2018-2019 About Time Design Documentation

Documenters:

Matthew Kouzios, Eduardo Aviles

Overview

Project Summary

The About Time project exists to give students a way to view their work progress, generate a visual report on student work progress for documentation purposes, and to give instructors a way to view student progress. The project may eventually go to the plugin marketplace for global use, and if so the student/instructor roles would just be condensed into a general user role.

Document Purpose

The purpose of this document is to provide information about each part of the About Time project, specifically for usage if the project becomes repurposed for another website such as GitLabs. This document will cover all components of the plugin, and detail what features/designs are required for each component. It will also give suggestions on design decisions, but the final say goes to the product owner.

Homepage

The homepage is focused on providing the user the ability to navigate between features, learn about the project, and report issues found in the project.

Navigation

The homepage allows users to go to their desired feature, with the intention that each individual feature can be navigated to via the homepage. This allows users to have a clear understanding of how to get to the feature they need, meaning they can go back to the homepage and find all the features from there.

Information

The homepage should also provide information about how to set up the plugin as a shortcut, as well as information on the plugin and information on the team(s) working on it. The plugin shortcut setup tells the user how to add the About Time plugin to the website's MSOE-specific navigation bar; however, this may not be applicable to non-Jira websites. The team information section gives a brief overview of what features the project offers, as well as a small explanation on why the project was created.

Issue Reporting

While issue reporting does not need to explicitly be on the homepage, it should be an easy-to-access feature that allows users to submit issues they come across. The reporting should be directed to either the development team or the product owner so that the issue can find its way into the proper hands to fix it. Any user may submit an issue; however, it may be prudent to limit how many submissions may be made in a certain time period in the event of a malicious user.

Filtering

In order to display project data, the project requires either a date range or sprint for a given project. This is done through filtering, where the user selects a time period and a project and then gets the project data. Due to there being multiple features that use this information, the filtering feature can be placed on the homepage. However, not every feature will end up using the filtering information (see: Dashboard) thus another solution may be prudent. When selecting a project, the drop down should break into two sections: recent projects, and alphabetical. The topmost section is the recent projects section, where the most recently worked on projects by the current user are displayed. The alphabetical section displays all projects, including those in the recent projects section, in alphabetical order.

Time Logs

The time logs page is dedicated to displaying user's work logs visually both through a time table and a work log table. These tables are included so that users can check their work progress for their desired time period in a way that is condensed but visually appealing.

Time Table

The time logs page will have a time table to reflect the team's hours logged along with work logs for the day. As shown in Figure 1 below, the time table must have the team members listed, the daily hours logged for each user in the project, the summed daily hours for all users in the project, the summed daily hours for all users in the project, and the summed weekly hours for all users in the project.

Figure 1: The current time table implementation

Each day in the table is also clickable to show that days work description along with unusual logging if there was any, see Figure 2 below for an example. The days with unusual logging have their logged hours in red. The application should detect these issues in the users time logs, and specify them as unusual logs. In addition, the logging should specify what entries in the worklogs table are causing the issues. The issues the application should check are when:

- 4 or more hours are logged in a row by the same user
- 10 or more hours are logged in a day (including across multiple projects)
- A work log description is missing
- Logged times are overlapping
- Time was logged to a story at a time before the story was created (usually done by doing an action, creating a story for it, then logging time for the action done earlier in the day)
- Time was logged two or more weeks after the work was done

An example of the worklogs popup is shown below, in Figure 2.

Figure 2: This is an example of the work description after a day is clicked

Descriptions

The time logs page should also include the work descriptions from the specified time period for the entire team. It should include the status, time spent, and the work description. This can be ordered from the most recent to least recent or whichever other way is seen fit. An example is provided in Figure 3 below. However, a different design can be implemented to reflect the product owner's criteria.

Figure 3: This is an example of the descriptions portion for the time log page

Emojis

FOR INDIVIDUAL USERS:

The emojis can be found in the "*custom.pug*" file. Let's say x = the user's/team member's actual hours that they've logged. The way it's currently set up is that if x < 50% of their expected hours, it will print . If 50%<= x < 70% of their expected hours, it will print . If 70%<= x < 100% of their expected hours, it will print . If x >= 100%, it will print . If you reference the emoji key on the Time Table page ("View Logs" button), you'll see the time log constraints.

These emojis and their time constraints can be found around line 324 of the "*custom.pug*" file. The emoji that's selected is selected via the use of simple if statements. You can just go in and alter these if statements to either change the time constraints and/or the emojis that are displayed. Also, the emojis stand out seeing that they're surrounded by text, so locating them should be easy. There are also emojis (the same ones) at the bottom of the "*custom.pug*" file, but that's for the whole team as opposed to the individual team-members. The next section will cover emojis for the whole team.

FOR THE WHOLE TEAM:

These emojis are also in the "custom.pug" file, but they're down towards the bottom of the file. You can find the logic for this around line 452 where you'll see a line of code that says "tr(class="healthrow")" where tr stands for table row. Basically, it creates the row that will display the overall team's health. The logic underneath it handles the time log constraints and the emojis that are used. This is pretty much the exact same as the individual user - reference the emoji key on the Time Table page ("View Logs" button) to see the time time log constraints for the overall team. These time log constraints and the specific emojis that are displayed can be changed here just like you did above for the individual team members; you can just alter the if statements and the emojis to suit your needs.

Status Report Generation

The Status Report Generator is a feature that allows teams to view a summary of a chosen date range or sprint for a given project. If the team has a documentation page, they should be able to publish it there as well. The report should at least include the team's sprint goal, burndown, and individual contributions.

Heading

The heading of the sprint report should include the specified sprint along with the dates that are being covered in the report. The heading should also include the sprint goal and burndown chart/table. The burndown chart/table should have a reflection of each individual members hours. The burndown chart was attempted by one of the previous teams but was abandoned. It isn't pertinent but would be useful to have.

Body

The body of the status report should be filled with the work logs of each individual team member. The work logs should be separated into stories, bugs/defects, internal improvements, or overheads. The work logs should be further categorized into specific stories with their respective status, and then followed with the subtasks and their respective status. The subtasks are then followed by the time spent, the work description of the individual, and any further comments for the individual. An example is provided below in Figure 4.

Figure 4: Example of the status report generator body feature

Editing

Some members of a team may finish their work before others and wish to make statements about their work logs. However, if they do so and then publish the report, the work logs of other members of their teams may not be accurate. Therefore there should be a way for a user to add comments to their status report and save those comments without needing to generate the status report.

Dashboard

The dashboard feature is dedicated to showing multiple projects in one location, with time graphs and tables to present the project data in a condensed manner. The page will be primarily used by professors or those in a similar role as a way to quickly preview the work progress of multiple teams.

Filtering

As the dashboard is more of a standalone feature from the rest of the project, it should have the ability to filter projects and time periods for display. At a minimum, it should be able to sort by current sprint or a custom date. It's imperative to note that the current sprint may differ based on the project, so there should be validation to ensure each project displays the current sprint even if the dates for those sprints encompass different time periods. See Figure 5 below for an example.

Figure 5: The time period navigation bar

Projects Portfolio

The dashboard needs a way to allow a user to view multiple projects at a time as this feature is not introduced in the homepage. This can be done in any manner of ways, but the selected projects should be saved so that that user does not need to re-select them if they intend to use the dashboard feature in the future. In addition, the saved projects should be able to be forgotten, in the event the user does not wish to see the worklogs for that project anymore.

Time Graph

This feature displays the times of the team over the specified time period as a line graph, allowing for a clean visual view for user time comparisons. It is advised to allow users to select a position in the graph to see the hours logged at the clicked location for user convenience. See Figure 6 below for an example time graph.

Figure 6: Time Graph over the period of November

Time Table

For each project being stored, the dashboard should display a time table relating to that project and the time period selected. For more information on the time table, see the Time Table paragraph under the Time Logs section.