TEAM PROJECT

SUBMISSION 2

Bogdan Gheorghe

2329324

Agile Estimation of Cards :

- **Notification System : Inbox Interface(12 Hours)**

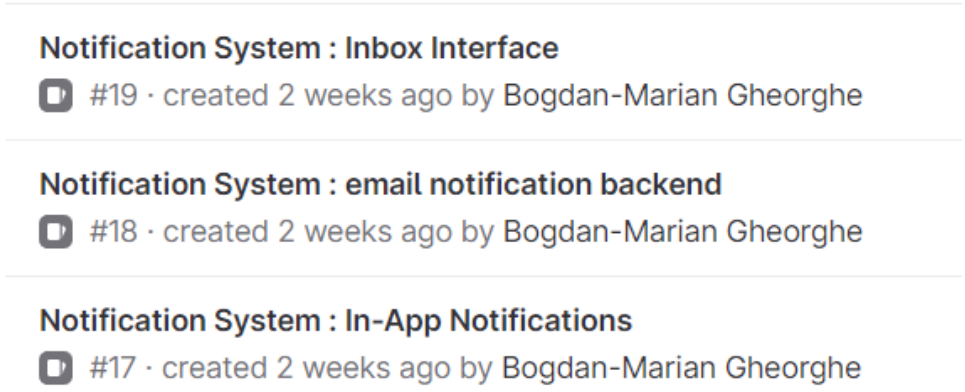
The Inbox will be one of the features of the app which will include a list to display the emails as well as filtering and sorting options. The estimated time will be used to style and design elements to create a visually appealing and user friendly interface, such as icons colors and animations.

- **Notification System : email notification backend(8 hours)**

This involves using a third-party email service to handle the actual sending of the emails, creation of a data-model for the email notifications, I will be using Nodemailer to send the email messages. For this feature a logic for generating the emails is also required.

- **Notification System : In-App Notifications(12 Hours)**

This involves adding a button in the top left corner of the interface to always be able to view a short summary of activities such as, an event from the To-Do List being in progress or happening soon or the anti-procrastination feature being active or scheduled.



Technical Report on Application Architecture

Introduction:

This technical report outlines the application architecture of a time-management application that includes different features such as a Live Scheduler, anti-procrastination feaature, Alarm/Timer, To-Do List, Diary, Notifications System, History charts. The goal of this application is to help users manage their time more efficiently and increase their productivity.

Application Architecture :

-Front-End Interface : This component is responsible for presenting a visually appealing, user-friendly and easy to navigate interface. We will be using Angular to develop it, Angular is a popular front-end framework, a component-based framework for building scalable web applications.

-Back-End Interface : This component is responsible for handling user requests and storing data. It will be developed using Java and Spring Boot, a popular back-end framework. We will be using a lot of different APIs that will allow our application to work as good as possible.

-Database : This component is responsible for storing user's data. We are going to use PostgreSQL as the backend database which will be used when the application is deployed into production and H2DB as the backend database used when the application is developed locally.

Other Components:

- ***The Application will be developed using the application generation framework called JHipster.***
- ***Maven, a Java build and dependency automation tool which will help us manage the project build and dependencies***
- ***Npm is the package manager we weill be using to gather dependencies for the Angular front end***

- ***IntelliJ Ultimate is the IDE which will be used to work on the project***

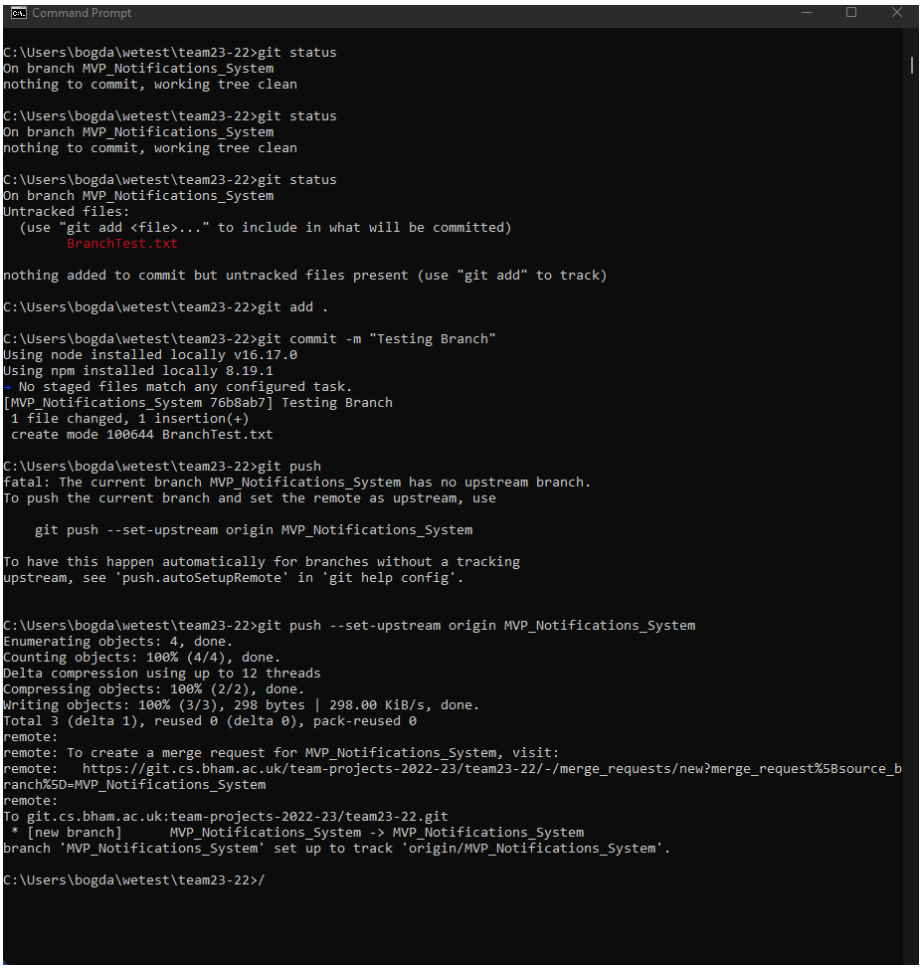
Features :

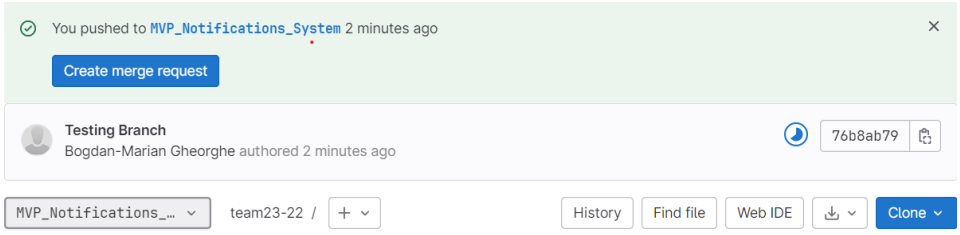
- ***Live Scheduler :***
  - ***Will include a calendar that displays what you should be doing at a certain time taken from the scheduler***
- ***Anti Procrastination Feature :***
  - ***The goal of this feature will be to block other websites and computer functions for a certain amount of time using a timer or an Alarm to alert the user of what is happening.***
- ***Alarm/Timer :***
  - ***A basic Alarm/Timer (normal time-management feature)***
- ***To-Do List :***
  - ***This feature will be a list of events that need to be completed and can be sorted by priority***
- ***Diary :***
  - ***Will be connected to the scheduler, will provide the ability to give more details about an upcoming event***
- ***History Charts:***
  - ***Will provide how much time the user managed to save while using the application , statistics about the to-do list events and overall performances of the user***
- ***Notification System:***
  - ***Will be divided in two parts, In-App notifications and Email notifications, will allow emails to be send to the user when a new activity is due soon or starting soon etc. Will also notify the user inside the application with notifications regarding his schedule.***

Conclusion :

The time-management application architecture outlined in this technical report is designed to be scalable, fault tolerant and user-friendly. The application includes various features including a Live Scheduler, anti-procrastination feature, Alarm/Timer, Email and In-App Notifications, To-Do List with priority classification, History Charts. The application will be developed using Angular, Java, Spring Boot, PostgreSQL, Maven, Jhipster, NPM, IntelliJ Ultimate, and varoius JavaScript libraries and APIs.

Tech stack/CI


Command Prompt
C:\Users\bogda\wetest\team23-22>git status
On branch MVP_Notifications_System
nothing to commit, working tree clean
C:\Users\bogda\wetest\team23-22>git status
On branch MVP_Notifications_System
nothing to commit, working tree clean
C:\Users\bogda\wetest\team23-22>git status
On branch MVP_Notifications_System
Untracked files:
(use "git add <file>..." to include in what will be committed)
BranchTest.txt
nothing added to commit but untracked files present (use "git add" to track)
C:\Users\bogda\wetest\team23-22>git add .
C:\Users\bogda\wetest\team23-22>git commit -m "Testing Branch"
Using node installed locally v16.17.0
Using npm installed locally 8.19.1
No staged files match any configured task.
[MVP_Notifications_System 76b8ab7] Testing Branch
1 file changed, 1 insertion(+)
create mode 100644 BranchTest.txt
C:\Users\bogda\wetest\team23-22>git push
fatal: The current branch MVP_Notifications_System has no upstream branch.
To push the current branch and set the remote as upstream, use
git push --set-upstream origin MVP_Notifications_System
To have this happen automatically for branches without a tracking
upstream, see 'push.autoSetupRemote' in 'git help config'.
C:\Users\bogda\wetest\team23-22>git push --set-upstream origin MVP_Notifications_System
Enumerating objects: 4, done.
Counting objects: 100% (4/4), done.
Delta compression using up to 12 threads
Compressing objects: 100% (2/2), done.
Writing objects: 100% (3/3), 298 bytes | 298.00 KiB/s, done.
Total 3 (delta 1), reused 0 (delta 0), pack-reused 0
remote:
remote: To create a merge request for MVP_Notifications_System, visit:
remote: https://git.cs.bham.ac.uk/team-projects-2022-23/team23-22/-/merge_requests/new?merge_request%5Bsource_b
ranch%5D=MVP_Notifications_System
remote:
To git.cs.bham.ac.uk:team-projects-2022-23/team23-22.git
* [new branch] MVP_Notifications_System -> MVP_Notifications_System
branch 'MVP_Notifications_System' set up to track 'origin/MVP_Notifications_System'.
C:\Users\bogda\wetest\team23-22>/


You pushed to MVP_Notifications_System 2 minutes ago
Create merge request
Testing Branch
Bogdan-Marian Gheorghe authored 2 minutes ago
76b8ab79
MVP_Notifications_…
team23-22 /
History
Find file
Web IDE
Clone