Constructor University Bremen

Project Management Software Requirements

February 18, 2025

Team:

Bricked Up

Author:

Nikolay Lachezarov Tsonev

1 Functional Requirements

The final product is a general project management software that will help organisations initiate, plan, budget their projects. The following is the current list of **functional requirements**. Note that these may be subject to change.

1. Users should be able to login or register to the platform.
2. Users should be able to log out and delete their account.
3. Each account must be unique and identified by a valid email address.
4. Users should be able to change their passwords.
5. Users should be able to create: organisations, teams and projects.
6. Clear distinction in user privilege as follows from highest to lowest {Admin, Project Lead, Team member}
7. Any team member should be able to create a project and invite other org. members to said project.
8. Project manager should be able to promote other to PM and be able to remove members from the team.
9. If the PM leaves with no other PMs in a given project, the org. admin shall become the PM.
10. be able to schedule tasks
11. Each task should have the following: {budget, person(s) working on it, scope, priority}
12. Each task can be prioritised from: {High, middle, low}
13. Software should allow a team member to have more than one task at a time even if overlapping.
14. Users should be able to customise the software in respect to color, themes and font from a given selection.
15. Software is web-based and should be able to run on most browsers: {Firefox, Safari, Google Chrome, Microsoft edge}.

16. Software should remind team members who have high priority tasks via Email or other forms of communication.

17. Project encapsulation should exist where org. members that are not in the project do not have read/write access unless directly granted by the admin/PM.

18. Software will have a light and dark mode.

2 Non-functional Requirements

These requirements are more towards the backend and features that the users may not directly experience but are crucial for the design and development of secure, scalable, maintainable and economic software.

1. Data privacy and not allowing unwanted or malicious users to access sensitive/unauthorised data.

2. The software must be dependable and must be resilient and minimise failure and have graceful failure if failure cannot be avoided.

3. The software should be efficient, scalable and have low overhead

4. The frontend should be designed and implemented in a structured modular manner for easier expansion and re factoring.