International Journal of Science, Strategic Management and Technology
Volume **02** Issue **05** May-**2026** | ISSN: **3108-1762 (Online)** | Impact Factor: 3.8
An International, Peer-Reviewed, Open Access Scholarly Journal Indexed in recognized academic databases

Studyflow: The Smart Study Planner

Ved Patel

Computer Science Engineering / Shri Ram Institute of Technology, Jabalpur, India

Corresponding Author Email: ved2032patel@email.com

https://doi.org/10.55041/ijsmt.v2i5.295

Cite this Article: Patel, V. (2026). Studyflow: The Smart Study Planner. International Journal of Science, Strategic Management and Technology, *02*(05). https://doi.org/10.55041/ijsmt.v2i5.295

Abstract—Students often start a semester with clear goals, but many find it difficult to turn those goals into a steady daily routine. StudyFlow: The Smart Study Planner is designed to make that routine easier by bringing task planning, deadline tracking, revision scheduling, progress monitoring, and adaptive reminders into one digital environment. This paper presents the design, implementation direction, and evaluation framework for StudyFlow, with a focus on students in secondary and higher education. The application allows students to add subjects, topics, deadlines, difficulty levels, estimated study hours, and personal preferences. Using these inputs, the planning engine creates a structured schedule, breaks larger academic goals into smaller sessions, suggests revision slots, and updates the plan when tasks are delayed. The paper also explains the system architecture, planning logic, expected outcomes, privacy considerations, and future scope. The main contribution of this work is a practical, student-friendly model for connecting academic time management with self-regulated learning principles.

Keywords— StudyFlow; smart study planner; self-regulated learning; time management; learning analytics; academic planning.

I. Introduction

Background of the Study

Academic success is influenced not only by intelligence or access to learning materials but also by the ability to organize effort over time. Many students know what they need to study, yet they postpone tasks, underestimate preparation time, or concentrate too much effort close to examinations. Digital planners are widely available, but many of them behave like generic calendars. They remind students about deadlines, but they do not always explain how a task should be divided, when revision should happen, or how a plan should change when the learner falls behind.

The StudyFlow application described in this paper is designed as a student-centered planning system. It brings together task entry, automatic scheduling, priority assignment, reminders, progress recording, and reflection. Instead of expecting students to manually decide every study session, the application converts academic goals into manageable sessions. It also supports self-regulated learning by prompting students to plan, act, monitor, and adjust their learning behavior [1], [2].

In real student life, study planning is often affected by uncertainty. A learner may have school classes, commuting time, part-time work, family responsibilities, and personal routines. A planner that ignores these realities can create schedules that look organized but are impossible to follow. Therefore, the StudyFlow system emphasizes flexible planning. It is not meant to control the student; it is meant to guide the student toward more realistic and consistent academic habits.

Problem Statement

Existing calendar and reminder tools are useful for storing events, but they rarely provide academic intelligence. They do not automatically consider the difficulty of a subject, the urgency of an upcoming deadline, the length of a revision cycle, or the student's available energy during a week. Learning management systems may show assignments and due dates, but they are usually course-centered rather than learner-centered. As a result, the burden of coordination remains with the student.

The core problem addressed by this study is the gap between academic workload and personal study organization. Students need a system that can translate deadlines and goals into daily action plans, while still allowing human judgment and modification. A StudyFlow can reduce cognitive overload by giving students a clearer view of what must be done, when it can be done, and how much progress has already been made.

Objectives of the Study

The main objective of this paper is to design a StudyFlow application that supports personalized study scheduling and academic time management. The study also aims to identify the functional requirements of such an application, propose a modular architecture, describe a planning algorithm, and present an evaluation framework for future testing.

More specifically, the paper focuses on four objectives: to design a system that converts student inputs into structured plans; to connect planning features with self-regulated learning principles; to provide diagrams and implementation details that developers can follow; and to suggest measurable indicators for usability, effectiveness, and student satisfaction.II. Background and Research Context

Student Time Management as a Learning Challenge

Time management is a common difficulty across school, college, and online learning environments. Students often have multiple academic tasks competing for attention, and the consequences of poor planning are usually visible only when deadlines are near. The issue is not simply that students lack motivation. In many cases, they lack a clear method for converting intentions into routine behavior. A well-designed planner can make this conversion easier by displaying the next manageable step instead of only the final deadline.

Research on self-regulated learning highlights planning, monitoring, and reflection as repeated learning processes [1], [3]. A student first sets a goal, then applies strategies, observes progress, and finally adjusts future behavior. A study planner application can support each of these stages. It can ask the learner to set deadlines and priorities, record completed sessions, compare planned time with actual time, and recommend changes when the original plan becomes unrealistic.

Technology-supported learning analytics has also shown that data about time, activity, and progress can help students become more aware of their behavior [2], [4]. However, the usefulness of analytics depends on how clearly the system communicates information. Students do not need complicated dashboards alone. They need simple explanations, practical suggestions, and an action path that fits their schedule.

Need for a Smart Planning Application

A smart planner differs from a basic planner because it uses rules or algorithms to create, adjust, and explain schedules. It can rank tasks by urgency and difficulty, distribute study sessions before a deadline, add revision slots after initial learning, and send reminders when progress is low. These features are especially helpful when a student manages several subjects at once.

The StudyFlow application focuses on practical intelligence rather than excessive automation. Students remain in control of goals, preferences, and final decisions. The system offers a recommended plan, but users can drag sessions, edit duration, mark tasks as difficult, or pause a subject during emergencies. This balance is important because student agency is a central element of self-regulated learning [5].

Another reason for a smart planning application is the growth of hybrid and online learning. In these settings, students are expected to handle more independent work, which increases the need for planning support. Reviews of online learning environments suggest that tools supporting self-regulation, time management, goal setting, and feedback can improve learning experiences when they are thoughtfully implemented [6], [7].III. Literature Review

Self-Regulated Learning and Study Planning

Self-regulated learning provides the theoretical foundation for the StudyFlow. It describes students as active participants who plan, monitor, and evaluate their own learning rather than passively receiving instruction. In the context of this paper, the planner is not only a scheduling tool but also a scaffold that helps students practice the habits of self-regulated students. Goal setting is supported through task creation, strategic planning through schedule generation, monitoring through progress tracking, and reflection through weekly reports.

Systematic reviews on smart learning environments have identified time management, task strategy, help seeking, goal setting, and self-evaluation as important self-regulated learning processes [1], [6]. These processes match the core modules of the StudyFlow application. For example, the task module allows students to define what needs to be done; the planning engine determines when it can be done; the progress module shows how much has been completed; and the reflection module encourages students to learn from previous planning behavior.

Learning Analytics and Feedback

Learning analytics can be defined as the collection and interpretation of learning-related data to improve learning processes. In a study planner, analytics may include the number of planned sessions, completed sessions, delayed tasks, revision frequency, and time spent by subject. These data points can be converted into feedback such as 'Mathematics tasks are often delayed' or 'Your revision plan is too close to the exam date.' Such feedback is more meaningful than raw statistics.

Prior research indicates that learning analytics can support self-regulation when feedback is timely, understandable, and connected to action [2], [4], [8]. A planner should therefore avoid overwhelming students with graphs. Instead, it should present a few useful indicators and recommendations. The goal is to help the student answer three questions: What should I do next? Am I following my plan? What adjustment would make my plan more realistic?

Mobile and Calendar-Based Learning Support

Mobile applications are suitable for study planning because students often carry their phones throughout the day. A planner that sends reminders, displays daily tasks, and records progress can become part of the student's normal routine. Studies on mobile learning analytics suggest that time tracking and mobile feedback can positively affect awareness of time management [3].

However, mobile planning also presents challenges. Too many notifications may create irritation, while poor interface design may discourage use. The application should therefore allow notification control, quiet hours, and personalized reminder frequency. A good study planner should feel supportive rather than intrusive.

Research Gap

The literature supports the value of self-regulation tools, learning analytics, and time-management interventions, yet many existing tools remain either too general or too complex. Generic calendars do not understand academic workload, while learning dashboards may show information without helping students plan the next action. The gap addressed by this paper is the need for an integrated student-facing application that combines academic scheduling, adaptive planning, progress analytics, reminders, and reflective feedback in a simple interface.

The proposed StudyFlow contributes to this gap by organizing features around actual student behavior. It does not assume that students always follow their first plan. Instead, it treats rescheduling, delay, revision, and reflection as normal parts of learning.

IV. Proposed System Design

System Overview

The proposed StudyFlow application is designed as a cross-platform mobile and web system. The user begins by entering academic subjects, tasks, deadlines, estimated difficulty, expected duration, and available study hours. The system then generates a study plan that distributes work across the calendar. It also includes spaced revision sessions, reminders, progress tracking, and weekly summaries.

The application is divided into five major modules: user profile management, task and deadline management, planning engine, notification service, and analytics dashboard. Each module performs a clear role. The profile module stores preferences and available hours; the task module handles academic workload; the planning engine creates and adjusts schedules; the notification service keeps the student aware of upcoming sessions; and the analytics dashboard shows progress and reflection points.

The system is designed for beginner-friendly use. A student should be able to create a study plan without understanding algorithms. At the same time, the application should be transparent enough to explain why a task was placed on a particular date. For example, the system may state that a science revision session was scheduled earlier because the exam is near and the student marked the topic as difficult.

Core Features

Key features include subject creation, assignment entry, examination entry, automatic session generation, daily study view, weekly calendar view, task completion marking, progress percentage, overdue warning, rescheduling, study streaks, revision suggestions, and simple analytics. The planner also supports manual editing because students must be able to adjust the schedule when real-life events interrupt the plan.

The application may include a focus timer, but this is not treated as the main feature. The central value of the system is planning quality. A focus timer helps students work during a session, while the planning engine helps students decide which session should happen in the first place.

Human-Centered Design Considerations

A student-friendly planner should respect the student's emotional experience. When a student misses a task, the system should not display harsh language. Instead, it should provide constructive options such as 'Move this task to tomorrow,' 'Reduce session duration,' or 'Split into two smaller sessions.' This approach makes the application more encouraging and less likely to be abandoned.

Accessibility is also important. The interface should use readable text, clear icons, consistent navigation, and simple color contrast. Students should be able to understand their schedule quickly, even during a busy day. The application should avoid unnecessary complexity and focus on what the student needs most: clarity, priority, and momentum.

V. System Architecture and Diagrams

Workflow of the StudyFlow

The workflow begins with student inputs and ends with reflective feedback. A student enters subjects, tasks, deadlines, and available study time. The planning engine processes these details and creates a calendar. The student follows the daily plan, marks sessions as complete, and receives reminders or adjustments when needed. Over time, progress data becomes the basis for weekly reflection.

Figure 1 illustrates this workflow. The diagram shows that planning is not a one-time event. The progress tracker sends information back to the planning engine, which can update future sessions. This feedback loop is important because student schedules are dynamic.

International Journal of Science, Strategic Management and Technology
Volume **02** Issue **05** May-**2026** | ISSN: **3108-1762 (Online)** | Impact Factor: 3.8
An International, Peer-Reviewed, Open Access Scholarly Journal Indexed in recognized academic databases

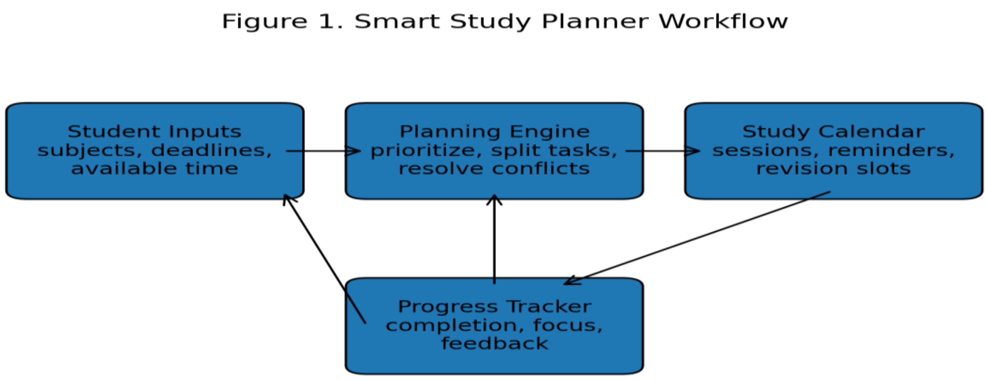

Figure 1. StudyFlow Workflow.

Layered System Architecture

The application architecture is organized into four layers: user interface, application services, intelligent planning layer, and data layer. This structure separates user interaction from business logic and data storage. A modular design also makes maintenance easier because changes to notification logic or scheduling rules do not require complete redesign of the interface.

The intelligent planning layer is the distinctive part of the system. It calculates priority scores, divides tasks into sessions, allocates them into free time slots, and adds revision intervals. In a future version, this layer could include machine learning models that learn from each student's behavior. For the current research design, a transparent rule-based approach is preferred because it is easier to explain and evaluate.

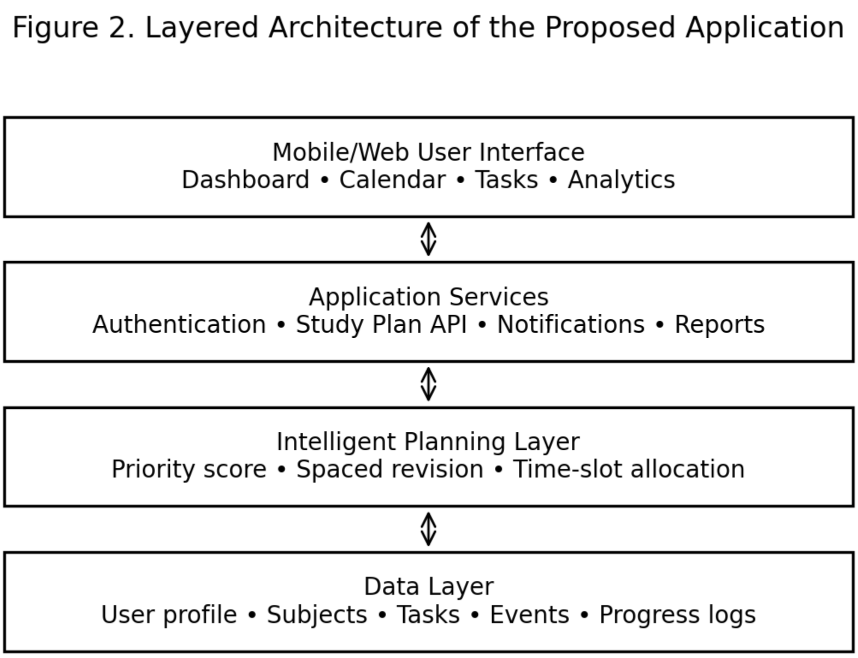

Figure 2. Layered Architecture of the Proposed Application.

Use-Case View

The use-case diagram presents the interaction between the student and the system. The student can create goals, add deadlines, generate a plan, track progress, and view reports. The notification service supports reminders and schedule adjustments. This view helps developers and evaluators understand the minimum expected behavior of the application.

Figure 3. Use-Case View of the Smart Study Planner

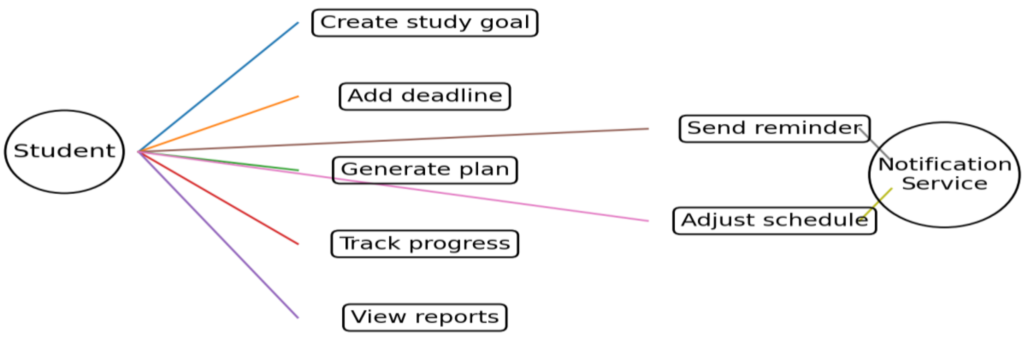

Figure 3. Use-Case View of the StudyFlow.

VI. Planning Algorithm

Input Parameters

The planning algorithm receives both academic and personal inputs. Academic inputs include subject name, task title, deadline, estimated work hours, difficulty level, exam date, and revision requirement. Personal inputs include available study hours, preferred study days, maximum session length, break preference, and blocked time. These inputs allow the planner to generate a schedule that is more realistic than a simple deadline list.

Each task receives a priority score. A task with a near deadline, high difficulty, and large estimated workload receives a higher score. The system then splits the task into sessions that fit within the student's available time. If a task cannot be completed before its deadline, the system warns the student and suggests options such as increasing study hours or reducing other commitments.

Priority Scoring Logic

A simple priority score can be calculated using urgency, difficulty, estimated duration, and task type. For example, an examination revision task may receive a higher weight than a routine reading task. The algorithm does not need to be mathematically complex to be useful. Its value lies in making planning decisions consistent and explainable.

The proposed scoring model is: Priority = 0.40(Urgency) + 0.25(Difficulty) + 0.20(Duration) + 0.15(Importance). Urgency is based on days remaining before the deadline. Difficulty is entered by the student or inferred from past performance. Duration represents total effort required, and importance represents whether the task affects grades, examinations, or personal learning goals.

Rescheduling Rules

Rescheduling is essential because students rarely complete every task exactly as planned. When a task is missed, the application first checks available future slots. If enough time exists before the deadline, the task is moved automatically or suggested for movement. If time is insufficient, the application shows a risk warning and recommends action. The system may also reduce large sessions into smaller sessions to make recovery easier.

A humane rescheduling process matters. Instead of treating delay as failure, the planner treats it as information. If a student repeatedly misses evening sessions, the application can recommend morning or afternoon sessions. If a subject is constantly postponed, it may indicate that the task is too difficult, too vague, or emotionally unpleasant. The planner can then recommend smaller steps.

VII. Methodology

Research Design

This paper follows a design science research approach. Design science is appropriate because the goal is to create and evaluate an artifact: the StudyFlow application. The process includes problem identification, objective definition, system design, prototype development, demonstration, evaluation, and communication.

The proposed study can be conducted in three phases. The first phase collects requirements from students through questionnaires and informal interviews. The second phase develops a functional prototype based on those requirements. The third phase evaluates usability and perceived learning support through user testing.

Participants and Sampling

For a full empirical version of this study, participants may include secondary school students, undergraduate students, or postgraduate students who use digital devices for academic work. A sample of 60 to 120 students would be suitable for initial usability testing, while a larger sample would be preferable for measuring learning outcomes. Stratified sampling can be used to include students from different years of study, academic streams, and levels of digital experience.

Participants should be informed about the purpose of the study, the voluntary nature of participation, and how their data will be used. No sensitive academic data should be collected without consent. If actual grades are used in future testing, ethical approval and institutional permission will be required.

Data Collection Instruments

The study may use a requirements questionnaire, usability testing checklist, System Usability Scale items, semi-structured interview questions, and application log data. The questionnaire can ask students about current planning habits, problems with deadlines, preferred reminder types, and desired features. Usability testing can measure how easily users create tasks, generate plans, edit sessions, and interpret progress reports.

Application logs can record non-sensitive behavioral data such as number of tasks created, sessions completed, sessions rescheduled, and reminders acknowledged. These logs should be anonymized and used only for research purposes.

Data Analysis

Quantitative data can be analyzed using descriptive statistics such as mean, percentage, standard deviation, and completion rate. If the study compares pre-use and post-use time-management confidence, paired sample tests may be used. Qualitative interview responses can be coded thematically to identify common user experiences, frustrations, and suggestions.

The analysis should focus not only on whether the application works technically but also on whether it supports student behavior. A technically correct planner may still fail if students find it confusing, rigid, or emotionally discouraging.

VIII. Implementation Details

Technology Stack

The StudyFlow application can be implemented using Flutter or React Native for cross-platform mobile development, with a web dashboard developed in React. The backend may be built using Node.js, Django, or Firebase services. A relational database such as PostgreSQL can store structured academic data, while Firebase Cloud Messaging or a similar service can handle reminders.

The choice of technology depends on the development context. For a student project, Firebase may reduce backend complexity and speed up prototype development. For a larger deployment, a custom backend may offer better control over data security, analytics, and integration with institutional systems.

Database Design

The database should include tables or collections for users, subjects, tasks, study sessions, reminders, progress logs, and reflection notes. The user table stores profile and preference information. The subject table stores course names and learning categories. The task table stores deadlines and estimated hours. The session table stores scheduled study blocks. The progress log stores completion status and actual study time.

Data relationships should be designed carefully. A single subject may have many tasks, and a single task may generate many study sessions. A study session may produce one or more progress records. This structure allows the application to track not only whether a task is complete but also how it was completed over time.

User Interface Screens

The main screens include onboarding, dashboard, subject manager, task creation form, calendar view, daily plan, progress dashboard, reminder settings, and reflection page. The dashboard should show today's tasks, upcoming deadlines, and a short progress summary. The task creation screen should be simple, asking only for necessary information first and allowing advanced options later.

The calendar view should support both weekly and monthly views. Students should be able to drag tasks, edit time, and mark sessions complete. The progress dashboard should avoid heavy analytics jargon and instead use student-friendly language such as 'You completed 4 of 6 planned sessions this week.'

IX. Evaluation Framework

Usability Metrics

Usability evaluation should examine learnability, efficiency, error rate, satisfaction, and perceived usefulness. Learnability can be measured by observing how quickly new users create their first study plan. Efficiency can be measured by task completion time. Error rate can include failed task entry, wrong schedule changes, or confusion between task and session. Satisfaction can be measured with rating-scale questions and interviews.

System Usability Scale items may be adapted to assess overall usability. However, usability alone is not enough. A study planner must also be evaluated for educational relevance. Users should be asked whether the application helps them understand priorities, reduce missed deadlines, and feel more confident about study planning.

Learning-Support Metrics

Learning-support indicators may include schedule adherence, task completion rate, revision frequency, number of overdue tasks, and self-reported time-management confidence. These indicators can be compared before and after application use. A short two-week or four-week pilot study can provide initial evidence of usefulness.

The application should also measure plan realism. A plan is realistic when students can follow it without excessive rescheduling. If most users frequently miss generated sessions, the algorithm may be too ambitious. In that case, the planner should reduce daily workload or ask students for more accurate availability information.

Prototype Testing Procedure

Prototype testing can begin with a guided session in which participants complete a set of tasks: create a subject, add an exam, generate a plan, reschedule a session, mark a task complete, and view a report. Observers record completion time, errors, and comments. After the session, participants complete a questionnaire and short interview.

A second stage can involve real-world use for several weeks. During this period, participants use the application for actual academic tasks. The evaluation then examines usage logs, user feedback, and changes in planning behavior.

X. Expected Results and Discussion

Expected Functional Outcomes

The expected outcome is that students will be able to create more structured study plans with less manual effort. The planner should reduce the burden of deciding what to do each day by automatically distributing work across available time. Users are expected to appreciate features such as deadline warnings, daily study lists, and flexible rescheduling.

In a prototype evaluation, the most important functional result would be successful completion of core tasks. If users can add subjects, create deadlines, generate a plan, and edit sessions without confusion, the application can be considered functionally promising. Further improvement would then focus on personalization and long-term engagement.

Expected Educational Outcomes

From an educational perspective, the planner is expected to support awareness, consistency, and reflection. Students may become more aware of workload distribution when they see tasks spread across the calendar. They may become more consistent when reminders connect daily action with future deadlines. They may also reflect better when weekly summaries show missed sessions and completed work.

The application is not expected to automatically improve grades without student effort. Its realistic contribution is to support behaviors that are associated with better academic preparation. A planner can create structure, but learning still depends on attention, strategy, feedback, and persistence.

Discussion in Relation to Previous Research

The StudyFlow application aligns with literature emphasizing self-regulated learning, learning analytics, and time-management support [1]–[5]. It extends these ideas by translating them into a practical system design. Instead of presenting self-regulation as an abstract concept, the planner operationalizes it through goal creation, schedule generation, monitoring, and reflection.

The design also addresses concerns found in previous research about the need for actionable analytics. A dashboard is useful only when it helps the learner make a decision. Therefore, the proposed planner uses analytics to support next-step recommendations rather than simply displaying data.

XI. Privacy, Security, and Ethical Considerations

Data Privacy

A StudyFlow stores personal information about study habits, deadlines, and routines. Although this information may not always be highly sensitive, it still deserves careful protection. The application should collect only the data required for planning and feedback. It should not request unnecessary personal details.

Users should be able to delete their data, export their study plan, and understand what information is stored. Privacy settings should be easy to find. If the application is used in an institution, students should not feel that every delay or missed task is being monitored by teachers unless explicit consent is provided.

Security Requirements

Security requirements include user authentication, encrypted communication, secure password handling, role-based access if institutional accounts are used, and regular backup. If cloud storage is used, database rules must prevent users from accessing each other's records. Notifications should not reveal sensitive information on locked screens unless the user permits it.

The system should also protect against accidental data loss. Students may depend on the planner during important academic periods, so data backup and synchronization are important for reliability.

Ethical Use of Analytics

Learning analytics should guide students, not label or shame them. The application should avoid language such as 'poor performer' or 'lazy user.' Instead, it should use supportive feedback like 'This plan may be too heavy for your current availability.' Ethical analytics emphasizes transparency, consent, and benefit to the learner.

XII. Limitations and Future Scope

Limitations

This paper presents a detailed design and evaluation framework, but it does not report large-scale experimental results. Therefore, claims about effectiveness should be treated as proposed or expected rather than proven. A full version of the study would require user testing over several weeks or months.

Another limitation is that student behavior is highly individual. A planning algorithm that works for one learner may not work for another. Some students prefer fixed schedules, while others prefer flexible task lists. The application must therefore support personalization rather than enforcing one planning style.

Future Enhancements

Future versions can include machine learning models that learn from user behavior and recommend better session times. The system may also integrate with institutional learning management systems to import assignments automatically. Gamification features such as streaks, badges, or progress milestones may improve motivation, but they should be used carefully so that the planner remains academic and not distracting.

Additional future work may include support for group study planning, teacher dashboards, parent views for younger students, offline mode, voice-based task entry, and accessibility improvements for students with visual or attention-related needs.

XIII. Conclusion

Summary of Contribution

This paper presented the design of a StudyFlow application built to support personalized academic time management. The StudyFlow system combines task management, automatic scheduling, reminders, progress tracking, and reflective analytics. It is grounded in self-regulated learning theory and supported by research on learning analytics and digital learning environments.

The main contribution of the paper is a practical and structured model for transforming academic goals into daily study actions. The system architecture, planning algorithm, database design, evaluation framework, and diagrams provide a foundation for prototype development and future empirical testing.

Final Remarks

A StudyFlow does not replace student effort, teacher guidance, or effective learning strategies. However, it can make effort more organized and visible. By helping students plan realistically, monitor progress, and adjust without discouragement, the application can become a meaningful academic support tool. The next step is to implement the prototype, test it with real users, and refine it based on evidence.

XIV. Tables and Design Specifications

Functional Requirements

Table I summarizes the major functional requirements of the StudyFlow system. These requirements are written in a way that can be directly used by developers during prototype implementation.

Requirement ID	Feature	Description	Priority
FR-01	User Profile	Allow students to enter academic level, available hours, preferences, and reminder settings.	High
FR-02	Task Entry	Allow creation of assignments, exams, readings, projects, and revision goals.	High
FR-03	Automatic Planning	Generate study sessions based on deadline, duration, difficulty, and availability.	High
FR-04	Rescheduling	Move incomplete tasks to suitable future slots and warn when deadlines are at risk.	High
FR-05	Progress Tracking	Track completed, missed, and rescheduled sessions with weekly summaries.	Medium
FR-06	Notifications	Send reminders before planned study sessions and upcoming deadlines.	Medium

Non-Functional Requirements

Table II presents non-functional requirements. These requirements are important because a planner used daily must be fast, reliable, secure, and easy to understand.

Category	Requirement	Rationale
Usability	A new user should generate a first study plan within five minutes.	Students may abandon the app if onboarding is too complex.
Performance	Calendar updates should load within two seconds under normal use.	Slow feedback reduces trust and daily usage.
Security	Personal study data should be protected by authentication and encrypted communication.	Academic routines and personal availability are private information.
Reliability	Study plans should be backed up and synchronized across devices.	Students may depend on the plan during exams.
Accessibility	Text, icons, and navigation should follow common accessibility practices.	The tool should be usable by learners with different needs.

Sample Evaluation Indicators

Indicator	Measurement Method	Interpretation
Task completion rate	Completed sessions divided by planned sessions	Higher values suggest better plan adherence.
Overdue task count	Number of tasks past deadline	Lower values suggest improved deadline management.
Rescheduling frequency	Number of moved sessions per week	Very high values may indicate unrealistic planning.
Usability score	Post-test questionnaire	Higher values indicate easier interaction.
Confidence change	Pre/post self-rating	Positive change suggests perceived time-management improvement.

References

[1] Y. Gambo and M. Z. Shakir, "Review on self-regulated learning in smart learning environment," Smart Learning Environments, vol. 8, no. 1, 2021. https://doi.org/10.1186/s40561-021-00157-8

[2] S. Alhazbi, A. Al-ali, A. Tabassum, and A. Al-Ali, "Using learning analytics to measure self-regulated learning: A systematic review of empirical studies in higher education," Journal of Computer Assisted Learning, 2024. https://doi.org/10.1111/jcal.12982

[3] B. Tabuenca, M. Kalz, H. Drachsler, and M. Specht, "Time will tell: The role of mobile learning analytics in self-regulated learning," Computers & Education, vol. 89, pp. 53–74, 2015. https://doi.org/10.1016/j.compedu.2015.08.004

[4] O. Viberg, M. Khalil, and M. Baars, "Self-regulated learning and learning analytics in online learning environments: A review of empirical research," in Proc. International Conference on Learning Analytics and Knowledge, pp. 524–533, 2020. https://doi.org/10.1145/3375462.3375483

[5] R. P. Alvarez, I. Jivet, M. Perez-Sanagustin, and others, "Tools designed to support self-regulated learning in online learning environments: A systematic review," IEEE Transactions on Learning Technologies, 2022. https://doi.org/10.1109/TLT.2022.3193271

[6] Y. Gambo and M. Z. Shakir, "Evaluating students' experiences in self-regulated smart learning environment," Education and Information Technologies, vol. 28, pp. 11285–11309, 2023. https://doi.org/10.1007/s10639-022-11126-0

[7] E. Araka, E. Maina, R. Gitonga, and R. Oboko, "Research trends in measurement and intervention tools for self-regulated learning for e-learning environments—Systematic review (2008–2018)," Research and Practice in Technology Enhanced Learning, vol. 15, 2020. https://doi.org/10.1186/s41039-020-00129-5

[8] S. Heikkinen, M. Saqr, J. Malmberg, and M. Tedre, "Supporting self-regulated learning with learning analytics interventions: A systematic literature review," Education and Information Technologies, vol. 28, pp. 3059–3091, 2023. https://doi.org/10.1007/s10639-022-11281-4

[9] W. Matcha, N. A. Uzir, D. Gašević, and A. Pardo, "A systematic review of empirical studies on learning analytics dashboards: A self-regulated learning perspective," IEEE Transactions on Learning Technologies, vol. 13, no. 2, pp. 226–245, 2020. https://doi.org/10.1109/TLT.2019.2916802

[10] J. Wong, M. Baars, D. Davis, T. Van Der Zee, G.-J. Houben, and F. Paas, "Supporting self-regulated learning in online learning environments and MOOCs: A systematic review," International Journal of Human–Computer Interaction, vol. 35, no. 4–5, pp. 356–373, 2019. https://doi.org/10.1080/10447318.2018.1543084

[11] R. Garcia, K. Falkner, and R. Vivian, "Systematic literature review: Self-regulated learning strategies using e-learning tools for computer science," Computers & Education, vol. 123, pp. 150–163, 2018. https://doi.org/10.1016/j.compedu.2018.05.006

[12] L. Anthonysamy, A. C. Koo, and S. H. Hew, "Self-regulated learning strategies and non-academic outcomes in higher education blended learning environments: A one decade review," Education and Information Technologies, vol. 25, pp. 3677–3704, 2020. https://doi.org/10.1007/s10639-020-10134-2

Appendix A: Sample Student Requirement Questionnaire

1. How do you currently plan your study schedule during a normal week?

2. Which academic tasks do you most often delay: assignments, revision, reading, projects, or exam preparation?

3. How many hours per day can you realistically study outside class?

4. How useful would automatic schedule generation be for you?

5. What type of reminders do you prefer: push notification, email, calendar alert, or in-app alert?

6. What would make you stop using a study planner application?

This appendix is included to make the manuscript more complete and directly usable for implementation or academic submission. The exact wording can be modified based on the final application prototype and institutional requirements.

Appendix B: Sample Usability Test Tasks

Task 1: Create a new subject named Mathematics.

Task 2: Add an examination deadline two weeks from today.

Task 3: Generate an automatic study plan.

Task 4: Move one study session from evening to morning.

Task 5: Mark a session complete and open the progress dashboard.

Task 6: Explain what the weekly report tells you about your study behavior.

This appendix is included to make the manuscript more complete and directly usable for implementation or academic submission. The exact wording can be modified based on the final application prototype and institutional requirements.

Appendix C: Sample Pseudocode for Planning Engine

INPUT: subjects, tasks, deadlines, difficulty, available_slots

FOR each task: calculate urgency, difficulty weight, duration weight, and importance weight

COMPUTE priority score for each task

SORT tasks by priority score

FOR each task: split estimated duration into sessions based on maximum session length

ALLOCATE sessions into available slots before deadline

ADD revision sessions where examination tasks are detected

IF insufficient slots exist: show risk warning and recommend plan adjustment

OUTPUT: personalized study calendar

This appendix is included to make the manuscript more complete and directly usable for implementation or academic submission. The exact wording can be modified based on the final application prototype and institutional requirements.

Appendix D: Sample Test Cases

TC-01: Verify that a user can create a subject with a valid name.

TC-02: Verify that a task cannot be saved without a title and deadline.

TC-03: Verify that the planning engine avoids blocked time slots.

TC-04: Verify that overdue tasks are shown on the dashboard.

TC-05: Verify that reminders follow user quiet-hour settings.

TC-06: Verify that weekly progress percentage updates after task completion.

This appendix is included to make the manuscript more complete and directly usable for implementation or academic submission. The exact wording can be modified based on the final application prototype and institutional requirements.