PUSS214201

v. 0.5

TimeMate

Software Requirements Specification

Group 2

Responsible: System Group

Authors: System Group, Development Group

2021-02-11

Contents

1   Document History

Version	Date	Responsible	Description
0.1	2021-01-25	SG	Document created.
0.2	2021-02-02	SG	Ready for informal review.
0.3	2021-02-04	SG	Fixed gramatical changes and typos.
0.4	2021-02-09	SG	Requirements added, removed or changed after comments from formal review.
0.5	2021-02-11	SG	Grammatical changes and fixes after comments from informal review.

2   Introduction

This document presents the requirements for TimeMate. TimeMate is a system which purpose and main functionality is to administer time reporting with web-usage capabilities. Figure 1 shows how the system is supposed to interact with the database, the server and the client.

3   Reference Documents

1. Software Requirements Specification: *BaseBlockSystem*, v. 1.0, Doc. number: PUSS12002.

4   Background and Goals

4.1   Main Goals

The goal is to develop and distribute a web-based system where the user can report time and administrate the system according to their roles.

4.2   Actors and their Objectives

The system can be used and administrated by the following actors:

- **User:** The user has the authority to log in to the system to report and change past reported time as well as review their reported times. The users are assigned roles as either "PG", "SG", "UG" or "TG".

- **Project Leader:** The project leaders' main objective is to administer groups within the project. This implies that the project leader has the authority to add and remove as well as assign users from/to designated roles.

- **Administrator:** The administrator has the authority to add and remove users from the system along with creating and removing project groups. The administrator is the only role that can assign the role "Project leader" to a user. The main goal of this role is to be able to administrate addition and removal of users.

5   Terminology

- **UG, SG & TG:** These roles are only name tags for users and grants no special permissions or rights.

- **Administrator:** Has access to all functions of the system.

- **Time report:** A document containing fields detailing time spent on various parts of development.

- **SHA-2:** A set of cryptographic hash functions used to protect users passwords.

- **Salt string:** A salt string is a random string that is used in conjunction with SHA-2 to create safer passwords and protect against attacks.

- **Sign:** An action available to the role "PG" which makes documents uneditable by users.

- **Unsign:** An action available to the role "PG" on signed documents which makes them editable again.

6   Functional Requirements

The requirements of *BaseBlockSystem* applies to this document and are therefore implemented in TimeMate. The following requirements will *not* be implemented; 6.2.3, 6.3.4

6.1   Login and Logout

6.1.1   Requirement

If the user requests a new password, they should receive a new, automatically generated password from the server to the email connected to their username. The user's former password is then deleted from the database and overwritten with the new password.

6.2   Data

6.2.1   Requirement

A password should consist of at least one character of each type: [a-z][A-Z][0-9] with a minimum of 8 characters. This replaces requirement 6.2.3 in *PUSS12002*, v1.0

6.2.2   Requirement

Passwords should be encrypted using SHA-2 and include a randomly generated salt string in the encryption process.

6.3   User

6.3.1   Requirement

The following scenario should be supported by the system.

Scenario: The user wants to create a new time report and submit it.

Prerequisites: The user is logged in to the system.

1. The user navigates to the page "Time Report".

2. The user is presented with an overview of their previously reported time.

3. The user presses the button "Create New Time Report"

4. The user fills in the blank text boxes with the time in minutes they have spent on their different time reports.

5. The user presses the button "Submit".

6. The time report is saved to the server and the user is presented with a popup window confirming that the changes have been made.

7. An updated view of the page is displayed.

6.3.2   Requirement

The following scenario should be supported by the system.

Scenario: The user wants to see a summary of their reported time.

Prerequisites: The user is logged in to the system

1. The user navigates to the page "Time Report".

2. A page is displayed which includes the option to report time, edit old time reports but also review a summarized view of their previously logged time.

6.3.3   Requirement

The following scenario should be supported by the system.

Scenario: A user wants to change previously reported time.

Prerequisites: The user is logged in to the system.

1. The user navigates to the page "Time report".

2. The user is presented with an overview of their previously reported time.

3. The user chooses a time report.

4. The user presses the button "Edit time report".

5. The user is presented with a new page which includes the previously reported time table.

6. The user fills in the new time.

7. The user clicks on the button "Submit Change".

8. The system is updated with the new time report.

9. An updated view of the page is displayed.

6.3.4   Requirement

The following scenario should be supported by the system.
   Scenario: A user wants to change their password.
   Prerequisites: The user is logged in to the system

1. The user clicks on the "Change Password" button in the menu.

2. The user is presented with a page which includes three relevant text boxes.

3. The user fills in the first text box with their current password.

4. The user fills in the second text box with their new password.

5. The user repeats their new password in the third text box.

6. The user clicks the button "Change Password".

7. The user's password is changed in the system.

8. The user is presented with a popup window confirming that the changes has been made.

6.3.5   Requirement

The following scenario should be supported by the system.
   Scenario: A user wants to change their password with an invalid entry.
   Prerequisites: The user is logged in to the system

1. The user clicks on the "Change Password" button in the menu.

2. The user is presented with a page which includes three relevant text boxes.

3. The user fills in the first text box with their current password.

4. The user fills in the second text box with their new password which does not meet requirement 7.2.1 from this document.

5. The user repeats their new password in the third text box.

6. The user clicks the button "Change Password".

7. The user's password is not changed and a popup window with an error message is displayed.

8. The user is sent back to stage 2.

6.3.6   Requirement

Users should be able to edit unsigned time reports submitted by them

6.3.7   Requirement

Users should not be able to edit signed time-reports submitted by them.

6.4   Project Leader

6.4.1   Requirement

The following scenario should be supported by the system.
   Scenario: The project leader tries to acquire a summary of reported time.
   Prerequisites: Time to be summarized exists, user has the role "Project leader" and the user is on the "Time Reporting" page.

1. The project leader clicks on the "SHOW SIGNED REPORTS" button.

2. A list with user's name and the total reported time will be displayed,

3. The project leader clicks on a user name and it displays the user's time reports.

6.4.2   Requirement

Project leader should be able to view all project members and their individual roles on the "User Management" page in the form of a list.

6.4.3   Requirement

Project leaders should be able to change a project members role except themselves and other project leaders.

6.4.4   Requirement

Project leaders should be able to *sign* time reports by clicking on the corresponding checkbox on the page displaying time reports.

6.4.5   Requirement

Project leaders should be able to *unsign* time reports by clicking on the corresponding checkbox on the page displaying time reports.

6.4.6   Requirement

The following scenario should be supported by the system.
   Scenario: Project leader signs a weekly report
   Prerequisites: User has the role "Project leader", weekly reports exists

1. The project leader clicks on the "Sign Time Reports" button.

2. A list of all the unsigned weekly reports is shown to the project leader.

3. The project leader can sign the reports by clicking on the corresponding checkbox.

4. The project leader confirms their selection by pressing on the "Confirm" button

5. The system sends an update to the database, changing the "Signature" column in the "Time Report" table to "signed".

6. The system returns an updated list of unsigned reports, or an empty list if none exists

6.4.7 Requirement

The following scenario should be supported by the system.

Scenario: Project leader assigns roles to members.

Prerequisites: User has the role "Project leader", members exist and the user is on the "User Management" page.

1. The project leader can change members' roles by clicking on the corresponding radio button.

2. Project leader confirms changes made by pressing "Confirm".

3. The system updates the "role" field in the database for the selected user.

6.5 Administrator

6.5.1 Requirement

Only the administrator should have access to the administration page.

6.5.2 Requirement

The following scenarios should be supported by the system.

Scenario: The administrator adds a new user.

Prerequisites: The administrator is on the "Administration" page.

1. The administrator clicks on the button *add a user.*

2. The administrator types in the username of the new user in an empty text field marked with the text *User Name.*

3. The administrator types in the email address of the new user in an empty text field marked with the text *Email Address.*

4. The inputs are sent to the server which adds it to the database.

5. A random password is generated automatically and gets associated with the new user.

6. The system creates a new entry in the user list and inserts the information about the new user.

7. The automatically generated password is returned to the corresponding entry in the user list, where it is saved.

8. The password is sent to the email address associated with the new user.

9. An updated view of the user list is displayed.

6.5.3   Requirement

The following scenario should be supported by the system.
  Scenario: The Administrator tries to add a user with the same name.
  Prerequisites: The administrator is on the "Administration" page.

1. Administrator clicks on the button to add a user.

2. Administrator inputs the name of the user.

3. The inputs are sent to the server, which notices that the name already exists in the database.

4. The server does not accept the inputted name.

5. Administrator is told that the user already exists with a message that shows up.

6.5.4   Requirement

The following scenario should be supported by the system.
  Scenario: The administrator tries to remove a user.
  Prerequisites: The administrator is on the "Administration" page.

1. Administrator clicks on the button to remove the user.

2. Administrator must confirm the removal by typing in the user's name and click on the "confirm" button.

3. The information is sent to the server.

4. The user is removed from the database.

5. The page is updated with the new user list, with a message confirming the change.

6.5.5   Requirement

The following scenario should be supported by the system.
  Scenario: The administrator tries to acquire a summary of all reported time.
  Prerequisites: Time to be summarized exists.

1. Administrator clicks on the "SHOW SIGNED REPORTS" button.

2. A list with the names of all the users and their total reported time will be displayed.

3. Administrator clicks on a username and it displays the user's time reports.

6.5.6   Requirement

The following scenario should be supported by the system.

Scenario: The administrator tries to assign roles to user.

Prerequisites: The administrator is on the "User Management" as defined in requirement 7.0.6. A page and user exist.

1. Administrator can change users' roles by clicking on the corresponding radio button.

2. Administrator confirms changes made by pressing "Confirm".

3. The system updates the "role" field in the database for the newly selected user.

6.5.7   Requirement

The administrator should be able to view all user information in the database in the form of a list on the administration page.

7   Design Requirements

The requirements of *BaseBlockSystem* applies to this document and are therefore implemented in TimeMate. The following requirements will *not* be implemented; 6.2.3, 6.3.4.

7.0.1   Requirement

The main font should be web safe and belong to one of the following font families: Serif or Sans-serif.

7.0.2   Requirement

The main text should have a font size of 16 pixels (+-2 pixels). All main text should be of the same size. Textfields and textfield captions should have a font size of at least 16 pixels.

7.0.3   Requirement

The color scheme should consist of three primary colors. These should, if possible, follow the 60/30/10 design rule. Secondary colors are permitted only when dealing with less important information. The background should consist of a lighter color and the main text should consist of a darker shade.

7.0.4   Requirement

The content of the web page should be centered in the browser window. The content should keep its centered position when the window is extended.

7.0.5   Requirement

A fixed navigation/menu bar shall be placed horizontally at the very top of the page.

7.0.6   Requirement

Selections *Main-page*, *Administration*,*User Management* and *Time Reporting* should be included in the menu bar. Other menu selections should be categorized as subheadings belonging to one of the primary menu choices and should not be visible or clickable by default. The administration page should only be visible to the administrator.

7.0.7   Requirement

When a menu selection containing at least one subheading is selected, a vertical menu bar containing its subheadings should be revealed. A downward pointing arrowhead,$\vee$, should be placed beside all menu selections containing at least one subheading. When a menu selection containing one or more subheadings is selected, the arrowhead should point upwards.

7.0.8   Requirement

The *Main-page* should be the default page the user sees after logging into the system.

7.0.9   Requirement

The *Administration* page should support the functionality of requirement 7.5.1, 7.5.2, 7.5.3, 7.5.4 , 7.5.5, 7.5.6 from this document.

7.0.10   Requirement

The *User Management* page should support the functionality of requirement 7.4.2 and 7.4.7 from this document.

7.0.11   Requirement

The *Time Reporting* page should support the functionality of requirement 7.3.1, 7.3.2, 7.3.3, 7.3.6, 7.3.7 and 7.4.1 from this document.

7.0.12   Requirement

The system logo and/or the system name should be placed on the far left of the menu bar.

7.0.13   Requirement

The logout button should always be visible when the user is logged into the system.

8   Project Requirements

8.1   Development Environment

8.1.1   Requirement

The back end of the system should use Java JDK 11.0.10 as a development environment.

8.1.2   Requirement

The front end of the system should use Bootstrap v.4.5.3.

8.1.3   Requirement

The database should be developed in MySQL v.10.4.17 (MariaDB).

8.1.4   Requirement

The database should be accessed through a VPN connected to LTH's network.

8.1.5   Requirement

The system should use Apace Tomcat 9.0.33.

8.1.6   Requirement

Error messages should be in the following format: "Error code – Reason".