

Team Formation App - Requirements Document

Executive Summary

This application assists course staff in large STEM classes by using AI to optimize team formation for project-based learning. The system leverages research-backed principles from Belbin's role theory, team-based learning practices, and cooperative learning to create diverse, complementary teams that maximize learning outcomes and knowledge sharing.

Problem Statement

Current team formation in educational settings suffers from several critical issues:

- Students are assigned to teams without consideration of complementary skills or learning characteristics
- Groups often lack diversity in knowledge domains, leading to knowledge gaps
- Schedule conflicts prevent effective collaboration
- No-show partners disrupt team dynamics
- Instructors lack tools to systematically form effective teams
- Little attention is given to developing actual teamwork skills

Core Objectives

Primary Goals

1. **Optimize Team Formation:** Use AI algorithms to create teams with complementary skills, diverse knowledge bases, and compatible schedules
2. **Augment Course Staff:** Provide tools that reduce administrative burden while improving educational outcomes
3. **Maximize Learning:** Ensure each team has coverage across required knowledge domains and diverse skill sets
4. **Enhance Peer Knowledge Sharing:** Facilitate effective collaboration through strategic team composition

Secondary Goals

1. Reduce team formation time for instructors
2. Minimize team dysfunction due to poor initial composition
3. Provide data-driven insights into team effectiveness

- 4. Scale effectively for large classes (100+ students)

Functional Requirements

1. User Management & Authentication

- **Instructor Registration/Login:** Secure account creation and authentication
- **Student Registration/Login:** Simple account creation, potentially through course enrollment
- **Role-Based Access Control:** Different permissions for instructors vs students
- **Profile Management:** Basic user information and preferences

2. Course & Project Management

- **Course Creation:** Instructors can create and manage multiple courses
- **Student Enrollment:** Add students via CSV upload or manual entry
- **Project Definition:** Create projects with specific requirements and timelines
- **Roster Management:** View and manage enrolled students

3. Questionnaire System

- **Customizable Questionnaires:** Instructors can create tailored questionnaires
- **Question Types:**
 - Multiple choice (Belbin roles)
 - Checkboxes (skills and knowledge areas)
 - Schedule availability grids
 - Likert scales (preferences and experience levels)
 - Text input (specific expertise or concerns)
- **Template System:** Pre-built questionnaires based on research best practices
- **Student Response Portal:** Simple interface for students to complete questionnaires

4. Team Formation Engine (Core Domain)

- **AI-Powered Algorithm:** Intelligent team formation based on multiple criteria:
 - Belbin role distribution
 - Skill complementarity
 - Knowledge domain coverage
 - Schedule compatibility
 - Learning style diversity

- **Constraint System:** Instructors can set parameters:
 - Team size (min/max)
 - Required role distributions
 - Mandatory skill coverage
 - Schedule overlap requirements
- **Formation Preview:** Show algorithm results before finalizing
- **Manual Adjustments:** Drag-and-drop interface to modify generated teams

5. Team Management & Visualization

- **Team Dashboard:** Visual representation of formed teams
- **Team Composition Analysis:** Show role distribution, skill coverage, schedule overlap
- **Team Stats:** Quantitative measures of team balance and diversity
- **Export Capabilities:** Generate team lists for LMS integration

6. Analytics & Reporting

- **Formation Effectiveness Metrics:** Track team performance over time
- **Post-Project Surveys:** Gather feedback on team dynamics and learning outcomes
- **Instructor Reports:** Insights into what makes effective teams in their specific context
- **Data Visualization:** Charts and graphs showing team composition trends

Non-Functional Requirements

Performance

- Support concurrent access by 500+ students during peak questionnaire periods
- Team formation algorithm must complete within 30 seconds for classes up to 200 students
- Web interface must be responsive across desktop and mobile devices

Scalability

- Support multiple courses and instructors simultaneously
- Handle classes ranging from 20 to 500+ students
- Database design must accommodate growing historical data

Security

- Secure authentication and session management

- Data privacy compliance (FERPA for educational records)
- Role-based access control to prevent unauthorized data access
- Secure API endpoints with proper authorization

Usability

- Intuitive interface requiring minimal training for instructors
- Simple, mobile-friendly questionnaire experience for students
- Clear visualization of team compositions and rationale
- Accessibility compliance (WCAG 2.1 AA)

Reliability

- 99.5% uptime during academic semesters
- Automated backups and disaster recovery
- Graceful error handling with user-friendly messages

Technical Constraints

Integration Requirements

- **LMS Integration:** Canvas, Moodle, Blackboard API compatibility for roster sync
- **Export Formats:** CSV, PDF for team lists and reports
- **Email Integration:** Notifications for questionnaire deadlines and team assignments

Data Requirements

- **Student Data:** Personal info, academic background, questionnaire responses
- **Course Data:** Enrollment lists, project requirements, deadlines
- **Team Data:** Formation parameters, generated teams, modification history
- **Analytics Data:** Team performance metrics, survey responses, usage statistics

User Stories

Instructor User Stories

1. **As an instructor**, I want to create a new team formation activity so that I can organize students for an upcoming project
2. **As an instructor**, I want to customize questionnaires so that I can gather relevant information specific to my project requirements

3. **As an instructor**, I want to set formation parameters so that teams meet my pedagogical goals
4. **As an instructor**, I want to preview and adjust generated teams so that I can ensure optimal team composition
5. **As an instructor**, I want to export team lists so that I can update my LMS and communicate with students

Student User Stories

1. **As a student**, I want to complete a questionnaire easily so that I can be placed in an appropriate team
2. **As a student**, I want to view my team assignment so that I can contact my teammates
3. **As a student**, I want to see why I was placed in my team so that I understand the rationale
4. **As a student**, I want to provide feedback on my team experience so that future team formations can be improved

Success Metrics

Primary Metrics

- **Team Formation Time Reduction:** 80% reduction in instructor time spent on team formation
- **Team Satisfaction:** 85%+ student satisfaction with team assignments (vs random assignment)
- **Knowledge Coverage:** 95%+ of teams have adequate coverage of required knowledge domains
- **Participation Rate:** 90%+ questionnaire completion rate

Secondary Metrics

- **System Adoption:** Number of courses and instructors using the system
- **Algorithm Accuracy:** Correlation between predicted team effectiveness and actual outcomes
- **User Retention:** Instructors continuing to use the system across multiple projects

Assumptions & Dependencies

Assumptions

- Instructors are willing to invest time in setting up questionnaires for better team outcomes
- Students will provide honest responses to questionnaires
- Belbin role theory and complementary skill matching will improve team effectiveness
- Class sizes will range from 20-500 students

Dependencies

- Access to course enrollment data (manual upload or LMS integration)
- Student participation in questionnaire completion
- Instructor feedback and iteration on algorithm effectiveness
- Academic calendar integration for project timing

Future Enhancements (Post-MVP)

Advanced Features

- **Machine Learning Evolution:** Algorithm that learns from past team successes/failures
- **Real-time Collaboration Tools:** Integrated project management features
- **Peer Evaluation System:** Ongoing assessment and team adjustment capabilities
- **Multi-Institution Support:** Cross-university collaboration capabilities
- **Mobile App:** Native mobile application for improved student experience

Integration Expansions

- **Calendar Integration:** Automatic schedule conflict detection
- **Communication Platforms:** Slack/Discord team channel creation
- **Version Control:** GitHub classroom integration for coding projects
- **Assessment Tools:** Integration with peer evaluation platforms