

Semester Volunteer Matching System (V1.0) Product Requirements Document

1. Introduction

1.1. Purpose

This Product Requirements Document (PRD) outlines the goals, features, and requirements for the initial release (V1.0) of the **Semester Volunteer Matching System**. This software tool is being developed to automate the process of connecting post-secondary **Learners** (students seeking assistance) with **Learning Peers** (volunteer tutors) for weekly, 1-hour study sessions. The primary purpose is to replace the current inefficient manual process, thereby significantly reducing the administrative matching time and maximizing the total number of students placed into groups.

1.2. Project goals and success metrics

The core goals of this system are to:

- **Reduce administrative matching time:** The automated process must reduce the time required for a full matching run to **under 48 hours** from the current average of 2–3 weeks.
- **Maximize matched students:** The system must maximize the total count of Learners successfully placed into a group each semester.

Success will be measured by two key metrics:

- Average matching time per semester/run: <48 hours.
- Total number of matched Learners per semester (to be tracked against historical manual matching data).

2. Scope and features

The V1.0 scope focuses entirely on the core matching logic, administrative workflow, data integration, and communication requirements necessary to execute and manage the initial student-Peer matching process.

2.1. In-scope features

- Secure API connection to the Google Sheet data source.

- Data ingestion, cleaning, and preparation for matching runs.
- Core matching algorithm implementation with hard and soft constraints.
- Administrator-only interface for initiating, reviewing, approving, rejecting, and re-queuing groups.
- Real-time reporting on unmatched participants and constraint failure reasons.
- Automated email notification system upon group approval.

2.2. Out-of-scope features (future considerations)

- Direct user (Learner/Peer) login portal or data entry interface.
- Automated time-slot booking or real-time scheduling.
- Any features related to tracking session attendance or performance data post-matching.

3. Users and user stories

3.1. Target users

The primary user of the V1.0 system is the **Program Administrator**.

User role	Description
Administrator	The single point of contact responsible for initiating, reviewing, and managing all matching runs.
Learner	Student seeking tutoring; data is ingested from the Google Sheet.
Learning Peer	Volunteer tutor; data is ingested from the Google Sheet.

3.2. User stories

ID	User story	Acceptance criteria
US-001	As an Administrator , I need a secure login/authentication mechanism so that only authorized personnel can access the matching system.	Given I navigate to the system login page, When I enter valid credentials, Then I am granted access to the Administrator dashboard, and When I enter invalid credentials, Then I am shown an "Invalid credentials" error message.
US-002	As an Administrator , I need to manually trigger a full matching run at any time so that I can iteratively process new requests throughout the initial weeks of the semester.	Given all data is ingested, When I click the "Start Matching Run" button, Then the system initiates the full matching algorithm.

US-003	As an Administrator , I need the system to ingest all participant data from the live Google Sheet via API so that the matching run uses the most current information.	Given a matching run is triggered, Then the system connects to the Google Sheet API, ingests all Learner and Peer data (schedules, expertise, availability, Instructor Match flags), and confirms successful data ingestion/update timestamp.
US-004	As an Administrator , I need to review a proposed group match (Learners + Peer + time slot) so that I can verify the match before committing.	Given a matching run has completed, Then I can view a list of proposed groups, including all Learners, the Peer, the common course, and the suggested 1-hour time slot.
US-005	As an Administrator , I need to be able to approve a proposed group so that the participants are notified and the group is finalized.	When I click "Approve" for a proposed group, Then the group is finalized, and Then the system automatically initiates the US-010 notification process.
US-006	As an Administrator , I need to be able to 'Reject and Re-Queue' a proposed group so that the participants are made available for the next matching run.	When I click "Reject and Re-Queue" for a proposed group, Then the associated Learners and the Peer are immediately returned to the available pool for future runs (US-002).
US-007	As an Administrator , I need a real-time list of unmatched participants so that I can monitor the coverage and identify problematic requests.	When I view the Unmatched Report, Then I see a list of all participants (Learners/Peers) currently unmatched, and I see the specific constraint failure reason for each Learner.
US-008	As an Administrator , I need the system to suggest an optimal alternative group/time for Learners who failed due to a single constraint so that I can facilitate manual placement.	Given a Learner failed to match due to a single constraint (e.g., Instructor Match conflict), Then the Unmatched Report provides a suggested time/group that satisfies all <i>other</i> constraints.
US-009	As a Learner , I need to be grouped with 1 to 4 Learners taking the same course so that the session is relevant to my studies.	Given a matched group, Then the group size is ≤ 4 Learners and ≥ 1 Learner for the same course.

US-010 As a **Learning Peer** and **Learner**, I need to **receive an automated email notification** simultaneously upon Admin approval so that I know my scheduled session details.

Given an Admin approves a group (US-005), Then a single, comprehensive email is sent to all participants detailing the 1-hour time slot (day/time), course name, and a clear list of all participants labeled as 'Learners' or 'Learning Peers'.

4. System requirements

4.1. Data integration

- **Primary data source:** Live Google Sheet.
- **Integration method:** Secure API connection.
- **Data ingestion trigger:** Must be automatically executed prior to or upon manual initiation of a matching run (US-002).

4.2. Core matching logic and constraints

The matching algorithm must identify the **optimal recurring, 1-hour weekly time slot** for the duration of the semester that satisfies all constraints for a group of 1–4 participants.

4.2.1. Hard constraints (non-negotiable)

Constraint type	Rule	Details
Group size	1 to 4 Learners per group, same course.	A valid group must contain 1, 2, 3, or 4 Learners, and all must be enrolled in the same course (though not necessarily the same section/instructor, unless specified below).
Volunteer load	≤2 groups total per Learning Peer.	A Peer can only be assigned to a maximum of two groups across all courses they are approved to teach.
Availability window	8:00 am to 8:00 pm.	The common 1-hour time slot must begin and end within this time frame.
Travel buffer	Minimum 5 minutes.	The 1-hour session time must be separated by at least 5 minutes from any participant's scheduled class time (before and after the session).

Instructor match	Must match Peer's instructor.	If a Learner is flagged as "Instructor Match Required ('Y')", they can only be grouped with other Learners sharing the exact same instructor as the assigned Learning Peer .
-------------------------	--------------------------------------	--

4.2.2. Optimization priorities

The system must prioritize the following objectives during the matching run:

- **Priority 1 (Hard): Maximize the total number of matched students (Learners).** This is the paramount objective and takes precedence over maximizing Peer utilization if a conflict arises.
- **Priority 2 (Soft/Filtering):** The proposed 1-hour meeting time should ideally be **2 hours before or 2 hours after** any participant's scheduled class time. This reflects a preference for on-campus meetings.

5. Deployment and maintenance

5.1. Operational environment

The software will run in a secure, cloud-hosted environment to ensure high availability and scalability to handle the semester request volume (upwards of 600 requests).

5.2. Maintenance and updates

The ability to update the system with new course data, Peer expertise, and Administrator logic should be maintained through version control and clear separation of the matching algorithm logic from the administrative interface.