IJIRSET
International Journal of Innovative Research in
SCIENCE | ENGINEERING | TECHNOLOGY

e-ISSN: 2319-8753 | p-ISSN: 2347-6710

INTERNATIONAL JOURNAL OF INNOVATIVE RESEARCH

IN SCIENCE | ENGINEERING | TECHNOLOGY

Volume 12, Special Issue 2, April 2023

ISSN INTERNATIONAL STANDARD SERIAL NUMBER INDIA

9940 572 462 | 6381 907 438 | ijirset@gmail.com | www.ijirset.com

International Journal of Innovative Research in Science, Engineering and Technology (IJIRSET)

|| e-ISSN: 2319-8753, p-ISSN: 2347-6710| www.ijirset.com | SJIF: 8.423|| Volume 12, Special Issue 2, April 2023 ||

8th National Conference on Advanced Computing Technologies (NCACT'23)

Organized by

Department of CSE, Velammal Engineering College, Chennnai, India

Productivity Management Tool For Project Management APPS-A Survey

Joshua Joseph Vaidyan [1], Dhakshunhaamoorthiy[2]

U.G. Scholar, Department of Computer Science and Engineering, Velammal Engineering College, Chennai, Tamil Nadu, India [1]

Assistant Professor, Department of Computer Science and Engineering, Velammal Engineering College, Chennai, Tamil Nadu, India [2]

ABSTRACT: Time tracking and productivity management tools are quint essential in today's world to managers, product managers and team leads to measure their teams or employees' productivity and boost the performance of the team. Productivity management tools also prompt and motivate the team to complete the work faster. Especially for time tracking software application time tracking software will help log time for track efficiently. The tool provides a reliable way to measure the time spent on each task thereby measuring and tracking billable and non billable hours. Time tracking software also provides a means to measure the overall work quality and turn around time of the team in doing tasks. This allows managers to understand key competencies of each member in the team and assign tasks accordingly to the team. A good time tracking software may help stay ahead of the competition.

KEYWORDS: Time sheets, Management, time tracking, Software, Productivity management.

I . INTRODUCTION

One of the tedious tasks of managers is tracking their teams productivity efficiently. Project management tools such as Asana, Trello, Zoho Projects, Wrike and many more provide a means to manage projects efficiently. However almost all of them lack a certain feature automated time tracking. In the current scenario time sheets are the tools employed to know how much time is spent on a task by each of the member in the team. But the current implementation is entirely manual. Meaning timesheets are added by the project users themselves. But think about it for a moment, if an employee forgets to log time for the day or if the employee forgets to turn of automated timer (some products have timers which can be started and stopped for a certain task to automatically add timesheets) or someemployees would purposefully log extra hours on a task.

All this can lead to losses for the team in terms of resource, time and budget. Moreover there is no way for the managers who approve the time sheets to know if the hours logged in are the actual hours spent on a task by the user. Above all this can lead to slowness inside the team which can cause reduction in the net work done by the team. In today's competitive world it is important to stay ahead in every aspect feature development, bug or issue fixes and customer support. So every minute spent inside the work place is really important for the team.

Current solution implementations for the above said problems are available in the market but as third party extensions. This is a major issue for the customers because trust is most important factor between those organisation and their clients. Most of the product management tools today follow the most advanced security protocols and policies to ensure the safety of the data of customers. Using third party software for such important purposes can lead to data leak which is a serious issue in today's world of cut throat competition. However as most of the product management tools are only available as just web apps it is impossible to get statistics of the system directly. As there is no API to get system information necessary to track productivity. The only API available to track users' productivity is by MediaCapture API[8] which can only record systems screen, system audio input or camera input.

II. INITIAL WORK AND ANALYSIS

As discussed earlier one of the ways to track system information is by recording users screen. The best way to do this is to record users screen using the mediaCapture API native to the browser. As most of the project management apps are browser based this can be the most comfortable way to integrate this feature into existing systems without having to

|| e-ISSN: 2319-8753, p-ISSN: 2347-6710| www.ijirset.com | SJIF: 8.423|| Volume 12, Special Issue 2, April 2023 ||

8th National Conference on Advanced Computing Technologies (NCACT'23)

Organized by

Department of CSE, Velammal Engineering College, Chennnai, India

compromise the functionalities or UI implementations of the existing feature. Also as mentioned with the integration of existing timer functionality time sheets can be automated. With the mediaCapture API screen recordings can be attached with the time sheets so the managers and product managers can approve them based on the recording submitted along with the timesheet [8].

The main challenge of this method is with the size of the video captured by the API. Though the mediaCapture API is native to the browser it uses systems capabilities like video, audio and screen grab methods to retrieve information. This data is captured as blob and converted into required format. Some of the most popular formats include ffmpeg, mp4, mkv and many more. However most of these formats are quite heavy [8]. For an analogy a 3 second video is as big as 3Mb which is very large for that given time period. So we can only imagine the size of the data that will be collected in an entire work day.

Moreover most of the clients will require this data to be available in the cloud for managers to approve timesheets rather than check this data on users computers. Also in post COVID world as the world is working in hybrid mode, cloud functionality is also very much necessary for the feature to be useful. This further improves productivity as the managers can approve timesheets from anywhere and assign tasks on the go if required to do so, data will be secure and available, accidental deletions can be avoided.

Hence in order to facilitate cloud storage the data collected should be reduced in a lossless format. Services such as Youtube, Instagram and WhatsApp employ such lossless conversion techniques to improve performance of their applications and facilitate storing of such data. The first key step to implementing such compression and storing technique is in selecting the right format for recording the data. As mentioned before data of different formats will have different sizes. On analysis the most suitable format was found to be webm or ffmpeg format. Webm format because it is native to the web and is of the smallest possible size and later because of pre compression technique employed in the codec inherently. Both of these candidates also perform well in most browsers and can be viewed directly without needing additional codecs or plugins to be installed in the browser [11].

The next step in implementing effective compressing and storage technique will be in implementing a suitable exclusive compression technique. Meaning apart from the codecs compression additional compressions can be done on the video file so that size can be further reduced. Whatsapp uses conversion of any available format to the standard famous H.264 format. This allows WhatsApp to reduce a 600 Mb MPEG4 media to as low as 6Mb while retaining most of the quality. This Is great not only for the end users, but also for WhatsApp servers [11].

The based way to include a video compression technique from the browsers side is to use the compressionStream API which is also native to most of the browsers. The compression stream API uses the GZIP format a standard used to compress web resources [10]. The compression and decompression of data can be simply done by piping the input stream from mediaCapture API to the compressionStream and obtaining the compressed ZIP. The compressed ZIP can be stored in the server effectively and then retrieved on request and decompressed on the browser side before displaying it to the browser. This method however has several pros and cons [9].

Pros of using GZIP compression technique is that the GZIP compression is a format used across all of the internet for web resources this ensures safety and support for the standard for years to come [10]. GZIP is supported by all browsers hence no special plugins or codecs are required on the client side to decompress the data. The data being stored in a ZIP format, there is no possibility of embedded malicious code entering the system. As promises are used upload and download can be handled effectively after compression process. Cons of using GZIP is that the video is not stored in its native format like H.264 hence processing the video may be difficult as server requires to have implementations to extract the GZIP content and perform analysis and manipulations using advanced technologies like AI to provide more insight to the video [discussed later in this paper]. The load on the browsers may increase as so many operations are performed [10]. However no such loads on the browser were experienced on our side while testing the pilot app.

Further as the app is implemented in client side suitable care must be taken to ensure that the file is uploaded to the cloud before any system failure, unexpected shutdowns or accidental closing of sessions happens. For this it is important to upload the stream to the cloud as recorded to on the client side. But current implementation of the pilot app will not be able to do this as the blob should be entirely converted to compressed format. A possible solution to this problem is splitting up the video files into several parts and uploading the parts as and when acquired on the client side

|| e-ISSN: 2319-8753, p-ISSN: 2347-6710| www.ijirset.com | SJIF: 8.423|| Volume 12, Special Issue 2, April 2023 ||

8th National Conference on Advanced Computing Technologies (NCACT'23)

Organized by

Department of CSE, Velammal Engineering College, Chennnai, India

[7]. To make it more fault tolerant it is better to download the parts acquired then and there into users system and download it using suitable backtracking mechanisms.

However such implementations are also difficult to establish as there are a number of factors that are to be considered while designing such a system. The primary case to be considered is that the user deleting the files accidentally before upload, tampering or correction or alteration of acquired data on the client side, losing track of files to be uploaded to the system. Considering all these challenges ahead, browser based implementation of the system will be futile how hard we try. Also it is not a good practise to expose system activity to browser as any simple attack on the browser can leave a backdoor to the entire core system and possibly the organisations network. Hence it is better to implement a system program for better security and more functionalities [5].

III. METHODOLOGY AND IMPLEMENTATIONS

Hence to overcome all the above limitations it was inferred that the best way to implement this system is as a native or a hybrid application along with shell scripting, rather as a mere web application. Hybrid application would be the best as it can bring benefits from both the worlds (the world of web and the power of scripting)[5]. The idea is to build three versions of the app. One for each of the major operating systems use around the world - Windows, Mac and Linux [6].

Speaking of Linux using electron will give ability to use existing implementations for he front end, because all the other implementations like JavaFX, GTK, Gnome and PyQt are far behind when compared to web technologies. Also a lot of time may be required to take the UI usable and populate the data to the UI and manage security for the app separately. Having hybrid implementation will also help in avoiding handling changes in the app for every rollout in the web app [6]. Speaking of the Scripting side. A generic script can be built and executed with inbuilt shell script executer of the electron framework [4]. Also electron provides necessary inbuilt methods to execute OS based initialisations. For different OS this helps in file initialisations, metadata creation and folder initialisations at OS level [3].

For linux and MacOS inbuilt commands can be executed. For example, Process Listing, time calculation for all processes and detecting application startups the “top” command can be used. Along with “grep” command and “awk” command incoming processes can be passed through pipe and into file system. For video capture in macOS “screencapture” command can be used and for linux “ffmpeg” commands can be used. Suitable methods for compression, streaming and storing in cloud suitable compressionStream methods can be used. Also for sleep time calculation in macOS “log” [2] command can be used and for linux systems “systemctl” command can be used. For fault tolerance and sync metadata will be stored for each folders accordingly data can be uploaded on failure. For Windows OS systems process listing can be done via Windows event viewer and using “WinEvent” through Windows PowerShell. For Video Capture and Sleep Time native features and APIs in electron can be used [1]. All the processes are diagrammatically explained in Fig 3.1

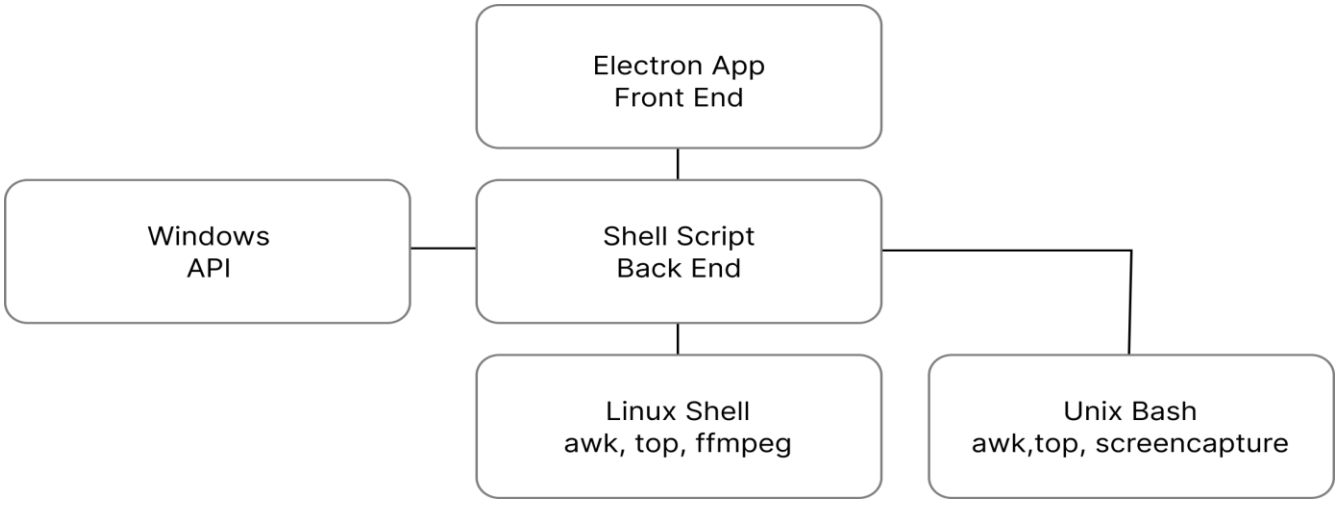

Fig 3.1 Overall Application Structure across OS

|| e-ISSN: 2319-8753, p-ISSN: 2347-6710| www.ijirset.com | SJIF: 8.423|| Volume 12, Special Issue 2, April 2023 ||

8th National Conference on Advanced Computing Technologies (NCACT'23)

Organized by

Department of CSE, Velammal Engineering College, Chennnai, India

IV. CONCLUSION

Based on all these analyses an app can be built that can effectively automate the task of automating timesheets with the necessary evidence of work for the task. This improves overall productivity of the team and trust. The app can be built in such a way that it won't disrupt the web implementation and also provide a similar experience to the users of the app on the local machine all the while maintaining security. Fault tolerance scheme also helps users to sync the data and re upload the data to the cloud if missed by the program.

REFERENCES

[1] Mac automating Tasks on Sleep:https://www.kodiakskorner.com/log/258
[2] Log command help desk: https://discussions.apple.com/thread/8021661
[3] Electron.js docs: https://www.electronjs.org/docs/latest/tutorial/quick-start
[4] Executing shell from Node.js: https://stackabuse.com/executing-shell-commands-with-node-js/
[5] Is scripting good for business: https://www.bairesdev.com/blog/shell-scripting-good-fit-for-your-business/
[6] Unified User Experience: https://medium.com/@pixelhalunke/about-unified-ux-frameworks-c950f9d891b0
[7] Memory Management in Mac:
https://developer.apple.com/library/archive/documentation/Performance/Conceptual/ManagingMemory/Articles/AboutMemory.html#/
[8] Screen Capture API: https://developer.mozilla.org/en-US/docs/Web/API/Screen_Capture_API
[9] Compression Streams: https://developer.mozilla.org/en-US/docs/Web/API/Compression_Streams_API
[10] GZIP: https://www.gnu.org/software/gzip/
[11] WhatsApp compression streaming: https://twitter.com/WABetaInfo/status/779789143928926209?lang=en

INNO SPACE
SJIF Scientific Journal Impact Factor

SJIF: 8.423

ISSN INTERNATIONAL STANDARD SERIAL NUMBER INTERNATIONAL CENTRE

निस्केयर NISCAIR

INTERNATIONAL JOURNAL OF INNOVATIVE RESEARCH

IN SCIENCE | ENGINEERING | TECHNOLOGY

9940 572 462 6381 907 438 ijirset@gmail.com

www.ijirset.com

Scan to save the contact details