

Requirements Analysis Document

Optional Subjects Selection System

1. Project Overview

Project Name: Optional Subjects Selection System (OSSS)

Purpose: To automate and streamline the process of selecting and assigning optional subjects to students at the Faculty of Computer Science, Alexandru Ioan Cuza University of Iași (UAIC), using a fair allocation algorithm based on student preferences and academic performance.

Scope: This system will replace the manual process of optional subject selection using Google Forms by providing a web-based platform where students can submit their preferences, and administrators can manage the allocation process using the Gale-Shapley algorithm for optimal matching.

2. System Actors

2.1 Student

Description: Registered students of the Computer Science Faculty who need to select optional subjects.

Responsibilities:

- Authenticate using institutional email
- View available optional subjects
- Submit ranked preferences for optional subjects
- View assigned optional subjects
- Receive notifications about selection periods and assignments

2.2 Administrator (Faculty Staff)

Description: Faculty administrative staff responsible for managing the optional subjects selection process.

Responsibilities:

- Manage optional subjects (add, edit, remove)
- Define selection periods
- View all student preferences

- Execute the allocation algorithm
- Manually adjust student assignments
- Facilitate student transfers between optional subjects
- Send notifications to students

2.3 System

Description: The automated system component.

Responsibilities:

- Authenticate users via institutional email
 - Retrieve student matriculation numbers from institutional API
 - Store and manage preferences
 - Execute Gale-Shapley allocation algorithm
 - Process mutual transfer requests
 - Send automated email notifications
-

3. Functional Requirements

3.1 Authentication and Authorization

FR-1.1: Institutional Email Login

- The system shall provide authentication through institutional email
- The system shall integrate with the university's authentication service

FR-1.2: Automatic Student Data Retrieval

- Upon successful authentication, the system shall automatically retrieve the student's matriculation number from the institutional API
- The system shall populate student profile information automatically without requiring manual input

FR-1.3: Role-Based Access Control

- The system shall support two distinct roles: Student and Administrator
- The system shall restrict access to functionalities based on user roles

3.2 Student Functionality

FR-2.1: View Available Optional Subjects

- Students shall be able to view a list of all available optional subjects for their academic year and semester
- Each optional subject shall display: name, description, coordinator

FR-2.2: Submit Preference Rankings

- Students shall be able to create a ranked list of preferred optional subjects
- The first subject in the list shall represent the most preferred option, with decreasing preference order
- Students shall be able to reorder their preferences before the deadline
- The system shall validate that students select the required number of preferences

FR-2.3: View Assignment Results

- Students shall be able to view their assigned optional subjects after the allocation process
- The system shall display the assignment status clearly

FR-2.4: Request Subject Transfer

- Students shall be able to request a transfer from their assigned optional subject to another
- The system shall track all transfer requests

3.3 Administrator Functionality

FR-3.1: Manage Optional Subjects

- Administrators shall be able to create, edit, and delete optional subjects
- For each subject, administrators shall define: name, description, coordinator, capacity and target academic year/semester

FR-3.2: View Student Preferences

- Administrators shall be able to view all submitted student preferences
- The system shall provide filtering and sorting capabilities by student, subject, or preference rank

FR-3.3: Execute Allocation Algorithm

- Administrators shall be able to trigger the Gale-Shapley allocation algorithm
- The algorithm shall consider student preferences and academic grades
- The system shall generate an initial assignment based on the algorithm results

FR-3.4: Manual Assignment Adjustments

- Administrators shall be able to manually modify student assignments when necessary

FR-3.5: Process Mutual Transfers

- The system shall identify mutual transfer requests (e.g., 10 students wanting to transfer from Subject A to B, and 10 students from B to A)
- Administrators shall be able to review and approve bulk mutual transfers
- The system shall execute approved transfers atomically

3.4 Notification System

FR-4.1: Email Notifications

- The system shall send automated email notifications to students for:
 - Opening of preference submission period
 - Upcoming deadline reminders
 - Completion of subject assignment
 - Approval/rejection of transfer requests

3.5 Allocation Algorithm

FR-5.1: Gale-Shapley Implementation

- The system shall implement the Gale-Shapley stable matching algorithm
- The algorithm shall use student preferences as one input
- The algorithm shall use student academic grades as the ranking mechanism for subjects
- The algorithm shall respect subject capacity constraints

FR-5.2: Constraint Handling

- The system shall enforce subject capacity limits
- The system shall ensure each student is assigned to the required number of optional subjects

4. Use Cases

Use Case 1: Student Submits Preference List

Actor: Student

Preconditions:

- Student is authenticated
- Preference submission period is active
- Student has not finalized their preferences

Main Flow:

1. Student logs in using institutional email
2. System retrieves and displays student's matriculation number
3. Student navigates to preference submission page
4. System displays available optional subjects
5. Student selects subjects and arranges them in preference order
6. Student reviews and confirms the preference list
7. System validates and saves the preferences
8. System displays confirmation message

Postconditions:

- Student preferences are stored in the database
- Student can modify preferences until the deadline

Alternative Flows:

- If student tries to submit after deadline, system displays error message
 - If student selects invalid combination, system shows validation errors
-

Use Case 2: Administrator Executes Allocation Algorithm

Actor: Administrator

Preconditions:

- Preference submission period has ended
- All student preferences and grades are in the system

Main Flow:

1. Administrator logs into the system
2. Administrator navigates to allocation management page
3. System displays statistics: total students, total subjects, submission completion rate
4. Administrator reviews configuration parameters
5. Administrator triggers allocation algorithm
6. System executes Gale-Shapley algorithm using preferences and grades
7. System generates assignment results
8. System displays allocation statistics and potential issues
9. Administrator reviews results
10. Administrator confirms and finalizes assignments
11. System sends notification emails to all students

Postconditions:

- All students are assigned to optional subjects
- Assignments are visible to students

Alternative Flows:

- If algorithm encounters errors, system displays error details
 - Administrator can abort and retry with adjusted parameters
-

Use Case 3: Process Mutual Transfer Requests

Actor: Administrator

Preconditions:

- Initial assignments have been made
- Multiple students have submitted transfer requests

Main Flow:

1. Administrator navigates to transfer management page
2. System displays all pending transfer requests
3. Administrator selects mutual transfers to approve
4. System validates that the transfers maintain capacity constraints
5. Administrator confirms batch transfer
6. System executes all transfers atomically
7. System sends confirmation emails to affected students

Postconditions:

- Student assignments are updated
- Transfer requests are marked as completed

Alternative Flows:

- Administrator can approve partial transfers or individual requests
-

5. Data Requirements**5.1 Student Data**

- Matriculation number (număr matricol)
- Name
- Email address
- Academic year and semester
- Current grades
- Assigned optional subjects

5.2 Optional Subject Data

- Subject ID
- Subject name
- Description
- Coordinator name
- Capacity (maximum number of students)
- Target academic year/semester

5.3 Preference Data

- Student ID
- Subject ID
- Preference rank (1 = most preferred)
- Submission timestamp

5.4 Assignment Data

- Student ID
- Subject ID
- Assignment type (algorithm-generated, manual, transfer)
- Administrator ID (if manual)

5.5 Transfer Request Data

- Request ID
- Student ID
- From Subject ID
- To Subject ID
- Request timestamp
- Status (pending, approved, rejected)

6. Constraints and Assumptions

Constraints

- Must integrate with existing university authentication system

- Must comply with GDPR and university data protection policies
- Development timeline: one academic semester

Assumptions

- Institutional API for student data is available and documented
- University email system can be used for notifications
- Students have access to computers/devices with web browsers
- Academic grades are available in digital format
- Internet connectivity is available to all users