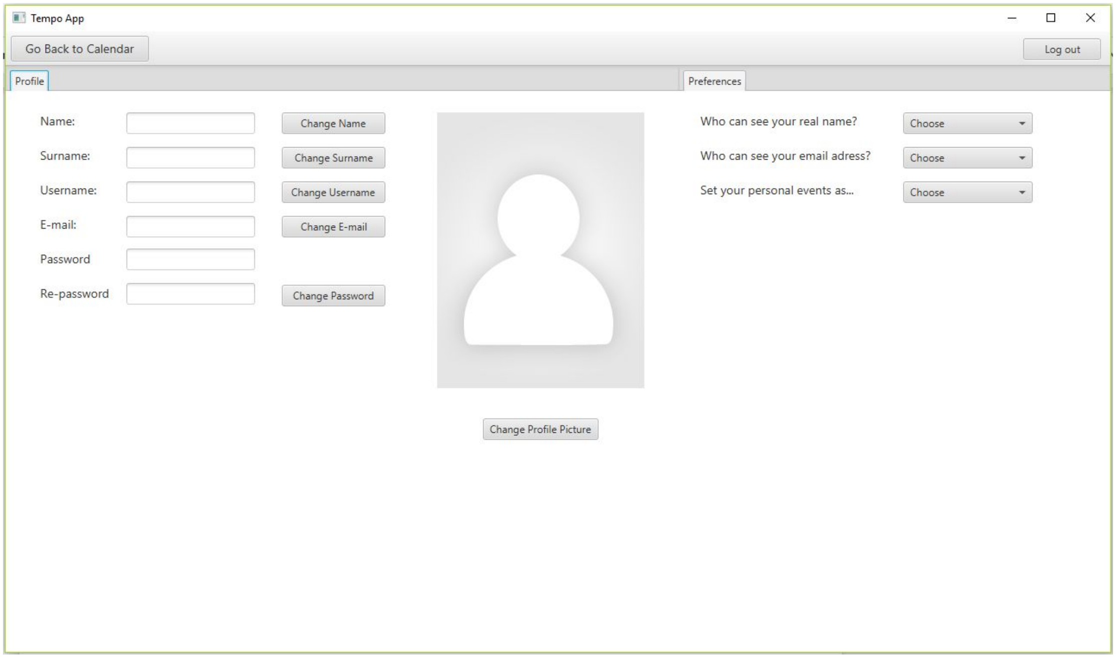
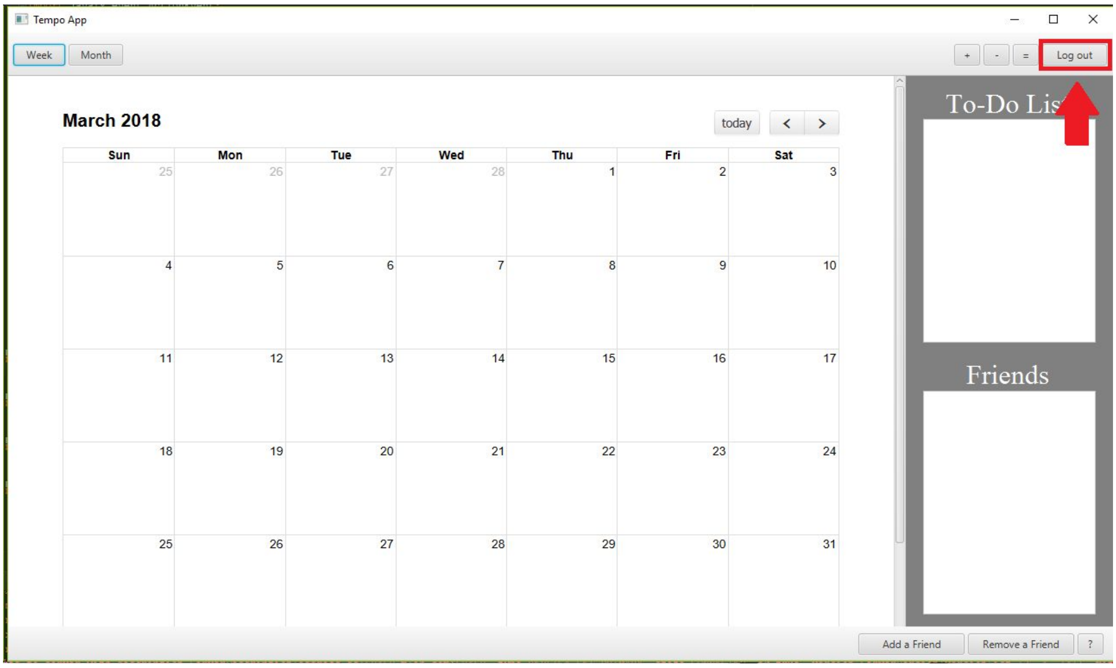


Figure 11 - Profile Page

User will be able to change his profile specifications and preferences.

4.2.7 Logging Out from the System



March 2018

Sun	Mon	Tue	Wed	Thu	Fri	Sat
25	26	27	28	1	2	3
4	5	6	7	8	9	10
11	12	13	14	15	16	17
18	19	20	21	22	23	24
25	26	27	28	29	30	31

Figure 12 - Main Page - Log out Button Highlighted

If the user is logged in to the system, the user will be able to logout from the system from the Main Page. On the right top of the Main Page there are 4 buttons. Clicking the rightmost button will logout the user but it won’t close the Tempo App.

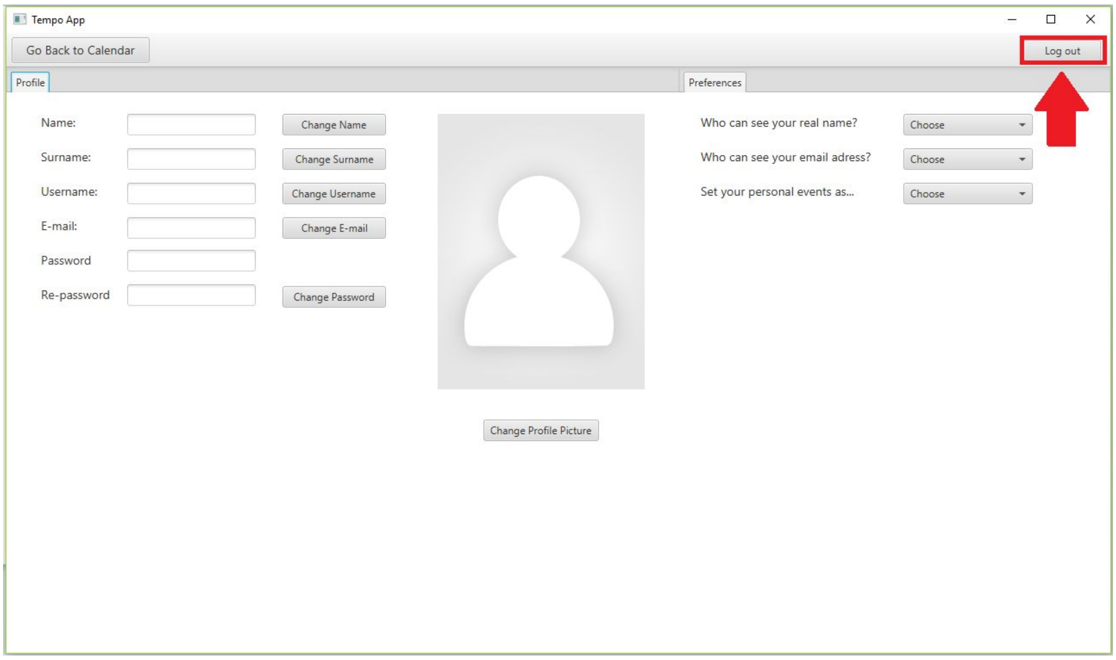


Figure 13 - Profile Page - Log out Button Highlighted

If the user is logged in to the system, the user will be able to logout from the system from the Profile Page. On the right top of the Profile Page there is a “Log Out” button. Clicking the button will logout the user but it won’t close the Tempo App.