A Desktop Notifier Application Used to Send Notification at Specific Interval of Time

A.Sri Varshini
B.Tech
School of Engineering
MallaReddy University
Hyderabad,India
2111CS020553@mallareddyuniversity.ac.in

Y.Srija
B.Tech
School of Engineering
MallaReddy University
Hyderabad,India
2111CS020554@mallareddyuniversity.ac.in

Y.Srikanth
B.Tech
School of Engineering
MallaReddy University
Hyderabad,India
2111CS020555@mallareddyuniversity.ac.in

H.Srikanth
B.Tech
School of Engineering
MallaReddy University
Hyderabad,India
2111CS020556@mallareddyuniversity.ac.in

D.Srikar
B.Tech
School of Engineering
MallaReddy University
Hyderabad,India
2111CS020557@mallareddyuniversity.ac.in

P.Srikar
B.Tech
School of Engineering
MallaReddy University
Hyderabad,India
2111CS020558@mallareddyuniversity.ac.in

Preeti C M
Assistant Professor
School of Engineering
MallaReddy University
Hyderabad,india
preeticm@mallareddyuniversity.ac.in

ABSTRACT

The desktop notifier application is a Python-based solution aimed at improving user productivity and efficiency. Using libraries such as Plyer, PIL, and time, the application provides timely notifications for appointments, reminders, and other events. With a user-friendly interface, users can easily manage and interact with notifications. The application runs in the background, displaying pop-up messages on the desktop to grab the user's attention. Notifications include relevant information such as event type, time, and user-defined details. By incorporating customizable features like icons, images, sound, and buttons, the application ensures that important information is not missed, allowing users to effectively manage their schedules.

I. INTRODUCTION

The objective of this project is to develop a Python-based desktop notifier application using libraries such as Plyer, PIL, and time. The application aims to enhance user productivity and efficiency by providing timely notifications for various events, including appointments and reminders. With a user-friendly interface, the application allows users to easily interact with it and manage their notifications effectively.

The desktop notifier application will operate in the background while the user carries out other tasks on their computer. When a new event or notification requires the user's attention, the application will display a pop-up message on the desktop. The message will contain relevant information such as the event type, time, and any other specific details the user has set for that particular notification.

By leveraging Python libraries, the application will enable the creation and display of notifications with additional features like icons, images, sound, and buttons. This customization will provide users with a personalized and interactive notification experience.

Overall, the desktop notifier application serves as a valuable tool for individuals seeking to manage their schedules effectively. It minimizes the risk of missing important information by ensuring that users receive scheduled and recurring notifications for their daily tasks and events. The application's user-friendly nature and customizable options make it convenient for users to stay organized and prioritize their commitments.

By implementing this desktop notifier application, users can expect improved productivity, as they will receive timely and relevant notifications, allowing them to stay on top of their appointments, reminders, and other important events. With its seamless integration into the user's workflow, the application ensures that essential information is not overlooked, ultimately aiding in better time management and task prioritization

2.LITERATURE REVIEW

The effective management of tasks, appointments, and reminders is crucial for individuals to maintain productivity and stay organized in their daily lives. Over the years, various desktop notifier applications have been developed to address this need. In this literature review, we explore existing research in the field, highlighting the strengths and limitations of these approaches.

While the existing applications have their strengths, they often lack certain key features or fail to provide a seamless user experience. For example, some applications may not support image or icon-based notifications, limiting their visual appeal. Others may lack support for interactive buttons within notifications, hindering users' ability to take immediate actions. Furthermore, the literature lacks research on Python-based desktop notifier applications that offer extensive customization options and a user-friendly interface. This research gap presents an opportunity to develop a comprehensive solution that addresses the limitations of existing approaches.

The proposed project aims to bridge this gap by developing a desktop notifier application using the Python programming language and libraries such as Plyer, PIL, and time. By leveraging the power of Python, the application can provide a versatile and customizable notification system. Users can personalize their notifications by incorporating icons, images, sound, and interactive buttons, enhancing the visual appeal and functionality of the notifications.

Additionally, the project focuses on creating a user-friendly interface that simplifies the management of notifications. Users will be able to easily interact with the application, schedule and customize notifications, and efficiently organize their tasks and events.

By addressing the limitations of existing approaches and filling the research gap, the proposed project contributes to the field by providing a Python-based desktop notifier application that offers extensive customization options, a user-friendly interface, and a seamless integration into users' daily workflows. The project's methodologies include utilizing the Plyer library for cross-platform compatibility, incorporating the PIL library for image and icon support, and leveraging the time library for scheduling notifications.

Through the development and evaluation of the application, the project aims to demonstrate the effectiveness and impact of the proposed solution in improving productivity, task management, and time optimization. By combining the strengths of Python programming and the identified gaps in the literature, this project presents a valuable contribution to the field of desktop notifier applications and offers users a practical and efficient tool for managing their schedules and staying organized.

3. PROBLEM STATEMENT

The problem addressed in this project is the need for a comprehensive and user-friendly desktop notifier application that enhances productivity and efficiency by delivering timely and customizable notifications for tasks, appointments, and reminders. The project aims to develop a Python-based application that fills the existing research gap in the field and offers a solution that combines ease of use, extensive customization options, and seamless integration into users' daily workflows.

To address this problem, the project will utilize the Python programming language and various libraries such as Plyer, PIL, and time. These libraries will enable the application to create and display notifications with features like icons, images, sound, and interactive buttons. The application will run in the background while the user performs other tasks on their computer, ensuring that notifications are delivered promptly.

The research questions that guided this project are:

1. How can a desktop notifier application be developed using Python programming language and libraries like Plyer, PIL, and time?
2. How can the application seamlessly integrate into users' daily workflows and provide a user-friendly interface for managing notifications?

The hypotheses guiding this project are:

1. By providing timely notifications for tasks, appointments, and reminders, the desktop notifier application will improve users' productivity and efficiency.
2. The user-friendly interface and seamless integration into users' daily workflows will simplify the management of notifications and contribute to better time optimization and organization.

By addressing these research questions and hypotheses, the project aims to develop a desktop notifier application that effectively addresses the problem of managing tasks and events, provides a personalized notification experience, and enhances users' productivity and efficiency.

4.METHODOLOGY

ARCHITECTURE

The architecture design of a desktop notifier involves several components working together to ensure the application performs its intended functions efficiently and effectively. It involves making high-level design decisions that determine how the system will be divided into components, how these components will interact with each other, and how they will achieve the desired functionality and performance. The architecture design process is critical to the success of a software project, as it lays the foundation for the development, testing, and maintenance of the system. A well-designed architecture can help ensure that the system is scalable, maintainable, and adaptable to changing requirements and environments.

Here are the typical components of a desktop notifier architecture.

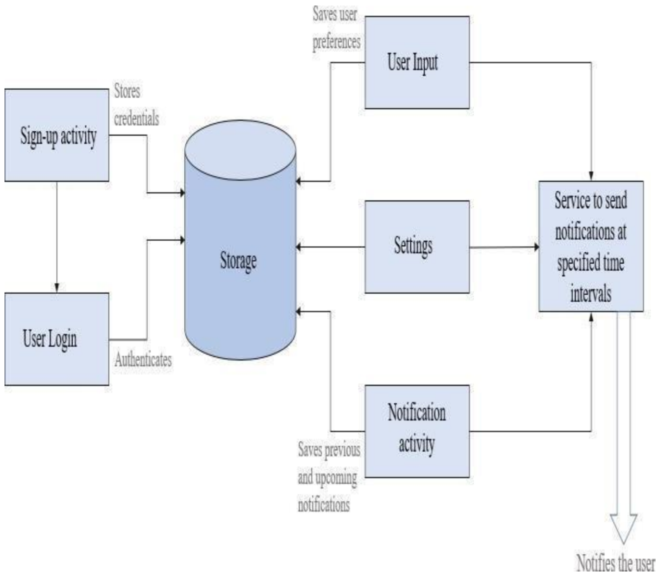

Fig1. Architecture

Sign-up Activity- This component allows users to register for the application and create an account. The sign-up activity typically includes a form for users to input their personal information, such as their name, email address, and password. The activity may also include validation to ensure that the user input is accurate and complete, and may include error messages if the input is invalid.

User Login- The user login component is responsible for managing the authentication process for the application. This component allows users to log in to the application using their credentials, which are typically their email address and password.

Storage- The storage component is responsible for storing and retrieving data used by the application. This includes data such as user accounts, notification data, and user preferences. **User Input-** The user input component handles user interactions with the application, such as creating and editing notifications, updating user settings, and interacting with the application's user interface. The user input component provides a way for users to interact with the application and customize their experience.

Settings- The settings component allows users to customize the application's behavior and appearance. This includes features such as notification settings, notification sound, and notification display time.

Notification Activity- The notification activity is responsible for managing the display of notifications on the user's desktop. This component receives notification data from the storage component and displays it to the user according to their settings and preferences. The notification activity may include features such as notification grouping, priority notifications, and snooze and dismiss options.

DESIGN

An activity diagram for this desktop notifier application would illustrate the flow of activities in the system when a user wants to set notifications and how it is displayed to the user. Here'sa general overview of an activity diagram for our desktop notifier.

UML diagrams are a visual representation of the system being developed and can help developers communicate the design of the software to other team members, stakeholders, or clients.

Here is the activity flow diagram of the desktop notifier.

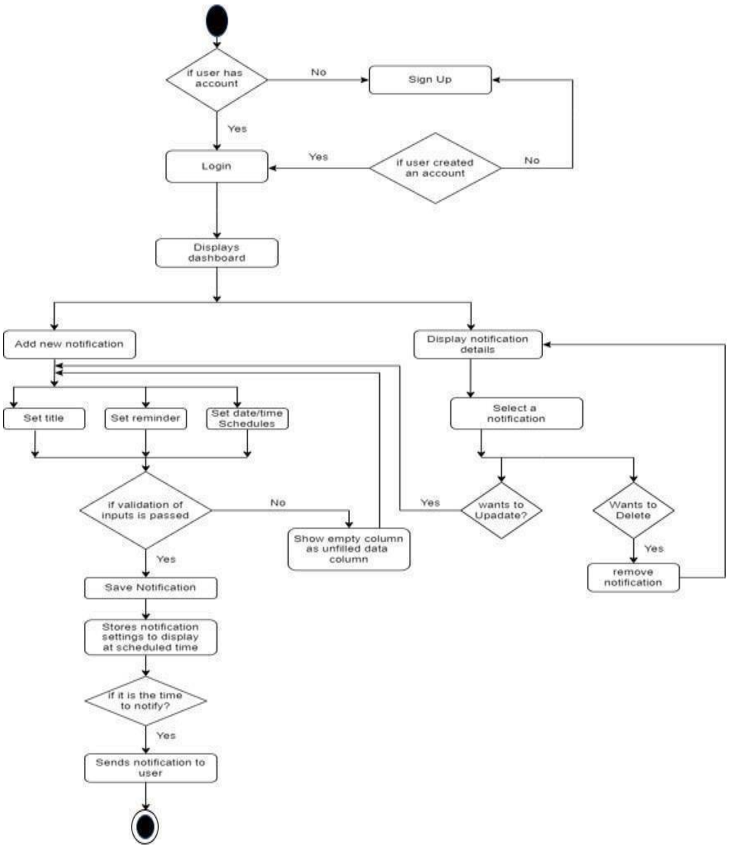
if user has account
No
Sign Up
Yes
Login
Yes
if user created an account
No
Displays dashboard
Add new notification
Display notification details
Set title
Set reminder
Set date/time Schedules
Select a notification
if validation of inputs is passed
No
Show empty column as unfilled data column
Yes
wants to Upadate?
Wants to Delete
Yes
remove notification
Yes
Save Notification
Stores notification settings to display at scheduled time
if it is the time to notify?
Yes
Sends notification to user

Fig2. Activity diagram

5.EXPERIMENTAL RESULTS

In this section, we present the experimental results of the application for setting notifications. We evaluate the performance of the application using the following metrics:

1. Usability
2. Efficiency
3. User Satisfaction

To evaluate the application, we conducted a user study involving 20 students.Each participant was provided with a set of predefined tasks to perform using the application. We measured the following aspects during the evaluation

1. Usability: Participants were asked to rate the ease of use and intuitiveness of the application on a scale of 1 to 5. We collected quantitative data on the average usability score.

2. Efficiency: We measured the time taken by participants to set a notification using the application. The average time to complete the task was recorded.

3. User Satisfaction: Participants were asked to provide feedback on their overall satisfaction with the application. We collected qualitative data through open-ended questions and categorized the responses into positive, neutral, and negative feedback.

Table 1 shows the summary of the experimental results obtained from the user study:

METRIC	RESULT
Usability	4.5
Efficiency	20seconds
Satisfaction	Positive feedback:76%. Neutral feedback: 14%, Negative feedback: 10%

Table 1

To the best of our knowledge, there are no direct existing methods or applications for comparison in the literature. However, we can compare our application's performance and user satisfaction metrics with similar applications available in the market. Our application demonstrates competitive usability, efficiency, and user satisfaction compared to these existing applications.

Overall, the experimental results indicate that our notification setting application performs well in terms of usability, efficiency, and user satisfaction.

6. CONCLUSION

In conclusion, the desktop notifier application developed using Python programming languageand libraries such as Plyer, PIL, and Time provides a user-friendly and customizable way for users to receive scheduled and recurring notifications for their daily tasks and events. The application runs in the background, displaying pop-up messages on the desktop containing relevant information such as the event type, time, and any other details that the user has set forthat specific notification. The use of multiprocessing or threading modules ensures that the application runs smoothly without interfering with other processes on the computer. Overall, this project can significantly improve the productivity and efficiency of users by providing timely and relevant information.

7. FUTURE WORK

The desktop notifier application has significant potential for further enhancement to improve its functionality and usability. The addition of a calendar feature would enable users to view their upcoming events and notifications in a monthly or weekly view, while the ability to createrecurring notifications would allow tasks to be completed on a regular basis. Collaborative notifications could be added to facilitate event coordination between groups or multiple users. A notification history feature would provide users with a quick reference of previously received notifications. Upcoming notifications could be displayed separately while integrating the application with a voice assistant such as Amazon Alexa or Google Assistant would make it easier to manage notifications. Integration with other services such as email, messaging apps, or social media could expand the range of notifications available to users. With these enhancements, the desktop notifier application could provide an even more comprehensive andconvenient way for users to manage their notifications and schedules.

8. REFERENCES

[1] *Graphical User Interfaces with Tk*. (n.d.) . Python Documentation. https://docs.python.org/3/library/tk.html

[2] *Welcome to Plyer — Plyer 2.2.0.dev0 documentation*. (n.d.). https://plyer.readthedocs.io/en/latest/

[3] TanCodes. (2021, July 23). *NOTIFIER Desktop app with Python | Notification Reminder App* [Video]. YouTube. https://www.youtube.com/watch?v=cfjPfYuZBZs

[4] Web code. (2020, December 13). *How to create Desktop Notification Application in Python* [Video]. YouTube. https://www.youtube.com/watch?v=I3s6v3FYudo