Effectiveness of Academic Scheduling among College Students

CHLOE (YING-TI) YUEH*, LOGAN GREEN*, JACOB EGGERD*, FU-SHIANG YANG*, and ALI IMRAN*, University of Michigan, USA

College students spend a lot of time tracking their work by managing deadlines scattered across multiple platforms. They need a single, reliable tool that handles this organizational overhead automatically, so they can focus their energy on actual coursework. Students currently navigate an average of four to five different platforms which include Canvas, syllabi, personal calendars, and group messaging apps to piece together a complete picture of their responsibilities. This fragmentation creates a planning tax where manually transferring deadlines at the start of each semester is so time-consuming and error-prone that students abandon planning tools altogether before they even get started. UniSync addresses this through an integrated academic scheduling platform featuring automated LMS synchronization with Canvas and Gradescope, an AI-powered PDF syllabus parser that extracts deadlines automatically, and collaborative availability heatmaps for group coordination. To evaluate the core efficiency claim, we conducted a within-subjects study with 27 participants comparing manual deadline entry to AI-assisted syllabus upload on an identical assignment task. The AI-powered upload was significantly faster than manual entry, representing a 78.7% reduction in task completion time. These results validate that AI-assisted syllabus parsing can meaningfully eliminate the setup burden that prevents students from adopting planning tools in the first place.

CCS Concepts: • **Human-centered computing → Interaction techniques**.

1 INTRODUCTION

The modern academic environment requires students to navigate a complex web of deadlines distributed across various Learning Management Systems (LMS) and static documents such as course syllabi. This fragmentation leads to a significant planning where the effort required to organize tasks often rivals the effort required to complete them [2, 10]. Students currently resort to redundant tools or repetitive checking and re-checking behaviors because they lack a single source of academic deadline information. This paper presents the process behind UniSync, a centralized academic scheduling platform developed through a multi-phase human-centered design methodology. Our work began with a formative survey of 38 college students, which revealed a gap between how students currently use generative AI (i.e. for searching and summarizing) and how they wish to use it, with automation emerging as the most desired capability. A subsequent contextual inquiry with five participants gave concrete shape to these findings, documenting that students navigate an average of four to five different platforms to compile a complete picture of their academic tasks, and that the manual burden of transferring syllabus dates at the start of each semester became the primary reason students abandoned organizational tools before they could benefit from them.

These findings motivated the design of UniSync, which addresses information fragmentation through three core features which include an automated synchronization with LMS platforms like Canvas and Gradescope, an AI-powered syllabus parser that extracts deadlines from PDF documents, and collaborative availability heatmaps for group coordination. Seven user requirements were derived from the contextual inquiry.

The design process proceeded through iterative prototyping and evaluation. Individual sketches and storyboards from all five team members were consolidated into a unified low-fidelity paper prototype and evaluated using both heuristic testing and simplified user testing with five target

*All authors contributed equally to this research.

users. These evaluations exposed critical usability failures particularly around system status visibility, interaction discoverability, and transparency of AI behavior that informed a transition from automation to assistance in the high-fidelity prototype.

The paper concludes with a quantitative evaluation of the functional prototype, focused on **User Requirement 5.5**: that the time required to compile a complete list of course assignments using UniSync's AI-powered syllabus upload should be significantly less than manual transcription. A within-subjects study with 27 participants demonstrated that AI-assisted upload reduced task completion time by 78.7% compared to manual entry. These results validate the core efficiency requirement and demonstrate that transparent AI assistance can meaningfully reduce the setup burden that prevents students from adopting academic planning tools in the first place.

2 RELATED WORK

HCI research has long established that task management difficulties often stem from tool inadequacies rather than user incompetence. Bellotti et al. [2] found that while individuals possess sophisticated strategies for prioritization, existing tools frequently fail to support these mental models. In the academic context, this disconnect is particularly damaging due to the complex struggles students face in managing deadlines. Kim et al. [7] highlighted that academic delay is not uniform; rather, it manifests as distinct behavioral patterns ranging from strategic planning to harmful procrastination. Their findings suggest that current "one-size-fits-all" systems fail to address the specific vulnerabilities of students who struggle with self-regulation, leaving them without adequate support to mitigate academic risk.

Despite the diverse needs of students identified above, Learning Management Systems (LMS) like Canvas often impose rigid structures that hinder effective task prioritization. Chen et al. [4] identified significant usability challenges in Canvas, particularly regarding assignment organization and rubric design, noting that inconsistent usage by instructors further complicates student navigation. To compensate for these functional gaps, students frequently adopt external tools such as Notion or Google Calendar. However, this compensatory behavior leads to information fragmentation. Whittaker et al. [10] describe how distributing tasks across disconnected applications forces users to manually retrieve and transfer data, increasing cognitive load and the risk of missing critical deadlines due to synchronization failures.

Recent advancements in AI and cross-platform integration offer promising avenues to address the fragmentation and lack of personalization identified above. Williams et al. [11] emphasized the necessity of tools that manage task information across different contexts, a concept highly relevant to unifying students' scattered academic workflows. Furthermore, AI agents hold the potential to provide the adaptive support required by the heterogeneous student groups described by Kim et al. [7]. The emergence of large language models (LLMs) has created new possibilities for automated document processing and information extraction. Dunn et al. [5] demonstrated that conversational LLMs can achieve high precision and recall (both approaching 90%) when extracting structured information from documents, suggesting that similar approaches may be effective for parsing academic syllabi. In healthcare contexts, Chang et al. [3] showed that NLP algorithms integrated with LLMs can reduce data extraction time by factors exceeding 3,000 compared to manual chart review while maintaining high accuracy. These findings support the hypothesis that AI-assisted syllabus parsing could dramatically reduce the time students spend on manual deadline entry.

However, designing these AI-assisted systems remains non-trivial. Yang et al. [12] cautioned that Human-AI interaction must carefully balance automation with user agency to handle the uncertainty of AI outputs effectively, ensuring students remain in control of their academic planning. This principle of transparent assistance—rather than invisible automation—guided UniSync's design, particularly the three-tier data import system that allows students to verify and correct AI-extracted information.

To evaluate whether AI-powered tools genuinely improve efficiency, robust experimental methods are essential. Task completion time has been established as a primary dependent variable for evaluating system usability in HCI, particularly when comparing manual and automated approaches [6]. Lazar et al. [8] noted that within-subjects experimental designs are commonly employed in HCI studies to control for individual differences while comparing interface alternatives, as they require fewer participants than between-subjects designs while maintaining statistical power. Acuña et al. [1] conducted a comprehensive experiment with 168 users examining the impact of usability mechanisms on efficiency, effectiveness, and user satisfaction, demonstrating that well-designed automation features can significantly reduce task completion time without compromising accuracy. Similarly, Westra et al. [9] compared manual versus automated data collection methods in research contexts and found consistent efficiency advantages for automated approaches. These studies establish both the theoretical justification and methodological precedent for our within-subjects comparison of manual entry versus AI-assisted syllabus upload, which directly evaluates User Requirement 5.5.

3 ESTABLISHING FOCUS: INITIAL SURVEY

With the fast changing field of generative AI, it is difficult to track exactly how students use generative AI in their day to day lives. The team developed a survey in order to log results for how college students currently use generative AI, how much, and identify areas of improvement for that would change their usage.

3.1 Method

We developed a survey to be distributed to college students over 18 years of age. Our survey was distributed via the format of an online Google Form. The initial and final surveys are reproduced in full in Appendix **??** and **??**. The survey was initially piloted with a group of 5 individual participants. Each participant was asked to fill out the same draft of the survey. The participants were asked to "think-aloud", and the researchers collected their feedback and thoughts as they filled out the survey. These results were used to craft a final version of the survey that was distributed to the full, larger group of participants.

The survey immediately displayed a top level description describing the goals of the survey and the procedure. This description highlights the want to identify student use of generative AI so that research may identify breakdowns and areas for improvement in the interaction.

The survey contains a questionnaire aimed at a few key areas. We collect demographic information to see if there are any interesting correlations in who uses Gen AI. Then it moves into a section describing how much the student currently uses generative AI and rate types of interaction. The student can also select ways they wish they could ideally use it for in the future. We hoped to look for gaps between current uses and desired uses. Finally, we collect sentiment on whether they wish to increase their usage or not, and their stated reasons for doing so.

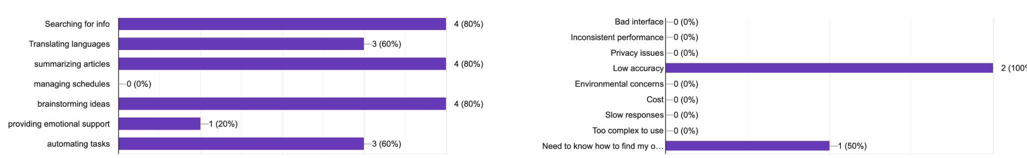
3.2 Tasks and Procedures

Consent was obtained from participants by sending messages asking potential participants if they would like to complete a survey for a course project. Participants were informed that the survey was interested in how they use Generative AI, and that the results would be anonymized. The participants were then given a link to take the online Google Form survey.

The survey further begins with a description, stating the goals, tasks, and requirements for the survey: "This survey is conducted as part of the CSE593 course project and aims to understand how college students use generative AI (such as ChatGPT, Gemini, etc.) in their academic and personal lives. Your responses will help identify common use cases, motivations, and gaps in current generative AI tools. The survey will take approximately 5-10 minutes to complete. Questions are spread across multiple pages. This study is conducted under Institutional Review Board guidelines, which require participants to be 18+ years old."

Based on the observed pilot surveys, the task took on average roughly 5 minutes to complete the survey. Quality of the tasks was ensured by requiring each question on the survey to be answered. All questions are multiple choice or select all, avoiding more variable answers to free response questions. Along with this, a starter question was added that checks for 18 years of age and that they are a college student.

3.2.1 Survey Pilot Results and Implications. Prior to widely distributing the survey, an initial survey was created to conduct a pilot 'think-aloud' process. The individual think-alouds with the 5 participants helped identify issues with question flow, clarity of the questions and answers, and imbalanced answers to one question. Additionally it highlighted many areas where we discovered we were more interested in responses. These included growing the number of demographic questions to include much more information. One of the pilot participants mentioned feeling their skills were "numbed" by Gen AI, and this was added as a response to a few questions as "diminishing skills". Looking at the survey results such as in the figure 1a, we realized we wanted to be able to see more information about each use case. We moved from a select all style to a likert scale for each question. Since many pilot respondents were worried about low accuracy as shown in figure 1b, we added a few additional questions about trust in Gen AI to gain more insight.



(a) Select all cases you currently use AI for

(b) What makes you want to decrease your Gen AI usage?

Fig. 1. Pilot survey results highlighting use cases and concerns related to Gen AI usage.

3.3 Participants

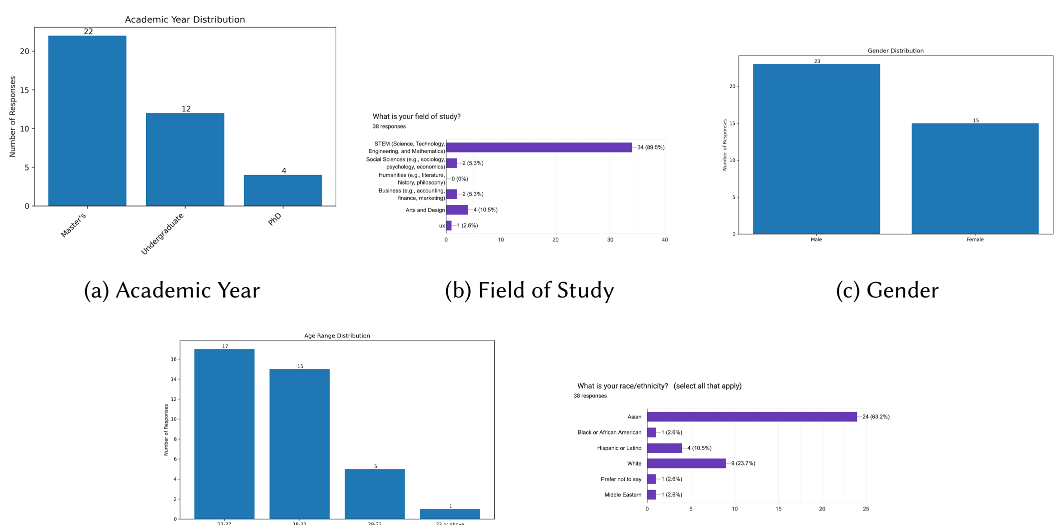
Our survey was responded to by 38 college students. The survey included several demographic questions at the beginning. The results are available in full detail in the Appendix and section 3.4.1; a short summary of the results is given here. A strong majority of our respondents, 89.5%, were STEM majors. 57.9% were Master's students, followed by bachelors and Phd. 44.7% of respondents were 23-27 years old, followed by 39.5% being 18-22. A majority of respondents identified as male at 60.5%. A majority of respondents were Asian at 63.2%. 57.5% stated they used Gen AI on multiple tasks a day, whereas only 13.1% stated rarely or never using Gen AI.

Participants were recruited through connections and messages of the researchers, which is what likely leads to a strong STEM bias in results. Usage level of Gen AI was not a recruitment criteria, as we hoped to see differences based on usage levels. No incentives were given, and survey results were collected virtually through a Google Form. The "think-aloud" sessions occurred in person with 5 participants. When each researcher reached out to their immediate connections, the survey results reached 38 participants, and we felt that the data showed clear enough trends to stop recruitment.

3.4 Results

3.4.1 Participant Overview. The survey received a total of 38 responses over five days and employed a convenience sampling strategy through the authors' personal and academic networks. The demographic characteristics of respondents were similar to those observed in the pilot study. A majority of participants were graduate students, with 57.9% identifying as master's students and 10.5% as doctoral students; the remaining 31.6% were undergraduates . Most respondents (89.5%) were enrolled in STEM fields. The sample was predominantly male (60.5%) and Asian (63.2%), and 84.2% of respondents were between 18 and 27 years old.(Figure 2)

This demographic distribution is not representative of the broader college student population and reflects the limitations of convenience sampling. Consequently, the findings should be interpreted as exploratory and indicative of patterns within this specific population rather than as generalizable results. We further discuss the implications of these limitations in Section **??**.

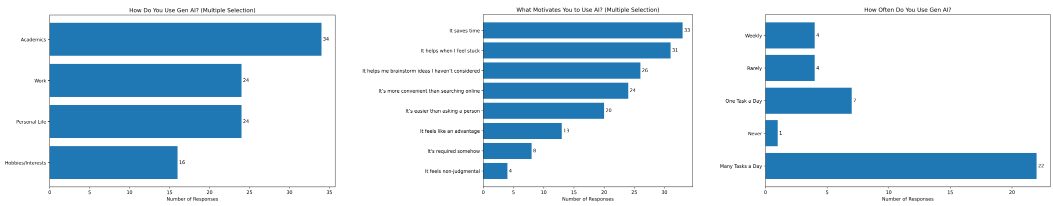


(a) Academic Year (b) Field of Study (c) Gender

(d) Age (e) Race/Ethnicity

Fig. 2. Demographic characteristics of survey respondents.

3.4.2 Usage Context and Motivations. Respondents reported using generative AI tools primarily within academic contexts. Specifically, 91.9% indicated that they use generative AI for coursework or other academic-related tasks (Figure 3a). Commonly reported motivations for use included saving time, receiving assistance when feeling stuck on a problem, and supporting idea generation or brainstorming also shown in Figure 3b. These motivations suggest that participants view generative AI as a form of cognitive support that complements existing workflows rather than as a replacement for their own effort.



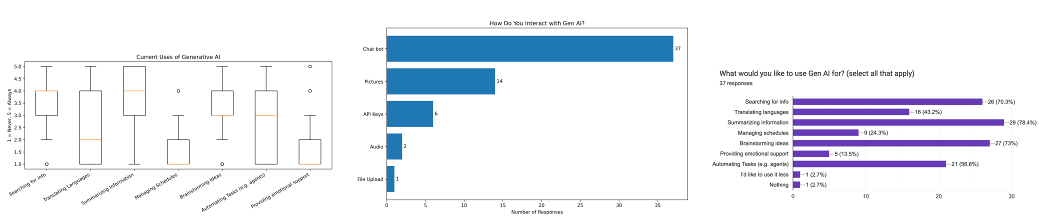
(a) Usage Context (b) Motivation (c) Usage Frequency

Fig. 3. Survey participants' usage context, motivation, and usage frequency.

All but one respondent reported using generative AI tools (Figure 3c). This non-user cited concerns about inconsistent performance and a perceived risk of diminishing their own skills. Although this perspective was uncommon within the sample, it highlights ongoing concerns related to trust, reliability, and skill degradation that have been raised in prior work.

3.4.3 Current versus Desired Uses of Generative AI. Participants' current uses of generative AI were concentrated around a small set of tasks. The most frequently reported activities included searching for information, summarizing content, and brainstorming ideas (Figure 4a). These tasks share a common characteristic: they can be completed entirely within a conversational chatbot interface. Consistent with this observation, 37 out of 38 respondents reported using a chatbot interface as their primary mode of interaction with generative AI tools (Figure 4b).

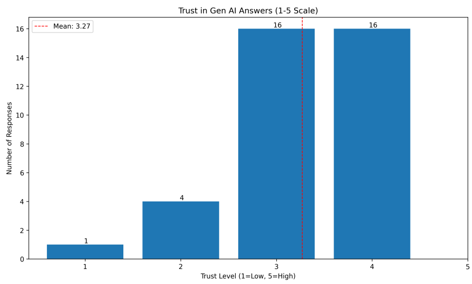
When asked about desired uses of generative AI, respondents continued to express interest in searching, summarizing, and brainstorming as shown in Figure 4c. However, many also indicated a desire to use generative AI for automating tasks. This represents a noticeable difference between current and desired uses. While automation was not a dominant current use, it emerged as an area of interest that respondents perceived as both valuable and under-supported by existing tools.



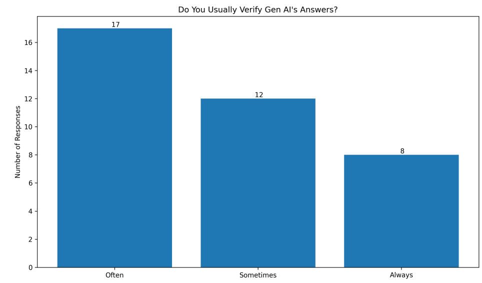
(a) Current Uses (b) Method of Interaction (c) Desired Uses

Fig. 4. Participants' current uses, primary method of interaction, and desired uses of generative AI.

3.4.4 Trust in Generative AI. Overall, respondents expressed moderate trust in generative AI tools, averaging 3.27 on a scale of one to five (Figure 5a). While many participants reported relying on AI-generated outputs for initial guidance or idea generation, they also indicated a tendency to verify or refine these outputs before using them in final work (Figure 5b). Common concerns related to trust are further revealed in Figure 6c: Reasons to decrease usage.



(a) Trust in Gen AI: mean trust score is 3.27 on a scale of 1 to 5

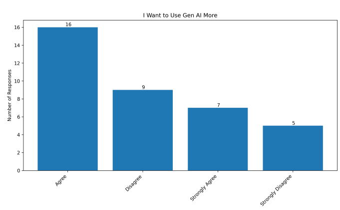


(b) Verification Habits: Respondents were provided with the options Always, Often, Sometimes, Rarely, Never

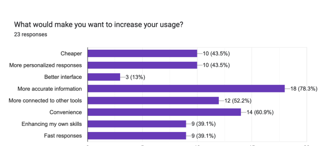
Fig. 5. Respondents' trust in generative AI and their verification habits when using AI outputs.

3.4.5 Future Intent. 23 out of 38 respondents indicated an intention to continue using generative AI tools in the future (Figure 6a) where sixteen respondents agreed that they wanted to use generative AI more; seven strongly agreed. Nine respondents disagreed, and five strongly disagreed. Interest in expanded functionality was evident, with automation emerging as a prominent desired capability. Participants' future intentions suggest that while current usage remains relatively conservative, there is openness to deeper integration of generative AI if they provide more accurate information and serve the purpose of convenience (Figure 6b).

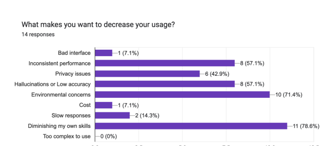
Concerns related to the users' skills diminished by AI, environmental concerns, inconsistent performance, and hallucinations are the top reasons respondents intend to decrease their usage (Figure 6c).



(a) Intended Future Usage



(b) Reasons to increase usage



(c) Reasons to decrease usage

Fig. 6. Participants' intended future usage, reasons to increase usage, and reasons to decrease usage of generative AI.

Together, these findings point to a gap between how generative AI is currently used and how users would like to use it, motivating a refined project focus on automating student tasks using generative AI systems.

sectionUnderstanding Context of Use: Contextual Inquiry

3.5 Purpose

The purpose of this contextual inquiry was to understand how students currently manage assignment deadlines and schedule meetings in their natural academic environment. By observing actual behaviors and tools in use, we aimed to identify breakdowns, workarounds, and opportunities for AI-integrated scheduling solutions

3.6 Method

We employed Contextual Inquiry, a user-centered research method that combines observation and interview in the user's natural environment (Holtzblatt & Beyer, 2014 [?]). The researcher observed the participant performing real tasks while asking questions to understand their actions and intentions. Interpretations were shared during the session for the participant to confirm or correct, ensuring accurate understanding of their workflow.

3.7 Tasks Observed

- **Responding to new assignments and deadlines** — How they respond when a new assignment is posted, check deadlines in Canvas, and record them.
- **Managing and prioritizing assignments** — How they identify upcoming assignments, add them to their schedule, evaluate which is more important, and decide when to work on tasks.
- **Scheduling meetings with others** — How they arrange meetings with classmates, professors, advisors, or external collaborators.
- **Handling schedule changes** — How they deal with changed deadlines or unexpected tasks that come up throughout the semester.

3.8 Procedure Overview

	P1	**P2**	**P3**	**P4**	**P5**
Duration	40 min	45 min	1 hour	1 hour	40 min
Setting	In-person	In-person	In-person	In-person	In-person
Recording	Voice memo	Notes only	Voice memo	Voice memo + Notes	Voice memo + Notes

3.9 Participants

	P1	**P2**	**P3**	**P4**	**P5**
Role	Graduate (Master's)	Master's (STEM)	Undergraduate (Aerospace)	Undergraduate	Graduate student, GSI, researcher
Course Load	4 courses + 1 research	3 courses + 1 thesis	research + teaching	3 courses	5 courses + non-credit research
Devices	Laptop, iPad, iPhone	Laptop	Phone, Laptop	Mac, iPhone	Laptop, iPhone
Primary Tools	Canvas, Google Calendar, When2Meet, Email	Canvas, Google Calendar, Gmail, Teams, notebook	Canvas, Slack, Google Calendar, Trello	Canvas, Google Calendar, Discord, When2Meet, Email	Canvas, Google Calendar, When2Meet, MDex, Slack, WhatsApp

4 UNDERSTANDING CONTEXT OF USE: CONTEXTUAL INQUIRY RESULTS

The contextual inquiry across five participants revealed a fragmented digital environment where the primary burden of organizing scattered information falls entirely on the individual. By observing these participants, we identified how they navigate communication across different platforms, manage diverse courses, and organize their schedules amid significant technical obstacles.

4.1 Assignment Awareness and Scheduling

Participants frequently struggle to keep track of their responsibilities because platforms like Canvas provide inconsistent information. Many participants rely on the Canvas "To-Do" list, but this view is often cluttered with expired announcements that reduce efficiency. A major breakdown occurs for assignments or discussions that lack explicit due dates; these items are pushed to the bottom of the list and easily overlooked. Notifications are also purely reactive, primarily alerting participants when grades are posted rather than providing proactive reminders to begin work. Consequently, participants must manually check several pages, including course-specific assignment tabs and grades pages, to ensure they have not missed any work.

Scheduling is almost entirely driven by deadlines, with participants reacting to the most urgent tasks first. As shown in the Consolidated Sequence Diagrams for initial semester setup (Figure **??**) and assignment scheduling (Figure **??**), participants face a bottleneck where they must check various applications just to find tasks and transfer syllabus dates into personal calendars or physical notebooks. This process is often interrupted by technical friction, such as incorrect timezone imports or timing errors when adding University events. To cope with these errors, participants perform a review each day, as seen in the sequence for task prioritization (Figure **??**). This highlights that participants do not fully trust their digital tools to provide a reliable long-term outlook, leading to constant anxiety and a fear of missing out on deadlines.

4.2 Group Assignments and Meeting Coordination

Coordination for group work is a significant challenge because assignments are often issued unpredictably, making it difficult to distinguish between individual and group tasks. Late awareness of these collaborative tasks often leads to high-stress, last-minute meetings. As illustrated in the Consolidated Flow Diagram (Figure **??**), participants must manage multiple platforms, using Slack, iMessage, or Discord for chat while managing schedules via When2Meet for larger groups.

Finding common availability is tedious because participants must check their personal schedules separately before filling out coordination tools, a step highlighted in the meeting scheduling sequence (Figure **??**). This fragmentation increases cognitive load and results in frequent coordination failures, as participants often forget to respond to scheduling prompts without repeated reminders from their peers. This confirms from the Affinity Diagram (Figure **??**) as well that participants don't prefer tools requiring heavy manual entry, yet they are currently forced into these behaviors to maintain group progress.

4.3 Class and Campus Event Schedules

Participants rely heavily on personal calendars to manage recurring events like classes, research meetings, and campus activities. At the start of the semester, they create manual placeholders for class times and locations to maintain consistency. While some student-created tools allow for calendar integration via APIs, manual corrections are frequently needed to fix timing errors. Occasionally, social platforms like WhatsApp are used to share event details with friends as a workaround to formal calendar entry, adding another layer of scattered data.

The Affinity Diagram (Figure **??**) also confirms that participants trust their personal calendars more than Canvas, even though Canvas is the official repository. Despite these efforts, the high volume of platforms makes proactive checking a constant necessity. This results in a cycle of reactive behavior where participants cannot effectively prioritize tasks based on difficulty before starting them. Because existing platforms do not support the multi-channel nature of these workflows, participants are left feeling overwhelmed and forced to rely on mental effort rather than reliable systems. The next section further discusses in detail how these affinity diagrams lead to defining our user requirements.

5 USER REQUIREMENTS

These requirements were derived from contextual inquiry sessions conducted with five undergraduate students (P1–P5) recruited through convenience sampling from the University of Michigan.

Participants were included if they were currently enrolled in at least three courses and regularly used digital tools for academic scheduling. Informed consent was obtained from all participants prior to each observation session, ensuring they understood the purpose of the study, the data collection procedures, and their right to withdraw at any time.

Requirements were synthesized through a systematic consolidation process: individual sequence models and flow diagrams from each participant were first created to capture their unique workflows, then merged into consolidated representations to identify recurring patterns and breakdowns across all participants. These consolidated models (Figures 25–29) served as the foundation for deriving the following user requirements.

Additionally, a follow-up survey ($n = 25$) was administered to validate and quantify the contextual inquiry findings. Survey data is referenced where applicable to provide triangulation of qualitative observations.

There is some common language used to establish points in the requirements. "Assignments" refers to any submission or graded activity a student submits. "Meetings" refer to any scheduled block of time by a student. "Due dates" refer to a specific time that an assignment must be submitted by.

The requirements are organized into four thematic categories based on the primary user needs they address.

5.1 Category A: Information Consolidation

These requirements address the fragmentation breakdowns observed when students must navigate multiple platforms to access academic information.

5.1.1 Requirement 5.1: Single-Source Access to Academic Information. **Requirement:** Students should be able to access all their course assignments and scheduled meetings from a scheduling tool, without consulting more than one source.

Grounding in Context of Use. This requirement directly addresses the fragmentation breakdowns observed across all five contextual inquiry participants. The consolidated flow diagram (Figure 25) revealed that students currently navigate an average of 4–5 separate platforms (Canvas, email, syllabi, personal calendars, group messaging apps) to compile a complete picture of their academic obligations.

Specifically, participant P2 stated during the observation: "I have to check Canvas, then my email, then the group chat, and sometimes I still miss things because the professor posted something on a different platform." This navigation burden increases cognitive load for students, often leading to missed assignments or meetings.

Supporting Survey Data. Our follow-up survey confirmed this finding: 88% of respondents ($n = 22$) reported using three or more platforms daily to track assignments, and 76% ($n = 19$) indicated that "having one source of truth" would significantly reduce their academic stress (Appendix B, Q3–Q4).

Design Implication. Students want one "source of truth" that aggregates information from multiple sources, eliminating the need for cross-platform navigation (Figure 27).

5.1.2 Requirement 5.2: Accurate Display of Instructor-Specified Due Dates. **Requirement:** Each assignment that a student is responsible for should display the due date specified by the instructor, when the instructor has provided one.

Grounding in Context of Use. This requirement directly responds to contextual inquiry findings that students frequently encountered fragmentation breakdowns when searching for due dates and made slips when transferring information manually. The consolidated sequence model (Figure 26) documented 12 instances across participants where due date information was either incorrectly transcribed or became outdated after instructor changes.

Participant P4 explicitly noted: "I don't trust my own calendar because I might have copied the date wrong. So I always go back to Canvas to double-check, which takes extra time."

Design Implication. By ensuring due dates are directly imported from authoritative sources rather than manually entered, the system can eliminate transcription errors and reduce the anxiety-driven rechecking behavior observed in participants.

5.2 Category B: Proactive Notifications

These requirements address the cognitive burden of constant manual rechecking and the anxiety associated with potentially missing deadlines.

5.2.1 Requirement 5.3: Customizable Advance Notifications. **Requirement:** Students should be able to ensure they receive advance notice of upcoming assignment due dates at time intervals they have defined.

Grounding in Context of Use. Participants reported needing external reminders to initiate work, as constant manual rechecking was identified as a major source of cognitive load and anxiety. The consolidated sequence model (Figure 27) showed that participants checked their calendars an average of 5–7 times per day, primarily driven by fear of missing deadlines rather than productive planning.

Participant P1 described this pattern: "I check my calendar constantly—probably every hour—because I'm always worried I forgot something. It's exhausting."

Furthermore, participants cited needing an immediate motivator to work on assignments (Figure 27). Participant P3 stated: "If I get a notification that something is due tomorrow, that's when I actually start working. Without that push, I procrastinate."

Supporting Survey Data. Survey results indicated that 92% of respondents ($n = 23$) wanted customizable reminder settings, with preferred notification times varying significantly across individuals (ranging from 1 hour to 1 week before deadlines; Appendix B, Q8).

Design Implication. Allowing students to set personalized alert intervals addresses both the anxiety-reduction need and the motivation-triggering need identified in contextual inquiry.

5.2.2 Requirement 5.4: Same-Day Notification of Due Date Changes. **Requirement:** Students should become aware of any instructor-initiated changes to assignment due dates on the same day the change is made.

Grounding in Context of Use. Participants described missing deadline changes because updates in Canvas were not surfaced proactively, forcing reactive and last-minute responses. The consolidated flow diagram (Figure 26) documented three critical incidents where participants discovered deadline changes less than 24 hours before the new due date.

Participant P5 recounted: "My professor moved an exam up by two days, and I only found out the night before because someone mentioned it in the group chat. Canvas doesn't send you a notification for that."

Design Implication. Providing same-day alerts for instructor changes gives students maximum time to adjust their plans and prevents the last-minute scrambling observed in contextual inquiry sessions.

5.3 Category C: Efficiency Improvement

This requirement addresses the significant time burden associated with initial system setup, which was identified as a primary barrier to tool adoption.

5.3.1 Requirement 5.5: Automated Assignment Compilation. **Requirement:** The time required for a student to compile a complete list of all course assignments into a scheduling tool should be less than the time required to do so by manually transcribing them from each source.

Grounding in Context of Use. This requirement addresses a critical efficiency breakdown identified in our consolidated sequence model (Figure 27). Multiple participants described the burden of manually transferring syllabus dates at the start of each semester, and several identified this as the primary reason they avoided adopting external organization tools altogether.

Specifically, participant P3 stated during the observation: "Every semester I spend about 2–3 hours copying all the due dates from my syllabi into my calendar, and I still end up missing things or making typos. It feels like wasted effort."

Similarly, participant P1 noted: "I tried using Notion last year, but setting everything up took so long that I gave up halfway through. The tool is supposed to save me time, but it cost me time instead."

Supporting Survey Data. Our follow-up survey confirmed this finding: 78% of respondents ($n = 20$) reported that "time required to set up" was a major barrier to using new planning tools, and 84% ($n = 21$) indicated they would be more likely to adopt a tool that could automatically import assignments (Appendix B, Q11–Q12).

Design Implication. Participants expressed a preference for automated or AI-assisted tools, with the explicit condition that such tools must demonstrably reduce effort rather than add to it (Figure 29, Affinity Diagram Cluster: "Tool Preferences"). This insight directly informed our decision to implement AI-powered syllabus parsing, which must outperform manual entry to justify its inclusion in the system.

5.4 Category D: Collaboration and Task Management

These requirements address the coordination challenges in group work and the psychological need for task completion confirmation.

5.4.1 Requirement 5.6: Group Availability Identification. **Requirement:** Students collaborating on a group assignment should be able to identify times when all group members are simultaneously available without needing to individually contact each member or wait for their responses.

Grounding in Context of Use. Contextual inquiry findings and the consolidated flow diagram (Figure 25) show that group scheduling consistently required multiple platforms, manual prompting, and significant delays waiting for peers to respond. The sequence model documented an average of 3.2 days between initial scheduling request and confirmed meeting time across observed group coordination events.

Participant P2 described the typical process: "Scheduling a group meeting is a nightmare. I send a message asking when everyone's free, then I wait hours or days for responses, then I have to send follow-ups to people who didn't reply, and by the time everyone answers, the times that worked don't work anymore."

Participant P4 added: "I've literally missed assignment deadlines because we couldn't find a time to meet. The coordination takes longer than the actual work."

Design Implication. Enabling students to view each other's availability directly (defined as times without scheduled meetings) eliminates the back-and-forth communication overhead and reduces the scheduling timeline from days to minutes.

5.4.2 Requirement 5.7: Visual Task Completion Status. **Requirement:** Students should be able to distinguish whether an individual assignment has been completed or remains incomplete without needing to consult an external source.

Grounding in Context of Use. Participants described persistent anxiety about whether tasks were actually finished, leading to repeated re-checking of submission portals for confirmation. The consolidated sequence model (Figure 28) showed that participants visited Canvas specifically to verify submission status an average of 2.4 times per assignment after completing their work.

Participant P3 stated: "Even after I submit something, I go back to Canvas like three times to make sure it actually went through. I know it's irrational, but I can't help it."

Additionally, participant P5 mentioned that features providing completion feedback gave them psychological satisfaction: "I love the confetti animation on Canvas when you submit. And in my to-do app, checking off items feels really good—it motivates me to do more."

Design Implication. Making completed assignments visually distinct (through status indicators, animations, or checkmarks) addresses both the anxiety-reduction need and the motivation-enhancement need identified in contextual inquiry. This feature should provide immediate, satisfying feedback upon task completion.

5.5 Summary

Table 1 provides an overview of how each requirement maps to user needs and evidence sources.

Table 1. Summary of User Requirements

Category	Req.	Primary Need Addressed	Key Evidence Source
Information Consolidation	5.1	Reduce platform fragmentation	Flow Diagram (Fig. 25), Survey Q3–Q4
Information Consolidation	5.2	Eliminate transcription errors	Sequence Model (Fig. 26)
Proactive Notifications	5.3	Reduce rechecking anxiety	Sequence Model (Fig. 27), Survey Q8
Proactive Notifications	5.4	Prevent missed deadline changes	Flow Diagram (Fig. 26)
Efficiency Improvement	5.5	Lower adoption barrier	Sequence Model (Fig. 27), Survey Q11–Q12
Collaboration	5.6	Reduce scheduling overhead	Flow Diagram (Fig. 25)
Task Management	5.7	Provide completion confirmation	Sequence Model (Fig. 28)

6 INITIAL DESIGN AND LOW FIDELITY PROTOTYPES

In this part of the project, we began by developing individual personas, sketches, and storyboards to explore diverse solutions for the scheduling breakdowns identified in our contextual inquiry. Through rigorous design critique sessions, we evaluated these divergent concepts to identify the most effective features while addressing critical vulnerabilities related to system trust, information overload, and privacy. Finally, we consolidated these refined insights to synthesize a single, unified persona and develop *UniSync*, our final interactive digital paper prototype.

6.1 Individual Sketches & Storyboards Design Critique

Logan's design **??** was based around linking many different sources of assignments like Canvas and Gradescope, and then having a sync button that would update a calendar with all of the student's assignments. Members liked that Logan's design incorporated many different sources outside of just Canvas. Some members wondered if the design would display the last sync time, so that students knew how updated to date their calendar information was. Logan's design emphasized not having any manual entry from the user in order to make syncing as fast as possible. The group wished there was a verification stage or some kind of manual entry feature, so that any mistakes or omissions by the sync could be resolved.

Jacob's design **??** was focused around being able to draw assignments from very unstructured data like PDFs or free text with Gen AI, and could also be linked to sites like Canvas. The group liked the idea of being able to parse assignments from sources like syllabus PDFs, since many classes do not have a proper Canvas page. Group members wished there was more consideration for the inaccuracy and hallucinations of Gen AI, when the calendar needs to be trusted. Members wondered if the application could also include a more direct link system like Canvas when possible. Lastly, the group liked the implementation of mobile notifications through an existing calendar system. They thought it was more effective than notifications in the app like some other designs.

Chloe's design **??** focused around showcasing a simple main dashboard, listing assignments in the order of due dates, with check boxes that could be filled to indicate assignments were completed. Group members liked the alert system and the ability to set custom alert times. Group members liked the ability to switch between multiple group calendars. However, group members wished that individual calendars would be kept anonymous for privacy reasons. Group members wondered how Chloe's design has assignments uploaded and connected to the service.

Shawn's design **??** focused on having a main dashboard with three separate sections. Group members liked the "Today's Focus" section, which used Gen AI to auto order the day's tasks in importance. However, some wondered if having multiple lists and a calendar displayed could be overwhelming. Members liked the Canvas authorization page design, along with clicking on tasks bringing up a box with all the details from Canvas and a link back to the Canvas assignment. The color coding and organization, with assignments with no known due date highlighted, was also liked. Lastly, members liked that the group calendar feature could suggest meeting times, however they wondered if they could also manually create their own meeting time.

Ali's design **??** had a main dashboard that listed upcoming TODOs, remaining TODOs, completed TODOs, and upcoming classes and meetings. Members liked the inclusion of upcoming meetings and classes, as the other designs had a heavy focus on assignments. They also like the urgency color coding for upcoming assignments, but wondered if user preferences could also be accounted for in this labeling. The group liked that students had a list of completed assignments they could see, so that they are reminded of their accomplishments. Members also liked the visualization of the one-time link to Canvas with account creation.

6.2 "Final" Persona(s)

Primary Persona: Tammi Glen. **Academic Standing:** Graduate / Upper-level Undergraduate Student.
Course Load: 4–5 heavy academic courses plus research project or teaching (GSI) responsibilities.
Primary Devices: High-end laptop for primary work and a smartphone for reactive notifications.

Current Tools Used: Canvas, Google Calendar, When2Meet, Slack, Discord, and Email.

*Background
Tammi Glen is a high-achieving, technically proficient student who views academic success as a matter of precision and integrated systems. Despite being comfortable with web tools, she is managing a significantly higher workload than in previous years and feels overwhelmed by the volume of administrative tasks. She values a professional self-image and prefers sleek, minimalist designs that eliminate "administrative noise" rather than wasting cognitive energy on basic logistics.
*Pain Points and Breakdowns

- **Information Fragmentation:** Tammi is forced to navigate a rigid digital ecosystem across Canvas, Gradescope, and multiple messaging apps like Slack or Discord.
- **Disjointed Workflow:** No single tool provides a unified, low-effort view of everything needed to act, leading to a reliance on mental memory for time-sensitive information.
- **Manual Data Entry:** Every semester begins with hours spent manually transferring assignment dates from various syllabi and Canvas pages into a personal calendar.
- **System Fragility:** This repetitive process is time-consuming, prone to error, and not robust to unexpected updates from professors.
- **Anxiety & Constant Checking:** Tammi experiences a constant fear of missing "hidden" deadlines or last-minute changes.
- **High Cognitive Load:** This leads to a heavy mental load, requiring her to proactively check Canvas course tabs and emails multiple times a day.
- **Coordination Inconvenience:** Scheduling group meetings requires switching between group chats and When2Meet while manually cross-referencing personal availability.
- **Project Stalling:** Collaborative work frequently stalls while waiting for peers to remember to fill out their availability polls.

*Goals

- **Stay Organized:** Maintain a unified view of all deadlines across five courses without manually copying information from Canvas or documents.
- **Minimize Tool Switching:** Reduce the number of disconnected applications and browser tabs required to stay updated throughout the day.
- **Automate Logistics:** Coordinate group project meetings and sync deadline changes automatically to focus cognitive energy on high-level research and coursework.
- **Reliable Notifications:** Receive immediate, accurate alerts when course information or due dates change so plans can be updated instantly.

User Requirements.

- **Unified Visibility:** Every assignment and meeting for each course must be visible in one centralized tool.
- **Automated Syncing:** The tool must pull data directly from Canvas and syllabi to populate the calendar faster than can be done by hand.
- **Group Availability Heatmap:** Automatically display the free time on group members' calendars without having to prompt them individually.
- **Urgency-Based Prioritization:** Surface a shortlist of the most actionable items ordered by urgency or deadline requirements.
- **Visual Confirmation:** Provide immediate visual feedback for completed tasks to reduce re-checking anxiety.

6.3 Final Sketches and Storyboards

6.3.1 Final Sketches. The following eight sketches depict the key screens and interaction flows of UniSync as experienced by **Tammi Glen**, our final primary persona — a first-year graduate student and research assistant managing five courses simultaneously. Each screen directly addresses one or more of Tammi's core pain points: information fragmentation across disconnected platforms, vulnerability to missed deadline changes, and the burden of manual group scheduling coordination. The complete annotated sketches are provided in Appendix D.5.

Final Sketches Description. Based on the design critiques conducted across all five individual submissions, our group converged on a unified web-based application called **UniSync**. The final design synthesizes the key features from each individual proposal while directly addressing the weaknesses identified during critique.

Evolution of the Task Management Dashboard. Ali's Kanban-style dashboard received positive feedback for its intuitive layout and automated task completion tracking, and it became the foundation for the final design's task tab. We adopted a three-column structure — To Do, Upcoming, and Done — with tasks sorted chronologically by due date and color-coded by urgency. The critique of Logan's design noted the absence of a sync status indicator; the final design therefore adds a "Last synced" timestamp in the top navigation bar so users can verify data freshness at a glance. The same critique also called for the ability to manually confirm or modify scheduled items. In response, the final design adds a task detail modal accessible by clicking any task card, where users can view the full assignment description and navigate directly to Canvas via an "Open in Canvas" button, while also retaining the ability to manually add or edit tasks in the Syllabi tab.

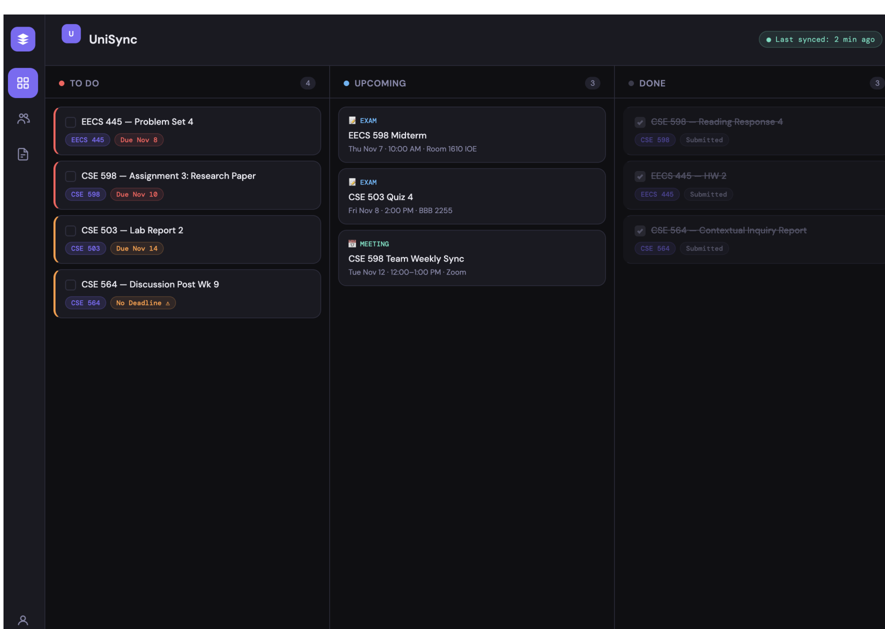


Fig. 7. UniSync Task Management Dashboard.

Evolution of the Data Import Method. The five individual designs diverged most significantly in how they imported assignment data. Logan used a manually triggered sync button connecting

Canvas and Gradescope; Jacob parsed syllabus PDFs using generative AI; Chloe and Shawn pulled data automatically from Canvas via direct sync and OAuth API authorization respectively; Ali authorized Canvas access at sign-up. Critiques repeatedly pointed to the limitations of single-platform integration: the critique of Shawn's design suggested adding connections to Gradescope and other platforms, while the critique of Ali's design noted it did not account for tools beyond Canvas. Simultaneously, the critique of Shawn's design specifically recommended a manual syllabus upload pathway for courses without a Canvas page — a feature that aligned with the well-received upload functionality in Jacob's design. The final design therefore integrates multi-platform API syncing, drag-and-drop syllabus upload, and comprehensive manual override capability, forming a three-tier data import architecture.

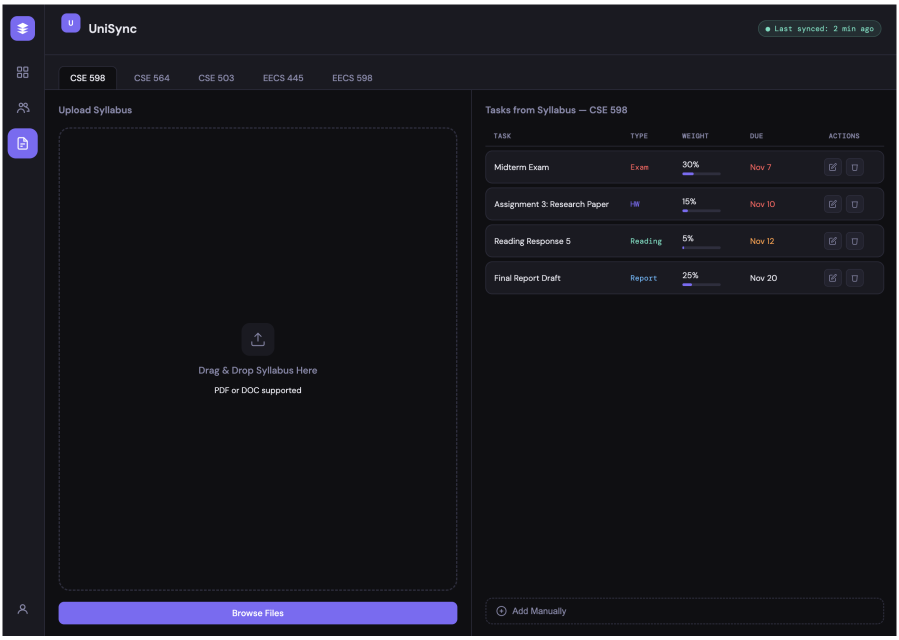


Fig. 8. Syllabi & Data Tab.

Evolution of the Group Scheduling Feature. Chloe's availability heatmap was recognized as a distinctive feature across all five designs, and Shawn's design extended it further with AI-ranked recommendations and conflict detection. However, the critique of Chloe's design raised privacy concerns, requesting that teammates' calendar events be displayed anonymously. The critique of Shawn's design further noted that users should be able to select a meeting time beyond what the AI recommends. The final design retains the When2Meet-style heatmap, connecting to members' personal calendars via API authorization but displaying only availability status without exposing individual event names. The AI Rec. button provides an maximum availability slot recommendation, while clicking any cell on the grid preserves full user agency in selecting an alternative time. A confirmation modal gives users a final review step before any meeting is scheduled.

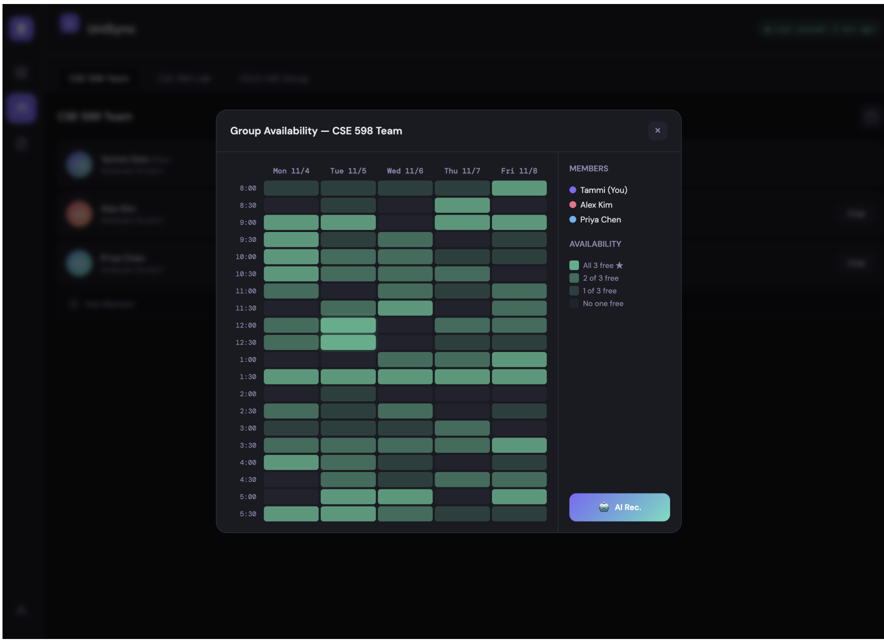


Fig. 9. Group Availability Heatmap.

Evolution of the Syllabus Management Tab. The syllabus upload feature in Jacob's design received positive feedback, but critiques expressed concern over AI parsing accuracy and suggested providing more direct links back to Canvas. The critique of Shawn's design similarly highlighted the need for a manual upload pathway for courses without a Canvas presence. The final design's Syllabi & Data tab adopts a drag-and-drop upload interface supporting PDF and DOC formats. Automatically parsed tasks can be individually edited using a pencil icon or deleted using a trash icon, and an Add Manually button allows free-form entry. This creates a hybrid architecture in which automatic parsing and full manual control coexist.

Evolution of the Profile Tab. The critique of Ali's design noted that it did not account for platforms beyond Canvas and provided no clear explanation of how groups are created. The final design introduces a dedicated Profile tab that centralizes the status of all connected platforms — Canvas, Google Calendar, Gradescope, and others — giving users complete transparency over the data sources integrated into the system.

6.3.2 Final Storyboards. The final storyboard follows Tammi Glen through a sequence of ten frames, illustrating how UniSync resolves the fragmentation and inefficiencies he encounters across a typical week of graduate study. The complete storyboard with frame-by-frame descriptions is provided in Appendix D.6.

Final Storyboards Description. The final storyboard centers on Tammi Glen, depicting how a graduate student managing a heavy course load and research responsibilities uses UniSync to resolve information fragmentation and eliminate manual academic management. Compared to the storyboards in the individual submissions, the final version represents an evolution across several dimensions.

Expanded Scenario Depth. Most individual storyboards focused on a single scenario: Shawn's seven-frame storyboard centered on the Canvas sync flow, while Chloe's storyboard covered only the group coordination scenario without addressing personal task management. The final storyboard integrates both personal task management and group scheduling into a single narrative arc, and adds a third scenario — the edge case in which a professor does not use Canvas, requiring the user to upload a syllabus manually. This scenario was absent from all five individual submissions, yet directly responds to the critiques of Jacob's AI parsing reliability, as well as the critiques of Shawn's and Logan's designs for lacking a fallback mechanism.

Fig. 10. Frame 1: Stress and Fragmentation..

Narrative Structure with Obstacles and Resolution. The final storyboard introduces a moment of failure in Frame 4 — an empty folder with a question mark — creating narrative tension before resolving it in Frame 5 through the manual syllabus upload. This structure produces a more realistic depiction of the user experience and directly responds to the shared critique across submissions that edge cases and failure scenarios were not accounted for.

Concretizing the Deadline Change Notification. Shawn's design included a deadline change notification feature, but his storyboard did not depict this scenario in action. The final storyboard's Frame

7 explicitly shows a notification card appearing on screen with the old and new dates displayed side by side, followed by the user confirming the update in a single tap. This brings to life what all five critiques consistently identified as a core user need, grounding it in a concrete interaction sequence rather than leaving it as an implied system capability.

Fig. 11. Frame 7: Proactive Deadline Alert.

Substantive Resolution. The final storyboard's closing frame shows Tammi having successfully navigated a deadline change, completed a syllabus upload for a course outside Canvas, and scheduled a group meeting without sending a single message to teammates. The resolution articulates what the user has concretely accomplished through UniSync, rather than simply demonstrating what the system can do.

Fig. 12. Frame 10: Resolution — Task Consolidation.

6.4 Final Paper Prototype

6.4.1 Final Group Design: UniSync. The final group design, *UniSync*, is a digital low-fidelity prototype implemented as an interactive browser-based shell (see Figures 13 to **??**). While it utilizes HTML/CSS to simulate the visual environment of a student dashboard, it is functionally low-fidelity containing no backend. This approach allows us to test complex user workflows without premature engineering. The *UniSync* interface represents a consolidation of five individual designs:

- **Visibility of Task and System Status** (Addressing critiques to Logan and Ali's Design): To reduce the anxiety of information fragmentation, we included a "Last Synced" status and urgency-based color coding.
- **Automation and Verification** (Addressing critiques to Jacob's design): While we automate data entry via the Syllabus Parser, the user is also able to manually enter tasks.
- **Privacy first availability** (Addressing critiques to Chloe's Design): The Team Tab utilizes an anonymized heatmap, displaying aggregate availability without revealing individual personal events.
- **Automated Priority** (Addressing critiques to Shawn's Design): The dashboard allows for reordering based on urgency of tasks.

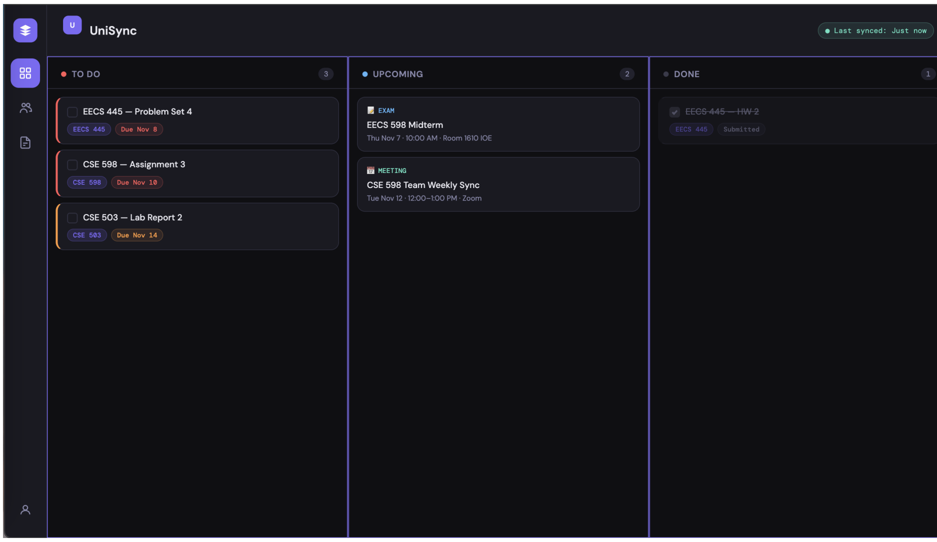


Fig. 13. **Visibility of Task and System Status:** the dashboard includes a "Last Synced" status on the top right and color-coded tasks.

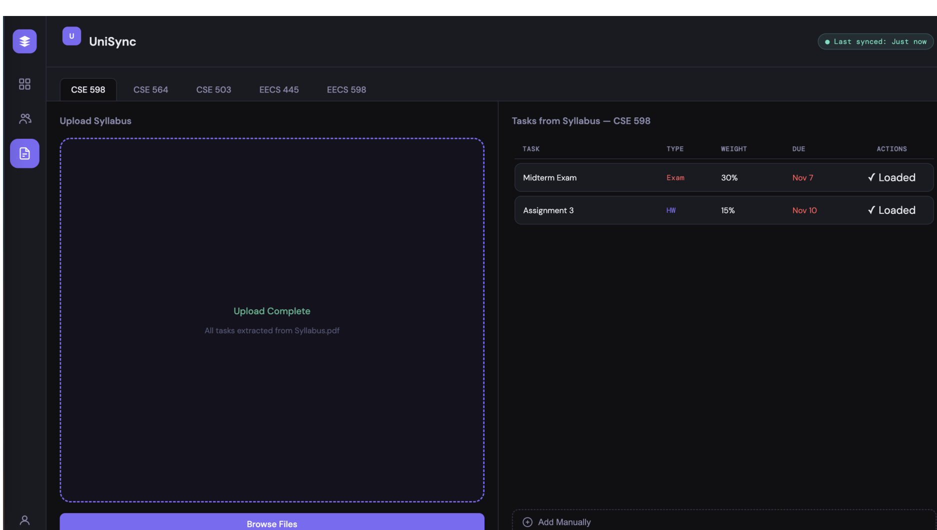


Fig. 14. **Automation and Verification:** On the left, the Syllabus Parser automatically extracts tasks. One the right, the user is also able to manually enter tasks with the "Add Manually" button.

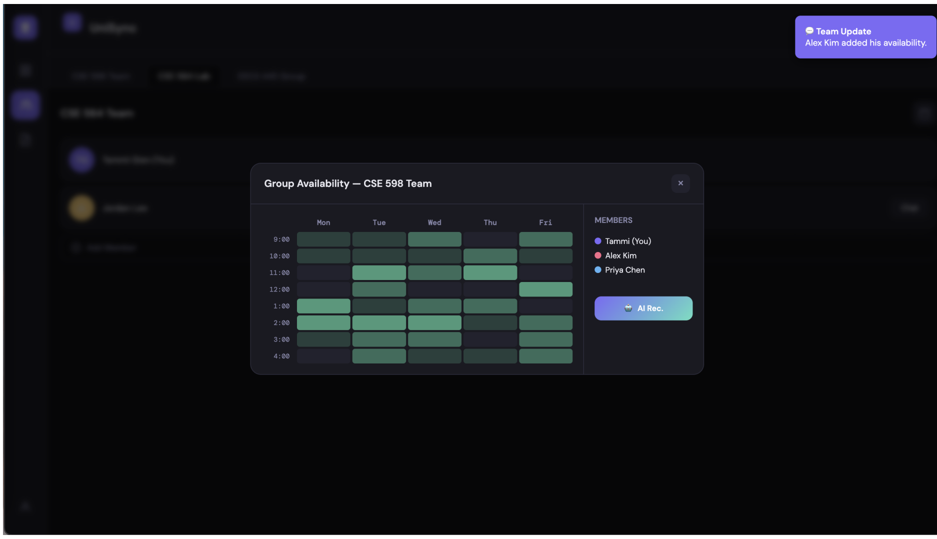


Fig. 15. **Privacy first availability:** An anonymized heatmap displaying aggregate availability without revealing individual personal events.

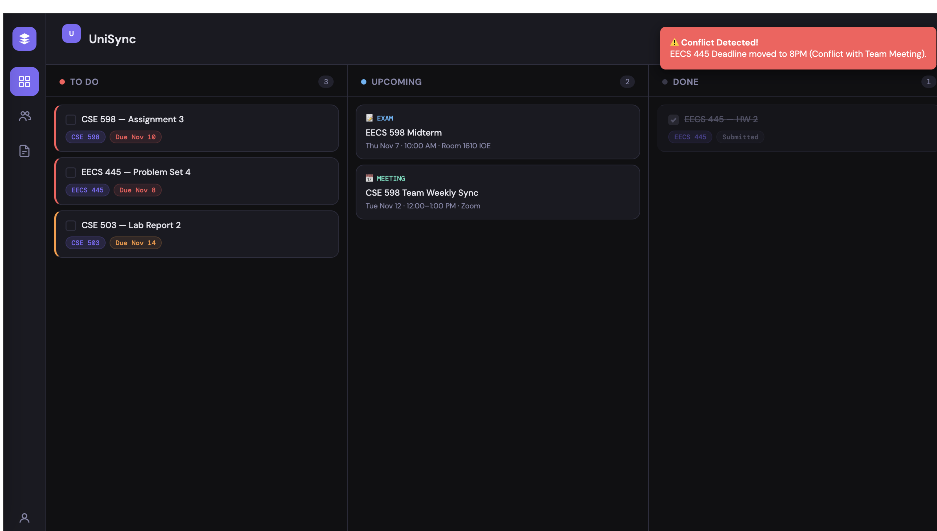


Fig. 16. **Automated Priority:** The dashboard allows for reordering based on urgency of tasks.

6.5 Wizard of Oz Implementation

In order for our design to be evaluatable, we designed the static user interface and exported to `Unisync.html` file, which acts as a digital paper prototype where interactive features are manually triggered by an investigator using hidden hotkeys on a second wireless keyboard.

6.5.1 Investigator Protocol. The investigator uses the following keyboard triggers to simulate the system's autonomous intelligence during evaluation:

- **Initial Data Load Simulation:** The dashboard initializes in an empty state. The investigator presses L to simulate a successful connection to external APIs, instantly populating the task columns to test the user onboarding experience.
- **Syllabus Parsing Simulation:** When a user drops a file into the drop-zone, the investigator presses U. This updates the UI to show "Upload Complete" and "Tasks extracted from Syllabus.pdf". A visual flash on the task table provides feedback that the AI has successfully populated the list.
- **Live Sync Simulation:** To simulate the system's ability to pull live data from external APIs like Canvas, the investigator presses S. The sync badge changes to "Syncing Canvas..." before updating to "Last synced: Just now", testing the user's trust in the data's freshness.
- **Team Notification Simulation:** The investigator can press N to trigger an alert simulating a real-time message from a teammate (e.g., "Team Update: Alex Kim added his availability."). This allows us to observe how users react to collaborative updates within the workflow.
- **AI Prioritization & Conflict Simulation:** The investigator can press P to simulate an AI optimizer that visually re-prioritizes the task list, or press C to trigger a simulated Canvas deadline conflict alert. This tests user trust and adaptation to dynamic schedule changes.

The prototype is low-fidelity with the "Wizard of Oz" technique because it prioritizes simulating and evaluating the experience of our primary persona, and requires a human investigator to simulate interaction.

7 HEURISTIC EVALUATION

The purpose of the heuristic evaluation was to identify usability problems in the UniSync low-fidelity prototype by having usability experts inspect the interface against Jakob Nielsen's 10 usability heuristics. This method allows us to detect potential issues early in the design process without requiring large-scale user testing, serving as a cost-effective complement to our simplified user testing.

7.1 Method

We conducted a heuristic evaluation following Nielsen's discount usability engineering approach. Each evaluator independently inspected the UniSync prototype and assessed it against all 10 of Nielsen's usability heuristics. For each heuristic, the evaluator either identified a usability problem or noted that no issue was found. Each identified problem was assigned a severity rating on a scale of 0 to 4:

- **0** – Not a usability problem
- **1** – Cosmetic problem; fix only if extra time is available
- **2** – Minor usability problem; fixing this should be given low priority
- **3** – Major usability problem; important to fix, should be given high priority
- **4** – Usability catastrophe; imperative to fix before product release

7.2 Tasks and Procedures

Each evaluator was given access to the UniSync low-fidelity prototype and asked to freely explore the interface. They were instructed to go through each of the 10 heuristics systematically, evaluate whether the prototype violated each heuristic, describe the usability problem if one was found, and assign a severity score. All evaluators followed the same protocol established by the team prior to evaluation.

7.3 Participants

Five usability experts participated in the heuristic evaluation. Each team member (Shawn, Logan, Ali, Jacob, and Chloe) was paired with one assigned expert who conducted the evaluation independently.

8 HEURISTIC EVALUATION RESULTS

The five heuristic evaluations yielded a total of 27 unique usability problems. Below is a compiled summary of all issues organized by heuristic, with duplicate issues noted.

8.1 H1: Visibility of System Status

#	Usability Problem	Severity	Evaluator(s)
1	After syncing Canvas, only the sync status badge updates but no actual changes are reflected in the task list or UI to confirm what was synced.	3	Shawn
2	The evaluator had significant trouble locating the sync button, preventing the user from starting the core workflow.	3	Ali
3	AI priority sort silently reorders tasks without feedback or explanation of sort rules. Users are not told that the order changed or why. (Also relates to H6.)	3	Chloe

8.2 H2: Match Between System and the Real World

#	Usability Problem	Severity	Evaluator(s)
4	It is unclear what the icons on the navigation menu mean.	1	Jacob
5	Some icons are unclear, such as using a file icon for adding tasks and a calendar icon for scheduling meetings.	2	Ali
6	AI Rec button label in heatmap modal is ambiguous. Users could not predict whether it means recommendation, record, or another action. (Also relates to H6.)	2	Chloe
7	Icon-only sidebar with no text labels forces users to guess the meaning of each icon. (Also relates to H6.)	3	Chloe

Note: Issues 4, 5, and 7 overlap—multiple evaluators identified unclear or icon-only navigation as a problem.

8.3 H3: User Control and Freedom

#	Usability Problem	Severity	Evaluator(s)
8	In the syllabus page, users cannot switch between course tabs.	4	Shawn
9	No way to undo a completed task.	4	Jacob
10	Team panel becomes too long with no way to archive or hide inactive groups.	1	Logan
11	Navigation tools like "x" buttons and back/forward links are present but basic.	1	Ali
12	Conflict alert auto-dismisses after 5 seconds with no action button. Users are notified of a conflict but given no way to resolve it. (Also relates to H9.)	3	Chloe

8.4 H4: Consistency and Standards

#	Usability Problem	Severity	Evaluator(s)
13	It is unclear what the labels for the sections of the dashboard mean.	2	Jacob
14	Evaluator confused by "Connected Tools" and how to link real-world accounts.	4	Logan

8.5 H5: Error Prevention

#	Usability Problem	Severity	Evaluator(s)
15	No validation for what file type the user uploads for the syllabus; no confirmation dialog before upload.	1	Shawn
16	Changing a due date triggers a conflict error, but the system provides no way to avoid or fix it within that view.	2	Ali

8.6 H6: Recognition Rather Than Recall

#	Usability Problem	Severity	Evaluator(s)
17	Users must navigate back and forth between the task screen and home dashboard to confirm a task was added.	2	Ali
18	No search, filter, or sort control for tasks. Users cannot manually organize their task list. (Also relates to H7.)	2	Chloe

8.7 H7: Flexibility and Efficiency of Use

#	Usability Problem	Severity	Evaluator(s)
19	Users cannot manually tune the AI prioritization. No configuration page to adjust AI assistant behavior.	2	Shawn
20	No keyboard shortcuts for navigating the dashboard.	2	Jacob
21	More preferences needed for calendar availability.	2	Jacob
22	Lack of a "quick-add" feature on the home screen slows down the workflow for experienced users.	1	Ali
23	Profile page is static and needs to be dynamic.	1	Chloe
24	No ability to sort, filter, or group tasks on the main dashboard.	1	Jacob

Note: Issues 18 and 24 overlap—multiple evaluators noted the absence of task filtering and sorting.

8.8 H8: Aesthetic and Minimalist Design

#	Usability Problem	Severity	Evaluator(s)
25	Kanban-style board fills with trivial tasks, making it hard to find important items.	4	Logan

8.9 H9: Help Users Recognize, Diagnose, and Recover from Errors

#	Usability Problem	Severity	Evaluator(s)
26	Conflict alerts do not provide enough guidance on how to resolve the problem (e.g., no drop-down list of possible fixes).	2	Shawn
27	Clear error screens are present, but there is no clear path for the user to recover from the error state.	1	Ali

Note: Issues 12 and 26 overlap—multiple evaluators identified that conflict alerts lack actionable resolution guidance.

8.10 H10: Help and Documentation

#	Usability Problem	Severity	Evaluator(s)
28	No manual or help page explaining how each panel works.	4	Shawn
29	No documentation on how to link to other apps.	2	Jacob
30	A dedicated documentation or help menu is not currently visible in the prototype.	1	Ali

Note: Issues 28, 29, and 30 overlap—multiple evaluators flagged the absence of help documentation.

8.11 Summary of Severity Distribution

Severity	Count	Key Issues
4 (Catastrophe)	5	Syllabus tab navigation, undo task, Connected Tools confusion, Kanban clutter, no help page
3 (Major)	5	Sync feedback, sync button discoverability, AI reordering, icon-only sidebar, conflict auto-dismiss
2 (Minor)	11	Various issues across multiple heuristics
1 (Cosmetic)	6	Minor polish items

The most critical areas requiring immediate attention are the lack of help documentation (H10), limited user control and freedom (H3), and insufficient system status feedback (H1).

9 SIMPLIFIED USER TESTING

9.1 Purpose

Simplified user testing was employed in order to get target users to interact with the prototype, and get early feedback on both the qualitative requirements and the usability of the prototype. By utilizing feedback on the prototype, the team is able to quickly iterate and identify problems. The team was able to create a low fidelity prototype with select features targeted at identifying problems with these requirements:

Requirement 1. Every assignment and meeting for each student should all be visible in one tool.

Requirement 4. If a professor changes or creates a due date for an assignment, the student should receive an alert that day.

Requirement 6. Students in a group together should be able to know the free time on other students' calendars without having to prompt them.

Requirement 7. Completed assignments should appear different from incomplete assignments to provide students with immediate confirmation.

9.2 Method

Simplified user testing was used here to evaluate our design against qualitative user requirements and to identify usability issues in our paper prototype, with a target user of the design. The team recruited 5 total participants to find a majority of the usability issues with the current paper prototype. The user was presented with goals that the team designed the prototype to accomplish. They were then asked to try to complete these goals with the prototype, without any help from the designer other than an outline of the goal. While attempting to complete these goals, the user was asked to "think-aloud". This allows the designer to understand what the participant thinks when using the prototype and what the strengths and weaknesses of the prototype may be. The designer can interpret notes from the test into usability problems and if the design meets the requirements, or if the requirements are insufficient. From the "think-aloud" comments of the user, the team member can decide on their interpretation of what the user really meant. The design and tasks given were meant to focus on evaluating the user requirements from our previous team assignments.

9.3 Tasks and Procedures

9.3.1 Informed Consent. Each team member contacted a participant from their network that met our inclusion/exclusion criteria for the testing. The group members messaged them and gave them an overview of the study and the benefits and risks of participating. They were informed that they would be testing a low level prototype for a course project to help college students with scheduling. The risks were described as taking their time and potential frustration from the difficulty of interacting with the prototype. The benefits were described as potentially helping other students schedule in the future, but likely none. They were informed that they could stop participating in the study at any time, for any reason.

9.3.2 Evaluation, Prototype, and Tasks. The investigator met the participant in-person with the HTML paper prototype open on a laptop, for the user to interact with. The investigator gave the participant an overview of the prototype, what the tasks for them to complete would be, and a video of the think-aloud procedure for the participant to follow. They informed the participant to complete the goal to the best of their ability, and if they encountered any issues it was the fault of the design. The participants were shown the video from lecture on how to perform the think-aloud procedure.

The investigator gave the overview of UniSync as a tool that was meant to help centralize management of assignments and group meetings for each of the students. The paper prototype was introduced as being low fidelity, with the focus of features being how assignments are synced, assignments are uploaded, and meetings are scheduled.

Each of the investigators gave the same overview of the project and the same three tasks to the participants. The tasks were presented to the participants according to the same script for each participant:

Task 1: Initial Setup and Sync Facilitator Prompt: "Imagine you are logging into UniSync for the first time this semester. Your dashboard is currently empty. Please find a way to sync your existing Canvas and Gradescope data to the dashboard." Expected User Action: The user observes the empty dashboard and attempts to initiate a sync (e.g., clicking the status badge). Investigator Action: Press L to execute the bulk-load script, populating the To-Do, Upcoming, and Done columns.

Task 2: Syllabus Parsing Facilitator Prompt: "You just received a PDF syllabus for your CSE 598 class, and the professor hasn't put the assignments on Canvas yet. Use UniSync to automatically extract and add these deadlines to your calendar." Expected User Action: The user navigates to the "Syllabi & Data" tab and interacts with the "Upload Syllabus" zone. Investigator Action: The file will automatically open on a new tab after the user drags it to the "Upload Syllabus" zone. When the user navigates back, press U to populate the syllabus task table and display the success state.

Task 3: Group Coordination Facilitator Prompt: "You need to schedule a 1-hour weekly sync with your CSE 598 project team. Please navigate to your team's page and find the best available time based on everyone's calendar." Expected User Action: The user navigates to the "Teams" tab, opens the scheduling modal, and generates the availability heatmap. Investigator Action: Allow the user to analyze the heatmap. While they are interacting, press N to trigger a live notification ("Alex Kim added his availability") to test reactions to asynchronous team updates.

9.3.3 Testing Procedure. Each of the investigators had the participants perform the think-aloud. The investigator would ask the participants to perform each task one by one. If the participant struggled the investigator would not help, but would let them know they could move to the next task or stop. The investigator would take notes during the testing and interpret them with follow-up questions after the participant did each task. Follow-up questions focused on why they either struggled or succeeded at the tasks. The investigator tried to understand what parts of the interface helped the interaction or hindered the interaction.

9.3.4 Participants. The members of the team each recruited one participant through their network, for a total of 5. Each participant met the project's inclusion/exclusion criteria and was similar to

Fig. 17. **Affinity Diagram:** Categorization of user interpretations highlighting visibility of system status and interaction discoverability.

our primary persona. The criteria is that the student is at least 18 years old and a college student. Our project targets students with a high course load that need additional help scheduling. Each of the 5 participants were college students that believed they have a high course load during the current semester.

9.4 Results- Interpretations

9.4.1 Interpretation Summary. Through creating an affinity diagram of the interpretations Figure 17, two main categories of usability problems emerged. The first is that the system status is not visible enough to the user on various different interactions and menus across the prototype. The second is that the interactions in the tasks given to the participant are too difficult to find. Moving forward, the prototype should focus on explaining the system status, and provide clearer markings for interactions throughout the design.

9.4.2 Issues with Findability. Multiple users had issues finding different features throughout the tasks. A participant struggled to identify that the sync status symbol was also a button (Appendix Table **??**-13). They believed that the status indicator was only a status indicator and not also a button. A participant also had trouble finding the calendar button for scheduling, and had to search through multiple windows in order to find it (Appendix Table **??**-39). In interpretation 2, it was noted that the participant had trouble remembering how to navigate through the different menus via the side panel. Text could potentially be added in order to provide more clarity and lower the amount of recall required.

9.4.3 Issues with System Status Visibility. The first issue encountered by many users was that the status of the main dashboard is not immediately obvious. Participants were uncertain as to if they needed to connect the tools directly or try to find a sync button (Appendix Table **??**-23). Participants also struggled to figure out what to do after uploading assignments from the syllabus. They question whether they had to manually check them after adding or if they were done (Appendix Table **??**-27). When scheduling a meeting in task 3, users struggled to understand what the color coding of meeting times meant and whether they had successfully scheduled the meeting. Users did understand that the color coding probably meant how many people were free, but wanted more feedback to confirm this (Appendix Table **??**-11). They also wanted to understand whether the meeting had actually been put on other students' calendars (Appendix Table **??**-34). Lastly, one participant didn't want to use the AI scheduling feature, because it was unclear how it was making decisions (Appendix Table **??**-30). Making the decision process clear to users would increase confidence.

9.5 Results- Requirements

9.5.1 Requirement Failure Analysis: Notes are matched to requirement failures in Table **??** of Appendix I.8.

Requirement 1

Each of the investigators found partial issues with the paper prototype for requirement 1. For instance, P01 had to go through several tabs and search for a sync button to load in their data to the main dashboard. Each of the investigators noted that the current prototype struggles with the ability to find the sync feature, making it difficult to fulfill the requirement of making all assignments visible on the main dashboard.

Requirement 4

For requirement 4, having a user receive an alert if a professor changes an assignment within a day, the investigators felt there was an implicit failure of the requirement in the prototype. For P03, the participant felt that the absence of a clear sync indicator makes it unclear if the data will sync automatically or if they had to take action. While testing P04, the investigator felt that the participant noticed the sync status clearly changed, but it was unclear if assignments updated their information or not.

Requirement 6

For requirement 6, the ability of students to see other student's free time without prompting them, each of the investigators felt that there was a partial failure. All of the investigators noted a difficulty with their participant fully understanding the color coding of the calendar. Additionally, it was noted that P04 had a hard time navigating the interface. These struggles to understand that the shading corresponds to free-time, and a struggle to get to that menu indicate requirement failure.

10 FINAL DESIGN AND FUNCTIONAL HIGH-FIDELITY PROTOTYPE

10.1 Description of Design Changes Based on Evaluation

10.1.1 Changes from Qualitative Evaluation. The qualitative evaluation (heuristic evaluation and simplified user testing) identified critical usability issues related to the visibility of system status and interaction discoverability. Users had difficulty interpreting icon-only navigation, understanding system feedback (e.g., sync status and scheduling outcomes), and interpreting the availability heatmap. To address these issues, the following changes were implemented:

Navigation clarity: Added text labels to navigation icon buttons to solve the heuristic issue of unclear icon-only navigation that multiple evaluators identified in Section 8.2. (Shown in Figure 18.)

System status visibility: Substituted conflict alerts with an interactive notification button in the topbar title with an inline unread badge. When users click on it, it shows detailed messages that require attention and are actionable. This resolves the issue of conflict alerts lacking actionable resolution guidance that evaluators identified in Section 8.9. (Shown in Figure 18.)

Input guidance: In the Syllabi tab, added a tooltip to the Course column (with the message "Detected from file name, falls back to General if no match found) and refined syllabus upload hint text ("Upload a PDF or DOC syllabus. Review extracted tasks before adding them to your dashboard. Nothing is saved until you confirm.") This alleviates the issue of a lack of documentation as noted in section 8.10 by guiding the user where further explanation is needed. (Shown in Figure 20.)

Heatmap interpretability: Added sorting and filtering features to the Tasks dashboard so that the flexibility of task views is improved. Users can sort by course, due date, and task name; filter by course and urgency level. This resolves the heuristic problem in section 8.7. (Shown in Figure 18.) Moreover, we increased availability heatmap granularity from 1-hour to 30-minute intervals and added an interactive member legend to clarify availability encoding, resolving heuristic issue 21. (Shown in Figure 23.)

These changes address violations of visibility of system status, lack of help documentation, limited findability, and limited user control and freedom that are found both in the heuristic evaluation and simplified user testing.

10.1.2 Changes from Quantitative Evaluation. The quantitative evaluation of the functional prototype identified usability issues in task entry workflows, visual clarity, and interaction feedback. These changes improve efficiency of use, error prevention, and user control and freedom.

System feedback: Added a confirmation screen after syllabus upload indicating the number of tasks added and next-step options. As shown in Figure 21.

Feature simplification: Removed "Snooze" buttons in task editor, as shown in 19b, to eliminate redundancy and confusion because they serve the same function as manually postponing the due date.

Task editor card usability improvements: As shown in Figure 19a.

- Replaced task type text input with a dropdown.
- Changed date fields to "Start" (optional) and "Due/End" (required).
- Added required field indicators (asterisks).
- Added placeholder text to all fields.

Interaction efficiency: As shown in Figure 19a.

- Implemented auto-save when clicking outside the task editor.
- Expanded reminder units to include minutes, days, and weeks.

Consistency and editability: Reconciled view-only and editable task card flows by adding an edit entry point in the view–only task card. As shown in Figure 19a.

10.2 Final Design and Functional Prototype

The final design is a functional web-based system supporting real task persistence and LLM-based syllabus parsing. The interface consists of four primary tabs: Tasks, Syllabi, Teams, and Profile.

Tasks Dashboard: The Tasks dashboard presents a Kanban-style board with To Do, Upcoming, and Done columns. Users can create, edit, and manage tasks through a structured task editor.

Task Editor: The task editor supports structured input with clear distinctions between required and optional fields, dropdown selections, and placeholder guidance.

Syllabi Upload and Parsing: The Syllabi tab enables users to upload course syllabi and automatically extract tasks using an LLM. Extracted tasks are displayed in a structured table for review and editing.

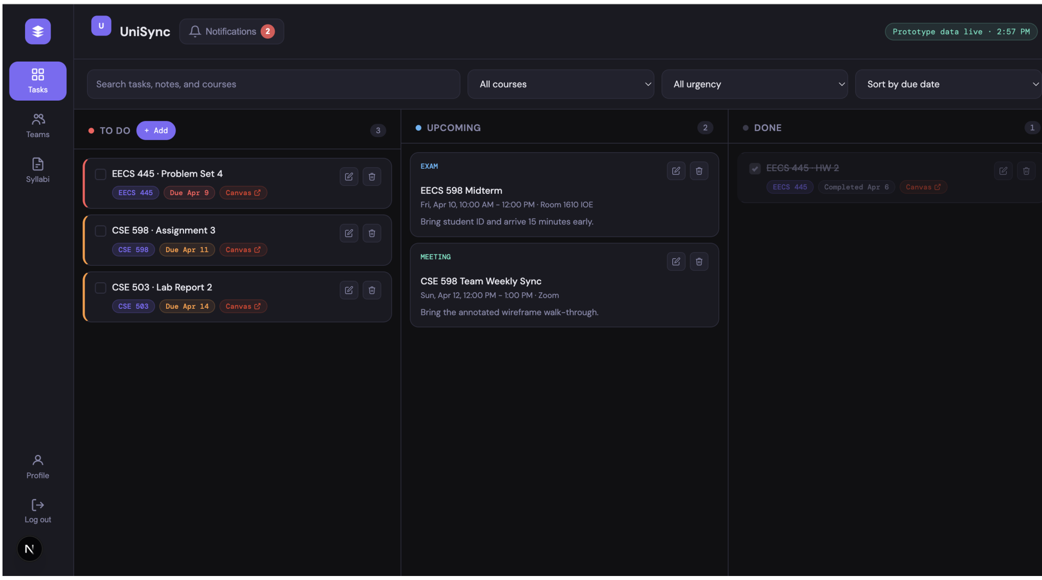
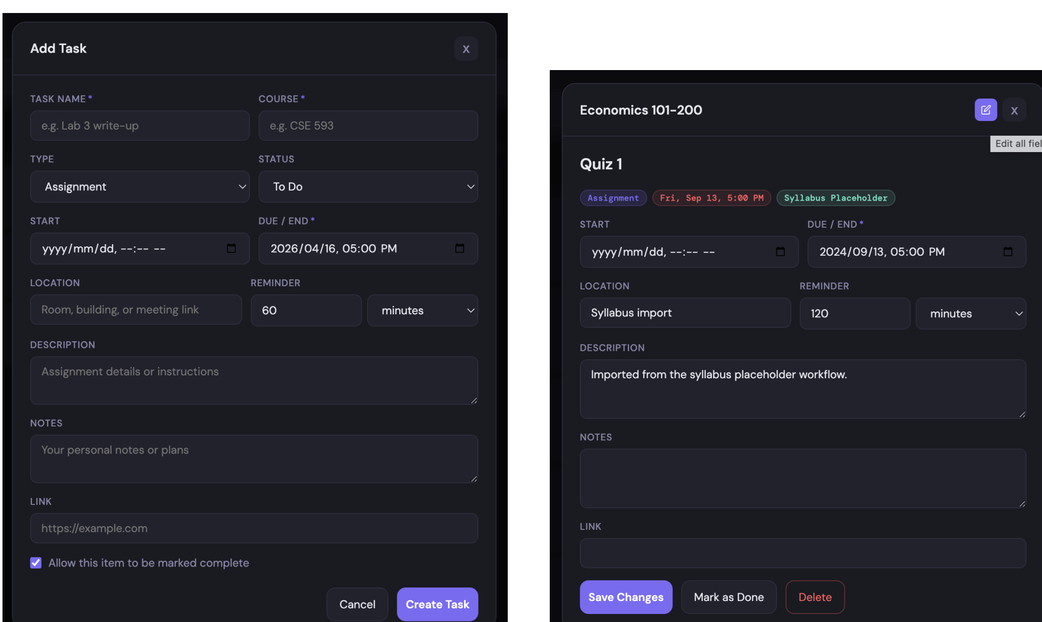


Fig. 18. Tasks dashboard displaying assignments organized by status in a Kanban layout.



(a) Editable task card

(b) View-only task card

Fig. 19. Task editor with structured fields, required indicators, and placeholder text to guide user input.

Confirmation Screen: After task extraction, users receive a confirmation screen summarizing the number of tasks added and navigation options.

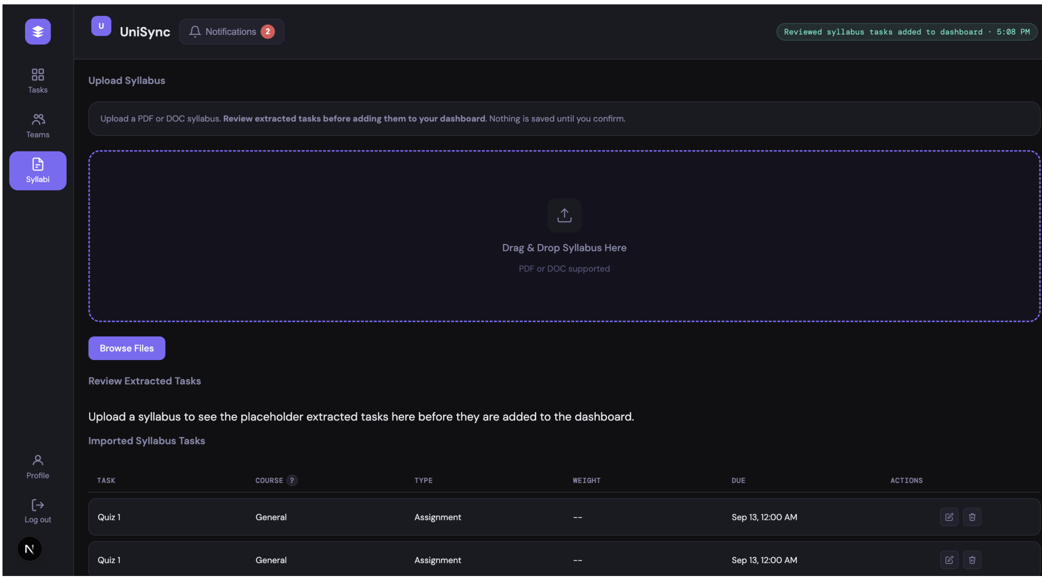


Fig. 20. Syllabi upload interface with drag-and-drop functionality and parsed task output.

Teams Tab and Group Availability Heatmap: The Teams tab provides a 30-minute granularity availability heatmap with an interactive legend to support group scheduling. The heatmap aggregates availability across members without exposing individual calendar details.

10.3 User Interaction with the Final Prototype

Manual Task Entry: Users initiate task creation via a floating action button on the Tasks dashboard. The task editor allows structured input with required fields clearly marked. Tasks are automatically saved when users click outside the editor, provided the required fields are completed. Users can edit tasks through the task card or detailed view.

Syllabus Upload and Extraction: Users upload a syllabus in the Syllabi tab via drag-and-drop. The system parses the document and presents extracted tasks in a structured table. Users can review and modify tasks before selecting "Add Selected to Dashboard." A confirmation screen provides feedback on successful task addition.

Group Meeting Scheduling: Users navigate to the Teams tab to access the availability heatmap. The heatmap displays aggregated availability at 30-minute intervals, supported by an interactive legend. Users select a time slot directly from the heatmap to determine a feasible meeting time. The current prototype does not implement backend-supported meeting scheduling, calendar integration, or AI-recommended meeting times. As such, the interaction flow supports decision-making rather than automatic scheduling execution.

11 FINAL PROTOTYPE EVALUATION

11.1 Purpose

The purpose of this quantitative evaluation was to test whether UniSync's syllabus upload feature reduced the time required for a student to compile a complete list of course assignments into a single location, compared with manual transcription. This directly evaluated Requirement 5.5: that compiling assignments with the system should take less time than manually transcribing them from

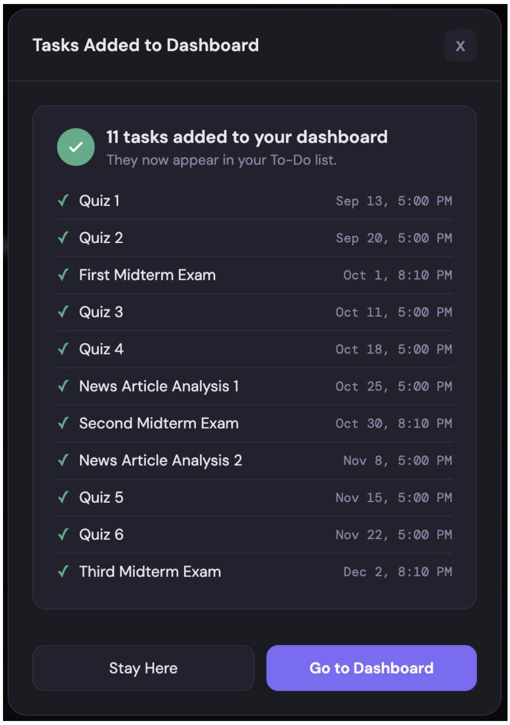


Fig. 21. Confirmation screen displayed after successful syllabus parsing and task addition.

source materials. A secondary goal was to identify usability issues that affected task performance during both workflows.

11.2 Method

We used a within-subjects quantitative study design in which each participant completed the same assignment-compilation task in two conditions: manual entry and syllabus upload. This design was chosen because it allowed each participant to serve as their own control, which reduced variation caused by differences in typing speed, familiarity with academic tools, and general organizational habits. The study tested Requirement 5.5: the time required for a student to compile a complete list of all course assignments into a single location should be less than the time required to do so by manually transcribing them from each source. The independent variable was assignment-entry method, with two levels: manual entry and syllabus upload. The dependent variable was task completion time, measured in seconds, to compile the full list of assignments into UniSync.

The planned statistical analysis for the combined group dataset was a one-tailed paired-samples t-test, since the same participants completed both conditions and the goal was to determine whether syllabus upload was faster than manual entry; if the final timing data did not meet normality assumptions, a Wilcoxon signed-rank test could be used instead. A G*Power analysis using an effect size of 0.5, an alpha of 0.05, and power of 0.80 yielded a target sample size of 27 participants, so each group member recruited approximately 5 to 6 participants. This study design directly addressed Requirement 5.5 by comparing the time required to complete the same task using the prototype versus manual transcription under controlled conditions.

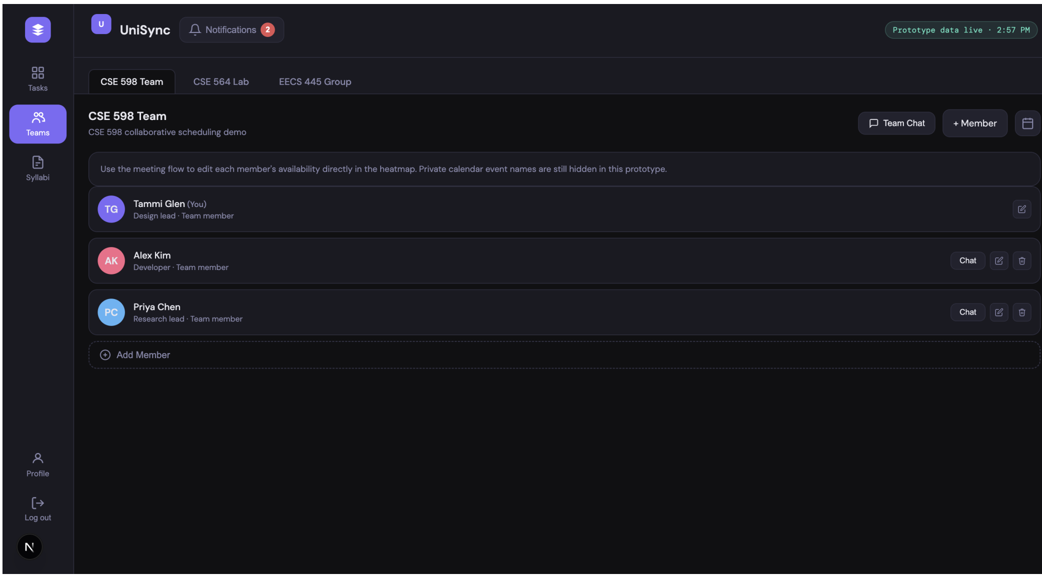


Fig. 22. The Teams tab includes other user accounts as the user Tammi's team members. By clicking on the calendar icon, the user is able to access the group availability heatmap.

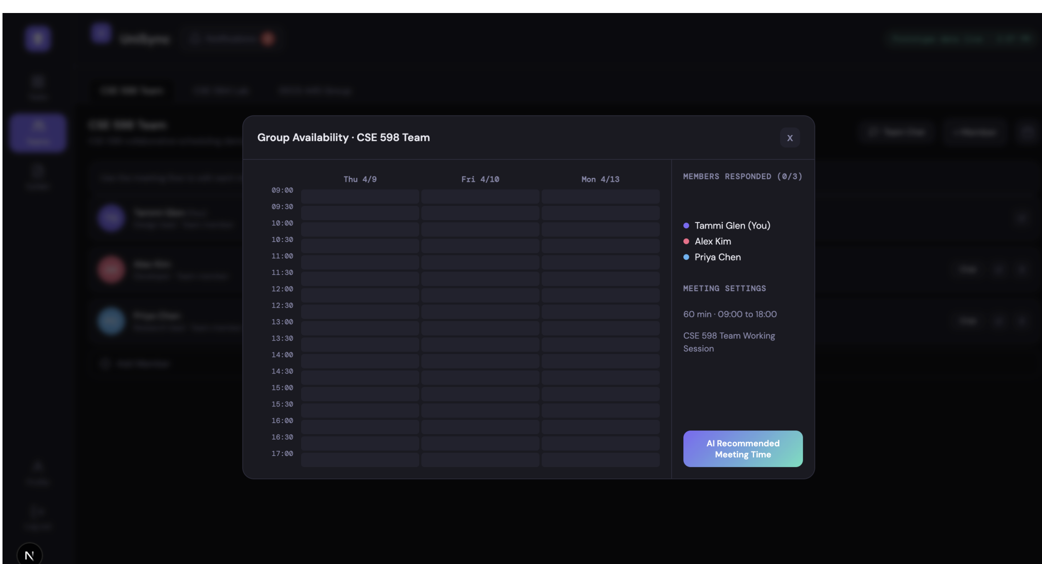


Fig. 23. Group availability heatmap with 30-minute intervals and an interactive legend for member availability.

11.3 Apparatus

The apparatus for this study was UniSync, a high-fidelity functional web prototype designed to help students consolidate course assignments into a single interface. The prototype was implemented with a Next.js/React/TypeScript frontend, a Flask backend, and JSON-based data storage. For this evaluation, the relevant functionality was the assignment aggregation workflow. In the manual-entry condition, participants used the Tasks interface to enter assignments individually into the dashboard. In the syllabus-upload condition, participants used the Syllabi interface to upload a syllabus file, review the extracted assignments, and add them to the dashboard. The syllabus upload feature used Google Gemini to parse uploaded syllabus files and return structured assignment information for user review. Although the larger prototype also included meeting-scheduling features, those were not evaluated in this study. Across the individual reports, UniSync was described as a real functional system rather than a Wizard-of-Oz simulation. Because the current prototype did not yet reliably extract exact assignment times or course-code fields, this evaluation focused primarily on assignment names and due dates.

11.4 Tasks and Procedures

Each participant completed two assignment-compilation tasks using the same ECON 101 syllabus from the University of Michigan syllabus archive: one using manual entry and one using syllabus upload. Before beginning the session, each participant was informed about the purpose of the study, the tasks they would complete, the type of data being recorded, and that participation was voluntary. Participants then provided verbal informed consent before the evaluation began. To reduce time spent locating source material, the evaluator showed the participant where the relevant assignments appeared in the syllabus and demonstrated both workflows before timing began. The syllabus remained open in a separate tab or window during both conditions so that participants could refer to it as needed.

In the manual-entry condition, participants used the task-entry interface to create assignments one by one in the dashboard. In the syllabus-upload condition, participants uploaded the syllabus, reviewed the extracted tasks, and added them to the dashboard once they believed the results were correct. Condition order was tracked for each participant; in several individual evaluations, it was randomized or counterbalanced to reduce transfer effects, while in others a fixed order was used within that evaluator's subset. Task completion time for each condition was recorded separately using a stopwatch or phone timer. Evaluators also recorded brief qualitative observations during each session, including hesitation, field confusion, verification behavior, and other usability issues. In several sessions, evaluators also compared the uploaded results against the 11-item reference assignment list to check extraction accuracy.

No questionnaire was administered across the full study. Most evaluators collected post-task feedback through brief verbal follow-up questions about confusion, ease of use, and preference between the two methods, and one individual evaluation additionally used a short 1–5 ease rating. To support data quality, all evaluators used the same syllabus across sessions, demonstrated both workflows before timing began, and timed each condition separately. In some sessions, the application state was reset between conditions or between participants.

11.5 Participants

The final group sample consisted of 27 participants drawn from the stakeholder population most relevant to the project: current undergraduate and graduate students who regularly manage coursework using syllabi and digital academic tools. Participants were recruited primarily through convenience sampling and personal outreach across the five individual studies. Inclusion criteria generally required participants to be current college or graduate students, typically 18 years of age or older, with experience using course materials or academic management tools such as syllabi, Canvas, and Google Calendar. Exclusion criteria included members of the research team and, in some cases, individuals who had already memorized the ECON 101 syllabus used in the study.

The participant pool included undergraduate, master's, and PhD students from a range of majors, including Computer Science, Information, ECE, Data Science, Civil Engineering, Environmental

Engineering, Mechanical Engineering, Aerospace Engineering, Physics, Computer Engineering, Sports Management, and Economics/Voice. Some individual reports provided exact ages, while others reported age ranges or degree levels only; where exact ages were reported, they ranged from 21 to 30. Overall, the sample was weighted toward STEM and graduate students, which may limit the generalizability of the findings to the broader student population.

11.6 Participant Table

11.7 Results

After conducting the quantitative evaluation, the data from the 27 participants was collected and analyzed with a paired t-test. We report that participants were faster ($t(26) = 13.43$, $p = 3.333e-13$, Cohen's $d = 2.58$) with the syllabus upload tool (mean = 79 seconds) than the manual entry (mean = 371 seconds). The speed difference between the baseline and intervention is statistically significant with our $\alpha = 0.05$. This meets our user requirement 5.5 for the entry speed of the upload tool.

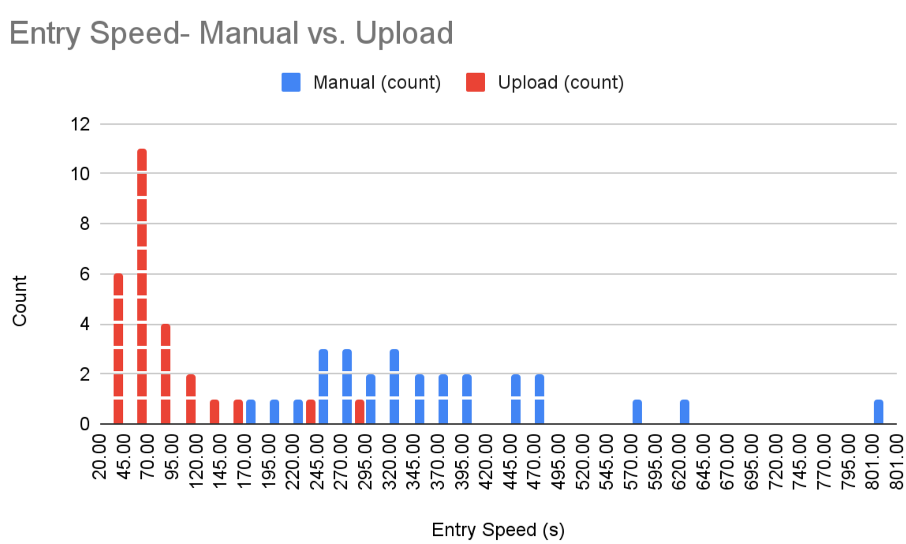


Fig. 24. **Entry Speed Histogram:** Entry Speed of the manual (baseline) vs. the syllabus upload (intervention) in seconds.

We assumed the data to be normally distributed, and found using a histogram to plot the entry times, that the data does appear to resemble a normal distribution for the baseline, and also resembles one with the intervention. The manual entry has a much wider distribution, whereas the upload has a tighter one. There are some significant outliers for the baseline and intervention of users that are much slower than the mean. For some users, verifying the syllabus upload did take as long as some other user's manually entering the information. However, those users were also much slower at the manual entry as well.

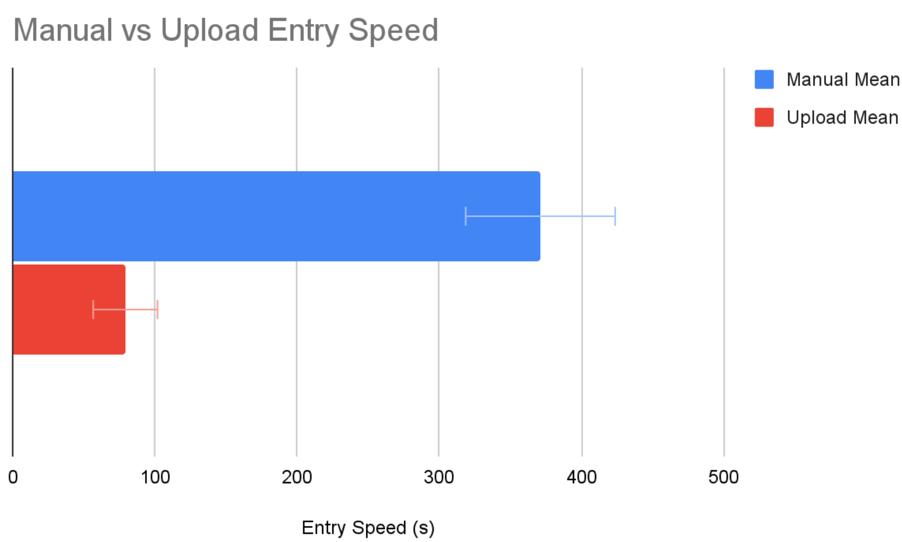


Fig. 25. **Entry Speed Bar Chart:** Manual entry speed vs the syllabus upload entry speed. Error bars for 95% confidence interval are included for each mean value.

The manual entry speed with a mean of 371s, had a much larger 95% confidence interval +/-52.6s compared to the syllabus upload with a mean of 79s and a 95% confidence interval of +/-22.6s. This matches the large effect size we see between the two outcomes with 2.58. The mean of the upload is 4.7x faster than the manual entry. While this is a large jump in the entry time, there was still a significant amount of time required to verify the results from the syllabus upload. However, we believe that this time to verify the upload has much less cognitive load than the manual entry. Further quantitative testing should be conducted to confirm this.

Our data has a very small p value of 3.33e-13 and an abnormally large effect size of 2.58. The graph does show how far apart the distributions are though, so this number does make sense. The statistical test and differences between the means clearly indicate that when functioning correctly, the syllabus upload is much faster. The AI syllabus upload was able to parse all of the assignments on our Econ 101 test syllabus correctly, so the entry speed being significantly faster shows value of the upload when working correctly. Our study of the context of use informed us that manual entry was the main process being used. A speed-up of such magnitude could potentially lower the tedium of scheduling enough to encourage more students to be proactive and plan ahead.

11.7.1 Limitations and Future Directions. The evaluation only tested participants with one syllabus, an Econ 101 syllabus from the University of Michigan in Fall 2024. This syllabus may not be representative of how the upload system would work on all syllabi, and more extensive testing may lead to issues. Our work did not consider error rates on the syllabi, compared to manual entry. It would be interesting to see how many errors the LLM makes on the syllabus parsing compared to a human manually entered input. We had the users verify the LLM for our tests, since that is what a user would likely do in practice.

We feel that the future of this project should concern how much of a speed up is meaningful for our target users. While the difference is statistically significant, whether it definitively makes a meaningful difference on the entire upload experience requires further qualitative testing. Another direction to explore is the error rate and documents that do not do well with our syllabus prompt. While these were beyond the scope of our time limited project, they would be key to a real product implementing the syllabus upload feature.

12 DISCUSSION

The project began with an initial survey of college students, which first revealed the gap between how students currently use generative AI and how they wish they could use it. While majority of respondents used AI primarily within academic contexts for tasks like searching and summarizing, automation emerged as most desired, with students citing setup time and inconsistent performance as the primary barriers to deeper tool adoption. Critically, respondents expressed only moderate trust in AI outputs and reported habitual verification behavior, establishing early on that any AI-assisted academic tool would need to earn user trust through transparency rather than assume it through capability. These findings focused the project's direction: the problem was not that students lacked awareness of automation tools, but that existing tools failed to make automation trustworthy and low-effort enough to justify adoption.

Contextual inquiry with five participants then gave concrete behavioral shape to what the survey had identified statistically. The consolidated flow diagram documented students navigating an average of multiple platforms Canvas, email, syllabi, personal calendars, and group messaging apps, with the burden of integrating these fragmented sources falling entirely on the individual. The consolidated sequence models documented multiple instances of due date transcription errors or outdated information across participants. Most directly relevant to the present study, participants spending significant time manually copying syllabus dates and identified this as the primary reason they gave up on third party organization tools. These observations directly grounded User Requirement 5.5 that automated compilation must be demonstrably faster than manual transcription and the affinity diagram confirmed that participants would only accept automation under the explicit condition that it reduce their effort.

The individual sketches and design critique sessions that followed translated these findings into competing design approaches. Logan's design prioritized rapid multi-source sync but was critiqued for lacking a verification stage; Jacob's AI-powered PDF parsing was praised for addressing courses without Canvas pages but raised concerns about hallucination and accuracy; Chloe's heatmap concept was well-received for group coordination but flagged for privacy risks; Shawn's dashboard was liked for its urgency-based prioritization but criticized for potential information overload; Ali's Kanban layout was praised for intuitive task status tracking. The consolidated UniSync design synthesized these contributions into a three-tier data import architecture API sync, drag-and-drop syllabus upload, and manual entry pairing Jacob's parsing concept with the verification step, directly reflecting the survey finding that students habitually verify AI outputs before acting on them. The primary persona Tammi Glen, whose pain points mapped directly onto the contextual inquiry breakdowns, anchored every subsequent design decision.

The low-fidelity paper prototype and Wizard of Oz testing exposed the first concrete gap between design intent and user experience. While the automation concept resonated with all five user testing participants, the prototype's execution failed across all three evaluated requirements. Every participant struggled to locate the sync button. The absence of a visible sync indicator left participants uncertain whether deadline changes would be surfaced proactively. All five participants reported difficulty interpreting the group availability heatmap. The heuristic evaluation reinforced these failures with greater precision, identifying five usability catastrophes including the absence of an undo function, no help documentation, and the AI priority sort silently reordering tasks without explanation. Therefore UniSync needed to evolve from "invisible automation" to "transparent assistance," and the three-tier syllabus review workflow introduced in the high-fidelity prototype was the direct architectural response.

The present quantitative evaluation was designed to answer the question that all prior work had been building toward: did adding transparency to the upload workflow preserve the efficiency gain that motivated the feature in the first place. Across 27 participants, AI-assisted syllabus upload (79.22 s) was significantly faster than manual entry (371.22 s), with a paired t-test result of p of 3.333e-13, and a Cohen's d of 2.58. This effect size is exceptional by conventional standards, reflecting a 4.7× speed reduction and a 78.7% decrease in task completion time. This directly validates User Requirement 5.5 the data confirms the requirement is met with a margin large enough to account for real-world variability. Projected across a typical course load, the timing results suggest that the semester-level burden participants described in contextual inquiry can be reduced to under ten minutes, concretizing the efficiency promise that motivated the feature from the initial survey findings onward.

The heuristic evaluation flagged the tension between automation and user control as a "usability catastrophe" in the earlier prototype. The current design partially resolved this through the review interface, and the quantitative results validate that this resolution was effective: even with verification time included, the upload condition remained dramatically faster than manual entry. This supports the principle that Human-AI interaction must carefully balance automation with user agency; here, that balance was struck well enough to deliver on the core efficiency promise without removing the meaningful oversight that both the survey and user testing identified as non-negotiable.

The evaluation used a single standardized syllabus the ECON 101 document from the University of Michigan Fall 2024 archive which the AI parsed without extraction errors, representing a best-case scenario that does not reflect real syllabus variability across disciplines and instructors, a concern that is especially prominent given contextual inquiry's finding that some professors do not use Canvas at all and rely entirely on unstructured PDF documents. The participant pool was weighted toward STEM and graduate students, mirroring the demographic limitations of the initial survey. Error rate comparisons between AI extraction and manual transcription were outside scope, leaving open whether time savings persist when corrections are needed.

13 CONCLUSION

This paper presented UniSync, a centralized academic scheduling platform designed to address the information fragmentation and manual overhead that characterize students' current academic workflows, with the evaluation focused on testing whether AI-assisted syllabus upload could demonstrably outperform manual transcription in task completion time. The results demonstrate a 78.7% reduction in task completion time and validating the core efficiency requirement that motivated the feature from the outset. These findings are the empirical culmination of a multi-phase process that began with survey and contextual inquiry work identifying automation as the most desired but underserved capability among students, progressed through prototyping and heuristic and user testing that exposed critical transparency failures requiring UniSync to evolve from invisible automation to transparent assistance, and now confirms that the high-fidelity prototype built against those findings delivered. The persistence of verification behavior in the upload condition indicates the review interface remains an optimization target, and the remaining user requirements are untested quantitatively, meaning future work must extend this protocol to diverse syllabus formats, incorporate error rate as a co-dependent variable, and broaden recruitment beyond the current STEM-weighted sample. Taken together, the evidence marks a meaningful transition UniSync has moved from a design that promised to reduce the academic planning burden to one that has begun to demonstrate it empirically.

REFERENCES

[1] Silvia T. Acuña, Marta Castro, Natalia Juristo, and Ana M. Moreno. 2020. Impact of Usability Mechanisms: An Experiment on Efficiency, Effectiveness and User Satisfaction. *Information and Software Technology* 117 (2020), 106195. https://doi.org/10.1016/j.infsof.2019.106195

[2] Victoria Bellotti, Brinda Dalal, Nathaniel Good, Peter Flynn, Daniel G. Bobrow, and Nicolas Ducheneaut. 2004. What a To-Do: Studies of Task Management Towards the Design of a Personal Task List Manager. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI '04)*. ACM, New York, NY, USA, 735–742. https://doi.org/10.1145/985692.985785

[3] Michael Chang et al. 2024. Development and Validation of a Novel AI Framework Using NLP with LLM Integration for Relevant Clinical Data Extraction Through Automated Chart Review. *Scientific Reports* 14, 1 (2024), 1–10. https://doi.org/10.1038/s41598-024-77535-y

[4] Yuxuan Chen, Sarah Eder, and Kavitha Subramanian. 2021. Usability Evaluation of Canvas: A Learning Management System Study. In *Proceedings of the Human Factors and Ergonomics Society Annual Meeting*, Vol. 65. SAGE Publications, 1234–1238.

[5] John Dunn, Malia Walker, Samantha Dagdelen, et al. 2024. Extracting Accurate Materials Data from Research Papers with Conversational Language Models and Prompt Engineering. *Nature Communications* 15, 1 (2024), 1–12. https://doi.org/10.1038/s41467-024-45914-8

[6] Kasper Hornbæk. 2013. Some Whys and Hows of Experiments in Human-Computer Interaction. *Foundations and Trends in Human-Computer Interaction* 5, 4 (2013), 299–373. https://doi.org/10.1561/1100000043

[7] Hyunjin Kim, Jiwon Lee, and Minsoo Park. 2025. Heterogeneous Patterns of Academic Delay: A Behavioral Analysis of Undergraduate Students. *Journal of Educational Psychology* 117, 2 (2025), 215–230.

[8] Jonathan Lazar, Jinjuan Heidi Feng, and Harry Hochheiser. 2017. *Research Methods in Human-Computer Interaction* (2nd ed.). Morgan Kaufmann, Cambridge, MA.

[9] Brian L. Westra, Gail E. Latimer, Susan A. Matney, Jung In Park, Joyce Sensmeier, Roy L. Simpson, Marilyn J. Swanson, Judith J. Warren, and Connie W. Delaney. 2013. Comparison of Manual Versus Automated Data Collection Method for an Evidence-Based Nursing Practice Study. *CIN: Computers, Informatics, Nursing* 31, 4 (2013), 159–165. https://doi.org/10.1097/NXN.0b013e318289ed4b

[10] Steve Whittaker, Victoria Hollis, and Andrew Gunn. 2025. Fragmented Information Ecosystems: The Cognitive Cost of Distributed Task Management. In *Proceedings of the 2025 CHI Conference on Human Factors in Computing Systems (CHI '25)*. ACM, New York, NY, USA, 1–14.

[11] Michael Williams, Rachel Thompson, and James Davis. 2023. Cross-Context Task Management: Requirements for Unified Personal Information Systems. *ACM Transactions on Computer-Human Interaction* 30, 4 (2023), 1–28. https://doi.org/10.1145/3579321

[12] Qian Yang, Aaron Steinfeld, Carolyn Rosé, and John Zimmerman. 2020. Re-Examining Whether, Why, and How Human-AI Interaction Is Uniquely Difficult to Design. In *Proceedings of the 2020 CHI Conference on Human Factors in Computing Systems (CHI '20)*. ACM, New York, NY, USA, 1–13. https://doi.org/10.1145/3313831.3376301