



Bilkent University Department of Computer Engineering

Senior Design Project

Project Name: Cronus

Project Specifications Report

*Group Members: Gizem Karal, Gökberk Boz, Hüseyin Fatih Karahan,
Irmak Çeliker*

Supervisor: İbrahim Körpeoğlu

Jury Members: Shervin Arashloo and Hamdi Dibeklioglu

Project Specifications Report Oct 11, 2021

This report is submitted to the Department of Computer Engineering of Bilkent University in partial fulfillment of the requirements of the Senior Design Project course CS491.

Table of Contents

| | |
|-----------------------------------|----------|
| Introduction | 3 |
| 1.1. Description | 3 |
| 1.2. Constraints | 4 |
| 1.2.1. Implementation Constraints | 4 |
| 1.2.2. Sustainability Constraints | 4 |
| 1.2.3. Economical Constraints | 4 |
| 1.2.4. Ethical Constraints | 4 |
| 1.2.5. Social Constraints | 5 |
| 1.3 Similar Technologies | 5 |
| 1.3.1 Jira | 5 |
| 1.3.2 Asana | 5 |
| 1.3.3 Trello | 5 |
| 2. Requirements | 6 |
| 2.1. Functional Requirements | 6 |
| 2.1.1. Client-side Web App | 6 |
| 2.1.2. Backend Server | 7 |
| 2.1.2. Admin Webapp | 7 |
| 2.2 Non-functional Requirements | 7 |
| 2.2.1 Usability | 7 |
| 2.2.2 Reliability | 8 |
| 2.2.3 Localization | 8 |
| 2.2.4 Availability | 8 |
| 2.2.5 Security | 9 |
| 2.2.6 Performance | 9 |
| 3. References | 9 |

1. Introduction

Project management is a very significant step during the creation of a software product. Especially in team projects, it can be very challenging to keep track of the project, distribute and order the tasks due to the number of people working in the project. Along with the fact that it can be hard to carry out a project with many different developers, due to the COVID-19[1] pandemic it has become even more complicated to keep track of others' work since the project management is also needed to be carried out on digital platforms. Our aim is to solve these kinds of complications and give relief to software developers during their project creation process by developing a project management tool, Cronus, for software developers.

Moreover, besides the complications that arise due to the number of developers in a project, it can also be challenging for people to keep track of their own process and work. Cronus will eliminate these problems and create a neat work environment for their users.

Every project in addition to normal tasks also has some repetitive tasks that need to be tracked. While Kanban[2] tools like Jira[3] or Asana[4] allow the tracking and ordering of issues, they become cumbersome very quickly when dealing with tasks that need to be repeated over intervals. Usually this is dealt with creating a new issue during each sprint or whenever it's needed. Our project management tool instead focuses on the management of tasks that repeat.

1.1. Description

Cronus is going to be a project management webapp for software developer teams that will be available online. Users will be able to plan the project, release their works and track each others' work on this platform that will create a neat environment for their project management process with the help of Cronus. While users will be able to track their process, they will also be on top of things by tracking others' progress and work. Moreover they will be able to split the work between the teammates and prioritize the tasks.

As mentioned before, different from similar products that offer the same qualities, Cronus will be able to be used for repetitive tasks that need to be tracked and eliminate the complexities that arise in similar technologies.

1.2. Constraints

1.2.1. Implementation Constraints

Cronus will be developed as a webapp.

Java will be used to handle the back-end, JavaScript and PHP will be used for the front-end.

GitHub will be used for joint development and Jira for work-share and project tracking.

1.2.2. Sustainability Constraints

To increase the usability and prevent bugs and crashes, the user feedback will be considered highly.

1.2.3. Economical Constraints

Cronus will be a free webapp for all the users and companies.

Open-source libraries and APIs, if needed, will be used.

Continuous servers may need a cost.

1.2.4. Ethical Constraints

An authentication system will be used so that unauthorized people will not be able to see the personal content.

Developers or any other third party company will not be able to view the stored information.

1.2.5. Social Constraints

In order to be used by anyone without any difficulty, the interface will be user friendly and the application will be in English.

1.3 Similar Technologies

There are many different similar technologies present, however the listed applications below are the most preferred ones among the users.

1.3.1 Jira

Jira is a project management tool that offers ease to software developers. It allows teams to plan, assign, track, report, and manage work[3] similar to our application. However it is not suitable for repetitive tasks unlike Cronus which will mainly focus on repetitive tasks.

1.3.2 Asana

Asana is also a project management tool, it offers distributing, organizing, tracking and managing timeline[4]. It is a more general application that does not focus on software developers unlike Jira. Cronus will be used for general purposes like Asana and will be able to synchronize across the company.

1.3.3 Trello

While Trello has the same functionalities with the other applications, it lacks in workflow functionality and like Asana it is not specific for software developers. It does not offer labeling and identifying a task[5], which will be present in Cronus.

2. Requirements

2.1. Functional Requirements

2.1.1. Client-side Web App

- The application should allow users to sign up and login.
- The application should allow users to subscribe and/or change their subscription for the system.
- The application should allow users to create a team to manage the users that will be using the system for their entity, such as a company or a school.
- The application should allow team creator(s) to create roles with different permissions and assign them to the members of their team.
- The application should allow users to create a project.
- The application should allow users to create a board under a project to manage different parts of the project.
- The application should allow users to create tasks that need to be completed in specified intervals.
- The application should allow users to add information to a task, such as a title, a summary, priority etc., or a custom field defined for the project.
- The application should allow users to create different labels for tasks on a per board, per project or per team basis and assign them to actions.
- The application should allow users to assign one or multiple users to a task and define if any given interval will be considered completed if it's done by one of the assignees or if it needs to be done by every assignee.
- The application should allow users to watch a given task to be notified whenever there's a change done to the task.
- The application should allow users to define a prerequisite to a task that needs to be completed before that task can be completed.
- The application should allow users to search, filter and sort through all tasks, boards, and projects they have access to.

- The application should allow users to view the log of all changes done by the users.

2.1.2. Backend Server

- The system should store all tasks, boards, projects, teams and all associated information.
- The system should log all the changes done by the users.
- The system should notify users when there's a change in the task they're watching.
- The system should notify users when they're assigned to a task by someone other than themselves.
- The system should process payments for subscriptions and send invoices following the team's subscription schedule.

2.1.2. Admin Webapp

- The admin application should allow admins to view and edit all and past subscriptions.
- The admin application should allow admins to view all the statistics about the financial data about the application.

2.2 Non-functional Requirements

2.2.1 Usability

The most important non-functional requirement of the application is usability, since it focuses on keeping track of users' process and work. The application targets to create a neat work environment for its users. To achieve user-friendliness in our design the following points are considered:

- The main purpose of the application is to present a neat environment for its users in a team, in order to track and order the issues in the project. The application should present the workload and project plan in a clear display with its user friendly UI. To ensure this point is satisfied,

we plan to use a well defined user interface system such as Material or Ant.

- It should be straightforward for users to add information/comments to achieve user-friendliness. To manage that, users will be able to fill in feedback at the end of the project.

2.2.2 Reliability

The system should be composed of highly reliable functions with a similar efficiency after extensive use. To achieve that, during the testing process, the application's critical failures will be checked and calculated. It gives the information of the system according to the failures. If the number of failures is low, that means the system is reliable.

Since the system provides continuous information to users during the project, the system should be robust, which means it should handle crashes in the software with a high level of control and efficiency.

2.2.3 Localization

The application has a localization non-functionality which is pretty important from the user's perspective. A project may consist of members who are from different countries. It leads to different time zones for the different geographical locations.

In order to handle it, the system will have aspects such as time zone. The time zone has to be calculated according to each member's location and schedule will be shown in this sense.

2.2.4 Availability

During the project, everyone has their own schedule which leads to different working hours. The users should be able to access the application at any time they want, some may choose the morning scale while the other ones choose to work in the evening.

2.2.5 Security

To protect the sensitive data of the user, the application provides password generation and security question answering:

- The application does not grant access until the user creates a strong password. For example, a strong password might contain a certain number of characters and a capital letter [6].

2.2.6 Performance

The application will respond to its users in a quick way showing which tasks are completed and will be repeated after finishing it. In order to manage that, the users should be notified with real-time notifications in a fast way within 1 second at most.

3. References

- [1] "Coronavirus Disease(COVID-19)," Google. [Online]. Available: <https://www.who.int/emergencies/diseases/novel-coronavirus-2019>. [Accessed: 11-Oct-2021].
- [2] "What is Kanban?," Google. [Online]. Available: <https://www.digite.com/kanban/what-is-kanban/>. [Accessed: 11-Oct-2021].
- [3] "Jira Software," Google. [Online]. Available: <https://www.atlassian.com/software/jira/guides/getting-started/overview>. [Accessed: 11-Oct-2021].
- [4] "Asana," Google. [Online]. Available: <https://asana.com/>. [Accessed: 11-Oct-2021].
- [5] "Trello vs. Asana: Battle of the Freemium Project Management Tools," Google. [Online]. Available: <https://www.cmswire.com/digital-workplace/trello-vs-asana-battle-of-the-freemium-project-management-tools/>. [Accessed: 11-Oct-2021].
- [6] "9 Non-Functional Requirements" Google. [Online]. Available: <https://www.indeed.com/career-advice/career-development/non-functional-requirements-examples>
- [7] "What are Non-Functional Requirements- with Examples?" Google. [Online]. Available: <https://www.perforce.com/blog/alm/what-are-non-functional-requirements-examples>