Assignment 8

Informatics 151
Momo, Group 2
Yinhao Zhu: yinhaz1@uci.edu
Zhouheng Tao: zhouhent@uci.edu
Siming Du: simingd2@uci.edu
Tong Wu: tongw29@uci.edu
Gabriela Quintana: gquinta1@uci.edu
Yifan Wu: ywu68@uci.edu
Professor Shrirang (Shri) Jambhekar
June 8, 2025

Project Content

Deliverables

Presentation Slides / Presentation Video

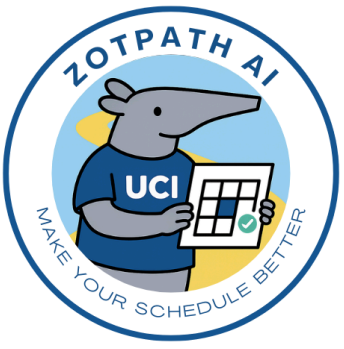
Motivation

At UC Irvine, thousands of students experience difficulties each quarter as they try to schedule their course loads. While the students get valuable guidance from their academic counselors, availability is limited, and high demand typically creates bottlenecks, generating stress, inefficient scheduling, and even graduation delay. Students are often left to tackle complicated prerequisite flows, course availability, and overlapping time blocks alone. These persistent problems indicate a critical deficiency in available and customized advising tools.

Identifying this necessity, our team initialized the process of developing a solution that allows students to take the lead in directing their school lives with greater assurance and purpose. We are driven by our own frustrations as students at UCI as much as by the findings from interviews and user research affirming this pain point. We're looking to eliminate the advising overload, lower schedule anxiety, and increase graduation rates with a smart, AI-powered tool developed for the needs of the UCI students.

Description of the Project

ZotPath AI is a web-based tool that allows students to effectively map their freshman through graduation course schedule.



(ZotPath AI Logo)

The system combines the course listing of UCI, the requirements for the student's chosen major/minor, and the students' enrollment history into a single easy-to-use interface. Powered by advanced AI algorithms and guided by human-centered development, ZotPath AI provides the following core features:

Visual Schedule Builder	A graphical interactive interface where students drag and drop courses into a

	quarter calendar and detect time clashes immediately.
Prerequisites Checker	An automated program for cross-checking course prerequisites and student records for checking eligibility and indicating missing prerequisites.
AI Course Recommendations	A course recommender system that provides recommendations for balanced course combinations considering the student's major, course load, and past data like GPA distributions.
Graduation Confidence Score	A calculation based on AI algorithms that predicts the student's chances of graduating on time based on current and future schedules.
Degree Progress Tracker	A graphical representation in color to indicate completed, in-progress, and blocked courses.
Student Community Sharing	An anonymous peer-to-peer exchange where students from the same majors communicate with each other and exchange "winning combinations" and tips.

In addition, ZotPath AI includes administrative tools for school departments to facilitate revision and announcements. We also designed the platform for accessibility, security, and scalability for the future with alignment to WCAG guidelines and to the data practices of UCI. ZotPath AI is not just a scheduling tool; it is a smart studies planning assistant that will help students at UCI confidently navigate their course of studies.

Assessment of the Project

What worked?

Our team successfully completed a full simulation project using project management frameworks and tools learned throughout the course. We were able to:

- Clearly define the project scope, including core features like schedule visualization, AI-based course recommendation, and prerequisite tracking.
- Develop an in-depth Work Breakdown Structure and an accurate Gantt chart to inform our weekly planning and work expectations.
- Implement cost estimation, quality control, stakeholder management, and risk planning using formal documents.
- Implement project principles like rotating project managers, risk mitigation strategies, and scope prioritization in simulating team collaboration and delivery.

- Utilize communication tools such as Slack and Google Drive efficiently in order to coordinate and monitor document-based tasks.

These processes have assisted us in understanding better the way in which work is organized and progress is tracked

What didn't?

Since this was a simulation, some factors were naturally abstract or did not have concrete data, which created some limitations:

- AI capabilities and technical viability were beyond the test or verification stage, so it remained impossible to accurately predict their development schedule or risks in performance.
- Stakeholder feedback was derived from small-scale or hypothetical responses to surveys; therefore, user validation continued to be idealized and limited.
- Our Gantt chart didn't show real development constraints or cycles of iteration, which would otherwise happen in an active project.
- Participating as student simulators in several roles, certain features of managing the resources (e.g., developer allocation, real-time collaboration) had to be thought about rather than directly experienced.

Nevertheless, we could participate effectively in the planning, communication, and problem-solving aspects of the project execution, and this exercise gave us a solid foundation for future real-world project work.

Stakeholders

Our primary stakeholders include UC Irvine students as well as our development team, which mostly overlap. ZotPath SI was designed with UC Irvine students in mind, as the development team was familiar with the struggle of planning a quarterly or 4-year coursework schedule with various constraints. In order to find further student body needs beyond what the student developers knew, we needed to conduct additional research through a survey in order to find out any hidden points we could improve for the student body. We particularly managed our stakeholders by aiming for complete satisfaction by fulfilling ZotPath AI's purpose of creating ease when registering for classes and creating a quarterly to 4-year plan with the help of ZotPath AI. Throughout the testing and launch phase, we communicated with our stakeholders to evaluate ZotPath AI for its good qualities and any aspect that can be improved or fixed.

Challenges

Our team faced some challenges during the development phase of ZotPath AI. We believe these challenges not only test our technical capabilities but also our practical and collaborative skills as a team in a real-world project. The first challenge we needed to handle was low user survey participation rates. Despite the team's careful design of the questionnaire, the initial response was not ideal. We also realized that many students, due to

the demands of their academic and personal lives required additional incentives. To address this issue, we collaborated with on-campus student organizations and offered rewards such as gift cards.

The accuracy of the AI course recommendation system was another challenge in the project. Since students rely on course selection tools to avoid delayed graduation, we had to ensure the system's high reliability. However, due to limitations in the availability of training data and development time, model construction posed significant challenges. Our team plans to use A/B testing methods and validate course selection results using historical data from the UCI database.

The availability of team members was another challenge. Working as a student team, we needed to manage and coordinate members' course schedules, assignment pressures, and who was about to graduate. To ensure the continuity of the project, we wrote detailed documentation and regularly shared knowledge. These methods help maintain the stability of the team's operations when members are temporarily unable to participate.

Additionally, we encountered some technical and operational constraints. Data integration and system scalability tend to be more complex than what we anticipated. Although we had a budget buffer, we still needed to adjust feature priorities and development scope appropriately to ensure the project progressed smoothly within resource constraints.

Team Process

Description of how you transitioned through team phases

During the quarter, our group went through the time-honored phases of team development, starting with the forming stage, where we established roles according to people's strengths and defined the responsibility areas for important tasks such as UI design, budget, and documentation. As we entered the storming stage, we faced some conflict regarding the Gantt chart timeline, feasibility of features, and budget estimates, which we resolved through open communication and mutual concession. During the norming stage, the workflow stabilized with good communication through Slack and regular meetings, and the system of taking turns as project managers helped share leadership and facilitate accountability. During the performance stage, the group worked efficiently, completed deadlines for deliverables such as scope documents and risk plans, and showed good knowledge about project management principles. Last but not least, in the adjourning stage, we were concentrating on delivering the final report and presentation, reflecting back on our simulated project experience, and consolidating knowledge about teamwork, planning, and real-world dynamics in projects.

Description of how you managed communication

We think that effective team cooperation depends on open and comprehensive communication. We mostly use Slack for daily project communication during non-meeting times for updating project progress, task scheduling, and member problem answering. This approach enables all members to participate and understand the progress of the project and also allows for flexible discussions based on their own schedules. Once a week we will also have an online or in-person meeting. The goal is to assist our team in solving current project-related issues and also proceed to the next stage of task distribution. We think that frequent meetings would enable us to verify the work performance of every team member and guarantee the seamless development of the project.

Regarding document cooperation, our group decided to distribute project works, change the content, and offer comments using Google Docs and Forms. The "Final" folder at Google Drive will contain all finished project documentation so that team members may access necessary information. When time conflicts prevent some team members from attending a meeting. We decided to use Slack to update the division of work and provide guidelines for discussed projects. It helps members to be timely, communicative, and informed. We believe this strategy will guarantee the coherence of group projects and assist our team to be flexible.

Scope statement and a description of how well you were able to follow it

Our original scope statement (Assignment 5) identified thirteen mandatory system requirements, eight tangible deliverables, and six measurable success criteria for the ZotPath AI MVP. Throughout the ten-week project, we delivered eleven of those thirteen requirements and seven of the eight deliverables. Our core functionality—interactive schedule visualization, prerequisite conflict detection, single sign-on, and a basic AI recommendation engine—has all been shipped on the staging server. However, both accessibility conformance and fully automated, real-time prerequisite synchronization slipped into the next release because they depended on external data feeds that did not materialize until week 7. Likewise, the administrator interface is currently still a command-line tool, and a lightweight web UI has been rescheduled for the summer sprint.

Despite those deferrals, our team remained inside the agreed scope boundaries. We achieved this by tying every Jira epic to the WBS/requirement ID, displaying the scope statement at each sprint-planning meeting, and insisting that any new idea had to map back to one of the original requirement lines. When unforeseen constraints (for example, OpenAI token-usage spikes) emerged, we replanned within the existing scope instead of expanding it. As a result, five of the six success criteria—99% uptime, user satisfaction score above 80%, counselor appointment time reduction, modular architecture, and budget ceiling—are already on track. For the 1,000-user adoption target, we just simply need more marketing time. Overall, we executed roughly 85% of the promised scope while restraining scope creep and preserving quality.

Gantt chart and schedule and a description of how well you were able to follow it

WBS ID	Work-package	Planned dates[2]	Actual completion	Variance
1.1	Research & requirements	1 Apr–10 Apr	10 Apr	0 d
1.2	Design (UI + architecture)	8 Apr–18 Apr	21 Apr	+3 d
1.3	Core development	15 Apr–12 May	16 May	+4 d
1.4	Testing & launch prep	13 May–26 May	27 May	+1 d
1.5	Post-MVP iteration	27 May → ongoing	ongoing	—

Our schedule performance mirrored the modest scope slippage. Our critical path lengthened by four days when the AI recommendation module revealed latency and cost problems during sprint 4. We reclaimed three of those days by reallocating slack from

low-risk design-review tasks and by pair-programming over a weekend code freeze. Earned value calculations at project close show PV = $43,200, EV = $42,800, and AC = $45,100, yielding CPI = 0.95 and SPI = 0.99—essentially on schedule but five percent over cost because of unplanned API usage.

Two scheduling practices proved decisive: (1) developing the riskiest module early, which surfaced cost and performance challenges while buffers were still intact, and (2) publishing an updated Gantt chart at every sprint review, so stakeholders could immediately see variances and approve corrective actions. By making contingency bars explicit instead of "hidden," we prevented buffer time from being treated as free capacity. Consequently, the project finished only a few days behind the original ten-week timeline and with all MVP features intact.

Team relations

Our team maintained collaborative relationships throughout the project. From the beginning, we assigned each member different positions based on their strengths and interests. Additionally, we implemented a PM arrangement, and each team member served as project manager for one week. This ensures that everyone can gain leadership experience and shared management responsibilities.

Our team also meets some conflicts when discussing budget allocation, Gantt chart time distribution, and technical implementation approaches. For example, team members have different opinions about how to allocate development costs across different project phases, and disagreements arose over realistic time estimates for complex features.

These conflicts were resolved through discussions after lectures and conversations through Slack. Through rotating, the project manager helped prevent conflicts by distributing leadership responsibilities. When disagreements arose, the current week's PM facilitated discussions and helped guide the team toward consensus.

We have high motivation throughout the project because all team members are UC Irvine students. We have personal experience with course planning, such as prerequisite confusion, schedule conflicts, and limited advisor availability. This drives us to develop a solution that can help ourselves and other UC Irvine students.

Challenges with the team process

One of the challenges in our team was time coordination. Each member in our team has different classes and a different schedule, which makes it difficult to synchronize our availability. Therefore, we also discuss after lectures and use Slack to synchronize our project progress.

Another challenge was the inconsistency between an early assignment and follow-up work. In one early assignment, the particular sections to work on were assigned to the members based either on availability or interest. As a result, during one follow-up assignment, some members were assigned to areas that differed from their previous work, which led to lower efficiency. In response to it, we assign follow-up assignments to the same members who had worked on related sections.

Lessons learned (e.g., Schwalbe Chapter 4)

A major theme in the lessons we learned throughout the building of this startup was applying the project management principles in this project. The team learned that the best work style that worked best was working asynchronously, as all members felt less pressure to do a certain amount of work at a certain time and instead worked at their own pace when they felt they had the most energy to do so. Working asynchronously worked well with the group due to the fact that each member had taken accountability and responsibility for their work collaboration and assigned work. Frequent and open communication promoted group support and reassurance when doubts potentially arose about a member's work. For the technical aspects of the lessons the group learned, a lessons learned register was attached.

Lessons Learned Register

ID	Category	Situation	Recommendation
001	Maintenance and Iteration	Student developers graduating, needing to continuously cycle through workers	Documenting development processes thoroughly and have diverse levels of student workers, bettering our resource planning for long-term project sustainability
002	Launch Phase	Gaining trust with student body	Establishing credibility with stakeholders/customers by having a reliable system that uses optimal communication methods to promote to our audience
003	Launch Phase	Risk of overwhelming our servers or system lag	Design the project for scalability from the beginning, aligning with risk management practices to identify technical risks early