Part Time Work Scheduler

Tushar Kini
NC State University
Raleigh, North Carolina, USA
tkini@ncsu.edu

Prasad Kamath
NC State University
Raleigh, North Carolina, USA
pkamath3@ncsu.edu

Shlok Naik
NC State University
Raleigh, North Carolina, USA
snaik2@ncsu.edu

Boscosylvester Chittilapilly
NC State University
Raleigh, North Carolina, USA
bchitti@ncsu.edu

Ankur Baner
NC State University
Raleigh, North Carolina, USA
abaner24@ncsu.edu

ABSTRACT

A project born out of the lack of a unified shift management system for students working at multiple on campus employers at US Universities

KEYWORDS

Linux Kernel, best practices, developers,tools,part time, scheduler,

ACM Reference Format:
Tushar Kini, Prasad Kamath, Shlok Naik, Boscosylvester Chittilapilly, and Ankur Baner. 2022. Part Time Work Scheduler. In *Proceedings of ACM Conference (Conference'17).* ACM, New York, NY, USA, 2 pages. https://doi.org/XXXXXXX.XXXXXXX

1 INTRODUCTION

It is common for many students in US universities especially international students to work part time at on campus employers either to supplement their coursework or to earn money in their free time for paying towards their tuition. Each of these employers may select a different application to keep a track of the shifts due to the lack of consensus on a standard application to be used by all. This often leads to different employers choosing different systems based on their own needs. While this is not a problem for students working at a single location,it often causes problem to students that have multiple employers as they have to check schedules, give requests for dropping/taking up a shift etc on different applications. This project presents a unified application which can be used by all on campus employers so that the process is easy and standardised for all the stakeholders involved.

2 LINUX KERNEL BEST PRACTICES

2.1 Process Scalabilty using Distributed Development Model

To keep the project scalable various points have been executed-

- Instead of using a relational database a NoSQL database MongoDb is being used so that the app is horizontally for different institutions willing to adopt this application. Sharding supported by MongoDb can also be used in the future.
- An hierarchical approach was adopted for designing the project. First the functionality of all users were decided. Then the database was made. They the UI was made and finally the testing of the system as a whole was started.
- Integration of individual files and progress were discussed on a google meet on a regular basis.
- The use of Node.js ensures easy vertical scalabilty in the future
- The choice of the version controlling tools, code formatters, code checkers, code editors etc were decided with consensus keeping in mind the current trends in the software development industry

2.2 Consensus Oriented Model

The rubric points that help establishing the above are-

- The existence of a chat channel helps us know that there is a mode of communication available for developers and stakeholders to communicate any issues.
- Issues being opened, discussed thoroughly and being closed is a very good sign that the concerned developers are notified and being involved in solving issues.
- Also the commits made to config files by different developers show that the they all are using the similar tools that has been decided by a consensus.

2.3 Zero Internal Boundaries

- The changes made to config files by different developer denote that they are using the same tools for development and
- The ability of all the developers to commit to anything in the project shows that there are no technological barriers in between.
- Evidence of people working at multiple places in the codebase denote that there are no technological or logical barriers for the developer to understand and extend the functionality.

2.4 The No Regressions Rule

- There mere existence of test cases and use of automated testing and analysis tools in a project supports this point.

Conference'17, July 2017, Washington, DC, USA

ACM ISBN 978-x-xxxx-xxxx-x/YY/MM…$15.00
https://doi.org/XXXXXXX.XXXXXXX

- The regular execution of automated tests and use of CI/CD platform ensures that all the subsequent releases or addition of functionality of the project does not render the earlier ones unusable.
- The presence of system architecture definition in GitHub Actions(CI Pipeline) make sure that the newer versions of the project run on all the systems where earlier versions were usable. the number of issues raised or still open give us a fair idea about the usefulness of the project as very few and a lot of issues, both are undesirable.

2.5 Short Release Cycles

- The presence of short release cycles in the project depict that there is stable development in small incremental steps. In a small project in a nascent stage as ours, it is difficult to have stable releases to be made frequently.
- The presence of frequent small significant commits depict that all the developers get the modifications and additions made by all the developers. A pretty high number of commits(>110) for such a small project shows that this thing was taken care of.

2.6 Tools Matter

- The use of version controlling tools like GitHub, CI pipeline like GitHub Actions help a lot in collaboration.
- The same tools like *VS Code*,*eslint*, *prettier* were being used by all developers.

2.7 Corporate Participation

- The functionality of the project was decided after consulting friends and team members who work at on campus jobs. The shortcoming were highlighted by them and the very idea of designing such a system is acknowledged to them.
- All the 5 team members have been cooperative during meetings and contributed in some way.
- The distributed workload and the number of commits to the project prove that there is no single developer running the show and it is a collective effort.

ACKNOWLEDGMENTS

To numerous friends and colleagues who shared their experiences to highlight the shortcomings of the existing shift management system and the need for a unified versatile version of the same

REFERENCES

A ONLINE RESOURCES

The online resources referred for development are-

- https://github.com/txt/se22/blob/main/docs/proj1.md#rubric
- https://tutorial.djangogirls.org/en/deploy/
- https://jestjs.io/docs/configuration#collectcoveragefrom-array
- https://about.codecov.io/blog/measuring-typescript-code-coverage-with-jest-and-github-actions/
- https://reactjs.org/docs/getting-started.html
- https://expressjs.com/en/5x/api.html
- https://expressjs.com/en/5x/api.html
- https://medium.com/@beaucarnes/learn-the-mern-stack-by-building-an-exercise-tracker-mern-tutorial-59c13c1237a1

,

Received 10 October 2022