**Literature Review**

The research done in this section is focused on various task management applications that have been developed or done research on. This study includes well-developed task management applications, specialized and need-based applications, and different types of existing to-do applications.

To-do list applications are need-based applications of software that create a better organized environment for the users in terms of better daily organization. Some of the most widely used to-do lists are exceptional with the execution of some of the essential and basic functions that a simple to-do list should possess. A task management application generally offers adding, removing, editing and displaying the existing tasks; however, some people demand more functionality as they need additional features from the task manager. An example of this could be ‘To-doist’ (www.todoist.com) which is a widely used app that offers many features such as group projects, project sections, auto-scheduling, due-date setup, collaboration with others, assigning tasks to others, file sharing and adding and so on.

Some task manager designs set out to develop an automation system where the user get recommendations based on the previous tasks. ‘TaskDo’ [3] designed an automation system which enables the application to collect feedback on task completion. After collecting the data over specific periods the recommender will start to make suggestions on the date, time or the type of task. For example, if the user regularly fails to complete hiking on Saturday at 11 am, and they successfully execute coffee and reading at 3 pm, then the system will recommend prioritizing coffee and reading on Saturdays. Additionally, if fishing tasks are regularly completed on Sunday morning, the system will suggest moving hiking to Sunday afternoon. Additional feedback on the suggestion will improve the quality of the system by eliminating badly selected recommendations and focusing on feedback-based evaluation.

Developing a C++ To-do list application requires clever usage of data structures such as linked lists, trees, priority queues and more. To use the right data structures there should be a predefined design for the application. If the developer wants to build a basic task manager which can add, delete and edit tasks, then they are more likely to use doubly linked lists than other data structures, because double linked list can be easily manipulated [2]. However, a more complex task manager can be developed using priority (priority-based task completion) or map and unordered map (for quick search and categorization)

In addition, there are some other approaches to the development of smart task managers that give the users better experience in terms of encouragement and reinforcement. Tamu To-Do [4] is an application developed with a gamification feature. This feature has drawn influences from some of the most popular apps like Duolingo ([www.duolingo.com](http://www.duolingo.com)) and Habitica (www.habitica.com) which artfully use gamification to boost user activity and encourage users to be more active. For example, Duolingo offers awards and points for certain task completions such as protecting daily streak and getting exercises done. So, Tamu To-Do uses a grade-based gamification system on a scale of 0 to 100. Tamu the pet displays four different expressions 0-25 ‘very sad’, 25-50 ‘sad’ 50-75 ‘happy’ and 75-100 ‘very happy’. This is supposed to reinforce a better user experience; according to the Tamu To-Do research which got positive results, where the vast majority reported significant boost in their daily task completion in addition to emotional attachment to the pet which also boosted the productivity of the users.

The literature review investigated various existing to-do list applications and analyzed their architecture and design with the purpose of understanding common types of task management tools that are used widely and successfully. From a reward-punishment system to a feedback-based recommendation system it can be noted that there are many well-structured approaches to the development of well-functioning applications that guarantee user satisfaction and longevity.

**References**

**[1]** Y. Gil and V. Ratnakar, “Automating to-do lists for users: Interpretation of to-dos for selecting and tasking agents,” in *Proceedings of the AAAI Conference on Artificial Intelligence*, 2008.

**[2]** A. Y. Nugroho and N. H. Sutanto, “Exploring the code foundation: A literature review of data structures in C++,” *Int. J. Mech., Ind. Control Syst. Eng.*, 2024.

**[3]** J. Han, H. Kim, and S. Park, “TaskDo: A daily task recommender system,” in *Proc. IEEE Int. Conf. Big Data Smart Comput. (BigComp)*, Kyoto, Japan, 2019, pp. 1–8. doi: 10.1109/BIGCOMP.2019.8862073.

**[4]** A. Kumar, R. Jain, and P. Sharma, “Gamification-based to-do list mobile application development,” in *Proc. IEEE Int. Conf. Recent Adv. Electron. Comput. Sci. Eng. (RAECS)*, Chandigarh, India, 2023, pp. 1–5. doi: 10.1109/RAECS2023.10277897.

**[5]** P. Landes and B. M. Di Eugenio, “A supervised approach to the interpretation of imperative to-do lists,” *arXiv preprint arXiv:1806.07999*, 2018.