PRD - TeamMatch

(by SkillSwap Team)

Assumptions

TeamMatch operates under several core assumptions that define its design boundaries. Students provide reasonably accurate self-assessments of their technical skills and availability. Instructors define valid team size constraints and project requirements. Courses using TeamMatch are structured around collaborative, project-based learning where team composition meaningfully impacts outcomes. Student skill descriptions can be normalized into structured technical categories with acceptable accuracy. Deterministic outputs (same input → same output) are preferred over randomized assignment to ensure fairness and reproducibility. Roster enrollment data is assumed to be accurate and up to date. These assumptions allow the system to remain transparent, explainable, and technically reliable.

Target Audience and Educational Benefit

Primary Target: Computer Science Instructors
TeamMatch is designed for instructors managing project-based courses who must form teams in a way that is fair, structured, and educationally effective. Instructors often face limited time, mixed-skill classrooms, and pressure to maintain equity across groups. Research on structured and heterogeneous team composition shows that intentionally balanced teams produce stronger engagement and improved performance compared to randomly formed groups. TeamMatch translates those research principles into a practical system. For instructors, TeamMatch provides a research-aligned framework for forming balanced teams, reduced manual effort in grouping students, transparent and explainable team formation logic, and confidence that team assignments support learning objectives.

Secondary Target: Undergraduate Students in Project-Based Courses
Students are the direct beneficiaries of improved team structure. Random or socially formed teams frequently lead to workload imbalance and limited exposure to diverse skill sets. By placing students in teams with complementary strengths and aligned availability, TeamMatch promotes equitable contribution, peer mentorship, and stronger collaborative learning experiences. For students, this results in more equitable workload distribution, exposure to diverse technical skills, stronger peer learning opportunities, and improved collaborative problem-solving.

Goals

2.1 Teaching Goals

Ensure evenly distributed skill sets across teams. Promote equitable workload distribution. Encourage positive interdependence among team members. Increase peer learning between students of differing experience levels.

2.2 Operational Goals

Reduce instructor time spent forming teams. Automatically enforce team size and enrollment constraints. Guarantee deterministic outputs. Provide explainable summaries for instructor review.

2.3 Technical Goals

Convert unstructured skill descriptions into structured representations. Implement a rule-based optimization engine that balances skill coverage, experience levels, and availability overlap. Maintain reproducibility, transparency, and auditability.

2.4 Autonomous Team Formation Agent

To move beyond a one-time optimization tool, TeamMatch includes an Autonomous Team Formation Agent that manages the lifecycle of team formation and monitoring. The deterministic optimization engine is responsible for generating balanced teams, while the agent operates at a higher system level. It monitors events such as new student submissions, instructor constraint updates, and project deadlines, determines when optimization should be triggered, and ensures required inputs are complete before execution. After optimization runs, the agent verifies constraint satisfaction, logs execution metadata for reproducibility, and generates structured diagnostic summaries explaining how constraints were satisfied and highlighting trade-offs such as skill variance versus availability overlap. The agent can also support periodic student check-ins during the project lifecycle by evaluating milestone-based updates on task responsibility, contribution balance, and schedule adherence. When imbalance or risk patterns emerge, the agent generates alerts and instructor-facing summaries. The agent does not modify the deterministic optimization logic or automatically reassign teams; instead, it orchestrates, validates, monitors, and contextualizes outputs, transforming TeamMatch into an event-driven, lifecycle-aware system rather than a single-trigger assignment tool.

2.5 User Stories

Instructor User Stories
As an instructor, I want to upload or import a class roster so that I can begin team formation efficiently. As an instructor, I want to define team size and constraint parameters so that the system reflects my course requirements. As an instructor, I want balanced teams generated automatically so I do not manually assign students. As an instructor, I want explainable

summaries of team assignments so I can review and approve them. As an instructor, I want deterministic results so repeated runs with identical inputs produce identical outputs.

Student User Stories
As a student, I want to submit my skills and availability so that I am placed on a compatible team. As a student, I want to be placed on a team with complementary strengths so workload is balanced. As a student, I want team formation to feel structured and fair rather than arbitrary.

High-Level Plan (Implementation Roadmap)

Step 1: Data Collection
Gather student inputs including skills, experience level, availability, and preferred roles. Collect instructor inputs such as project requirements, team size, and constraints.

Step 2: Skill Normalization
Convert unstructured skill descriptions into predefined technical categories. Apply project-specific weighting based on instructor objectives.

Step 3: Deterministic Team Optimization
Apply a rule-based greedy assignment algorithm. Enforce team size and enrollment constraints. Evaluate teams using skill variance minimization, experience balance, and availability overlap maximization.

Step 4: Explanation Generation
Produce structured summaries explaining team formation decisions. Provide instructor-facing metrics to support review and adjustment.

3.1 Out of Scope
To maintain focus and scope discipline, TeamMatch will not replace full LMS functionality such as grading, messaging, or assignment submission. It will not eliminate all interpersonal conflicts within teams or guarantee perfect student satisfaction. The system will not use opaque black-box machine learning models that reduce explainability. It will not provide long-term team performance analytics beyond initial formation in the current version.

Success Criteria

4.1 Quantitative Metrics
All teams satisfy instructor-defined constraints. Skill distribution variance across teams is minimized. Average availability overlap exceeds baseline random grouping. Reproducibility is verified such that identical inputs produce identical outputs.

4.2 Qualitative Indicators
Instructors report reduced effort in team formation. Students perceive team assignments as fair and balanced. Teams demonstrate improved collaboration compared to prior ad-hoc formation methods.

4.3 Research Alignment Validation
The system reflects established collaborative learning principles, including that mixed grouping improves academic interaction, structured team formation enhances engagement and accountability, and deliberate team composition influences learning outcomes.

4.4 Open Questions
Should skill weighting remain static or be adjustable per project? Should instructors be allowed to manually override assignments after generation? How should incomplete or inconsistent student data be handled? Should future versions incorporate probabilistic optimization while preserving explainability? What trade-offs are acceptable between minimizing skill variance and maximizing availability overlap?

Core Features & Research Alignment

Feature	Purpose	Research Support
Structured collection of student skills, roles, and availabilities	Captures predictable inputs for team formation	Mixed skill grouping enhances collaboration and problem solving
Instructor-defined team constraints	Allows customization for specific class contexts	Aligns grouping with educational objectives
Deterministic team formation engine	Produces consistent, reproducible strategies	Balanced roles improve performance consistency

Explainable team assignment reporting	Makes process transparent for accountability	Supports deeper learner engagement with team contexts

---

References

Aranzabal, A., Epelde, E., Artetxe, M., & Jauregi, E. (2022).
Team formation on the basis of Belbin's roles to enhance students' performance in project-based learning. *Education for Chemical Engineers, 38*, 22–37.
https://doi.org/10.1016/j.ece.2021.09.001

Center for Educational Innovation, University of Minnesota. (n.d.).
Forming teams. https://cei.umn.edu/teaching-resources/team-projects/forming-teams

Wang, X., Song, G., & Ghannam, R. (2024).
Enhancing teamwork and collaboration: A systematic review of algorithm-supported pedagogical methods. *Education Sciences, 14*(6), 675.
https://doi.org/10.3390/educsci14060675