

UniProgrammes

Requirements Definition

Sprint #1 – 15/10/2024

Date	Sprint #	Version #	Written by	Reviewed by
15/10/2024	1	0	Ali Hajizadeh Idloo	Andrea Cataluffi, Giuseppe Vitello, Satvik Sharma

The rest of this page was intentionally left blank.

SUMMARY

1. Introduction	3
1.1 Purpose	3
1.2 Scope	3
1.3 Requirements Gathering Process	3
2. Overall Description	4
2.1 Product Perspective	4
2.2 Product Functions	4
2.3 User Classes and Characteristics	4
2.4 Operating Environment	4
2.5 Assumptions and Dependencies	4
3. Functional Requirements	5
3.1 Data Collection	5
3.2 Static Visualization	5
3.3 Programme Reorganization	6
3.4 User Management	8
4. Non-Functional Requirements	9
5. Use Case Diagram	11
6. User Stories	12
6.1 User Stories for the User Role	12
6.2 User Stories for the Administrator Role	12
7. Future Considerations	13

1. Introduction

1.1 Purpose

This document outlines the definition of requirements for UniProgrammes. This tool is designed to assist university students and counselors in managing programme data, offering features for programme reorganization, validation of course requirements, and visual representation of programmes.

1.2 Scope

The UniProgrammes will allow university students and counselors to gather, visualize, and manage programme data, supporting the reorganizing of the programme. It will ensure that learning outcomes and course prerequisites are validated. The system will not handle financial aspects, student enrollments, or grading systems.

1.3 Requirements Gathering Process

The requirements for this project were initially identified through a combination of sources, including the project description, project presentation, one meeting with customers, one meeting with supervisors, and internal discussions within the development team. This collaborative process resulted in the creation of the High-Level Overview of the Requirements document, which serves as the foundation for defining the system's functionality.

As the project progresses, the requirements will be continuously reviewed and refined through subsequent meetings with the customer. Based on the evolving needs of the project, this document will be updated accordingly, with versioning applied to track changes and ensure that all stakeholders are aligned.

2. Overall Description

2.1 Product Perspective

The UniProgrammes will help university students and counselors in the process of collecting course data from university systems, visualizing course structures, and allow to modify and validate programme plans. The tree-based visualizations of the programme lead to identifying the plan's progression and the prerequisites of each course. In general, the UniProgrammes is a web-based solution designed to integrate with university data sources for the above-mentioned goals.

2.2 Product Functions

- Collect programme and course information from universities' sources.
- Display tree hierarchies of programmes and visualize relationships between courses.
- Validate course and programme requirements during reorganization.
- Analyze learning outcome progression and ensure degree requirements are met.
- Allow highlighting courses based on the main area, period/year, and learning categories.

2.3 User Classes and Characteristics

- Administrators: Responsible for managing programme structures and validating course requirements.
- Students/Counselors: Reorganize and modify courses to ensure compliance with degree requirements, and view programme progress and learning outcome pathways.

2.4 Operating Environment

- Accessible via standard web browsers like Chrome, Firefox, and Edge.
- Hosted on the infrastructures that support scaling the application.

2.5 Assumptions and Dependencies

The system assumes that universities will provide access to their courses/programmes data using APIs or datasets. The stability of these external data sources is a key dependency for proper functionality.

3. Functional Requirements

3.1 Data Collection

Requirement ID: DC-01

Title: Course Data Collection

Description: The system shall collect and parse course information (credits, educational level, descriptions, etc.) from university sources.

Priority: High

Acceptance Criteria: The system successfully collects and stores course data for all available courses, with no manual intervention required. The data is accurately displayed in the system's interface.

Requirement ID: DC-02

Title: Programme Data Collection

Description: The system shall collect and store programme-specific information, including specific course requirements and main areas of study.

Priority: High

Acceptance Criteria: Programme information is retrieved and displayed correctly, and users can see accurate course requirements and main areas for each programme.

Requirement ID: DC-03

Title: Learning Outcome Collection

Description: The system shall identify and store learning outcomes (e.g. electronics, group work, report writing) for each course based on the course information.

Priority: Medium

Acceptance Criteria: Learning outcomes are collected for every course, and users can access this information for each course within the system.

3.2 Static Visualization

Requirement ID: SV-01

Title: Programme Tree View

Description: The system shall provide a visual, hierarchical tree view of all courses in a programme.

Priority: High

Acceptance Criteria: Users can view all courses in a hierarchical tree format, showing dependencies and relationships between courses.

Requirement ID: SV-02

Title: Course Highlighting by Main Area

Description: The system shall allow users to highlight courses by their main area (e.g., ELA, DVA) within the programme tree.

Priority: Medium

Acceptance Criteria: Users can select and highlight courses in the tree based on their main area, and the highlighted courses are visually distinct.

Requirement ID: SV-03

Title: Course Highlighting by Learning Outcomes

Description: The system shall allow users to highlight courses in the programme tree based on specific learning outcomes (e.g., group work, report writing).

Priority: Medium

Acceptance Criteria: Users can filter and highlight courses by learning outcomes, and the highlighted courses are visually distinct in the programme tree.

Requirement ID: SV-04

Title: Course Highlighting by year/period

Description: The system shall allow users to highlight courses in the programme tree based on specific year/period

Priority: Medium

Acceptance Criteria: Users can filter and highlight courses by year/period, and the highlighted courses are visually distinct in the programme tree.

3.3 Programme Reorganization

Requirement ID: PR-01

Title: Course Addition in Programme Tree

Description: The system shall allow users to add new courses to the programme tree, even if they break the requirements.

Priority: High

Acceptance Criteria: Users can successfully add courses to the programme tree, and the tree automatically updates to reflect these changes.

Requirement ID: PR-02

Title: Course Removal from Programme Tree

Description: The system shall allow users to remove existing courses from the programme tree, even if they break the requirements.

Priority: High

Acceptance Criteria: Users can successfully remove courses from the programme tree, and the tree is updated to reflect the removal.

Requirement ID: PR-03

Title: Course Replacement in Programme Tree

Description: The system shall allow users to replace existing courses in the programme tree with other courses, even if they break the requirements.

Priority: High

Acceptance Criteria: Users can replace courses in the tree, which is automatically updated to reflect the replacement, showing new course dependencies.

Requirement ID: PR-04

Title: Course Requirement Validation

Description: The system shall automatically validate whether the courses added, removed, or replaced meet the programme's requirements.

Priority: High

Acceptance Criteria: When users make changes, the system validates course requirements and flags any violations with a clear notification to the user.

Requirement ID: PR-05

Title: Course Highlighting of Requirement Failures

Description: The system shall highlight courses that do not meet the programme's requirements after changes are made.

Priority: Medium

Acceptance Criteria: Courses that fail to meet requirements are highlighted with a distinct visual indicator (e.g., red highlight) immediately after changes are made.

Requirement ID: PR-06

Title: Programme Submission

Description: The system should allow the submission of the programme.

Priority: Medium

Acceptance Criteria: The system should allow the submission only if the programme meets the requirements

3.4 User Management

Requirement ID: UM-01

Title: User Authentication

Description: The system shall allow new users to Authenticate by providing their personal information (e.g., name, email, password), creating a new account, and logging in to the platform.

Priority: Medium

Acceptance Criteria: Users can create a new account by entering valid details (e.g., unique email, strong password). The system validates the input, displays relevant error messages for invalid input, and creates an account upon successful validation. The user with these credentials can log in to the system, The system authenticates the credentials and grants access to the user dashboard.

Requirement ID: UM-02

Title: Change User Credentials

Description: the system must allow the user to modify his credentials.

Priority: Medium

Acceptance Criteria: Users can successfully change their credentials (e.g., unique email, strong password). The system validates the input, displays relevant error messages for invalid input, and updates the credentials upon successful validation.

4. Non-Functional Requirements

Requirement ID: NF-01

Title: Interoperability - Data Retrieval

Description: The system should manage the import of different university programs, the different formats sent by the universities, the different types of data that can arrive, and in general all the changes that these data can have between the different universities. The prototype will be based on MDU requirements.

Priority: Medium

Acceptance Criteria: The systems will retrieve the data from the university in a file-based way and store correctly in the data layer. The file-based imports will be from a CSV file.

Requirement ID: NF-02

Title: Usability - Intuitive Interface

Description: The interface must be intuitive, allowing average users to perform all the operations that the system offers (like programme reorganization).

Priority: High

Acceptance Criteria: Users can perform core tasks, including programme reorganization, with basic guidance and support materials (e.g., tooltips, and tutorials) without extensive training.

Requirement ID: NF-03

Title: Maintainability - Modular Architecture

Description: The system will have a modular architecture, allowing easy updates and maintenance.

Priority: High

Acceptance Criteria: The system's modular design enables developers to add or modify components without impacting overall functionality. Updates must be executed without disrupting service availability, ensuring that maintenance can be performed seamlessly and without downtime.

Requirement ID: NF-04

Title: Security - Security

Description: The system stores the personal information of its users. As such, it is critical to maintain state-of-the-art standards in terms of cybersecurity. Data exchange must be

performed using HTTPS tunnels, and the database chosen for system implementation must support encryption-at-rest to store user data in an encrypted manner.

Acceptance Criteria: Adopting the secure protocols.

Requirement ID: NF-05

Title: Security - Data Processing

Description: The entire System must respect all the laws regarding privacy and data treatment, it must respect the EU GDPR.

Acceptance Criteria: Follow the GDPR rules.

Requirement ID: NF-06

Title: Security - Password Security

Description: the system must ensure that passwords are used which cannot be broken with a longer time than would be tried with brute force.

Acceptance Criteria: The system requires that the user not use passwords that may be references to him (for example, his dog's name), that must not be among the most common passwords (for example, use password as a password), and that must contain a certain number of characters. The password must be changed periodically.

Requirement ID: NF-07

Title: Availability - Non-Disruptive Updates

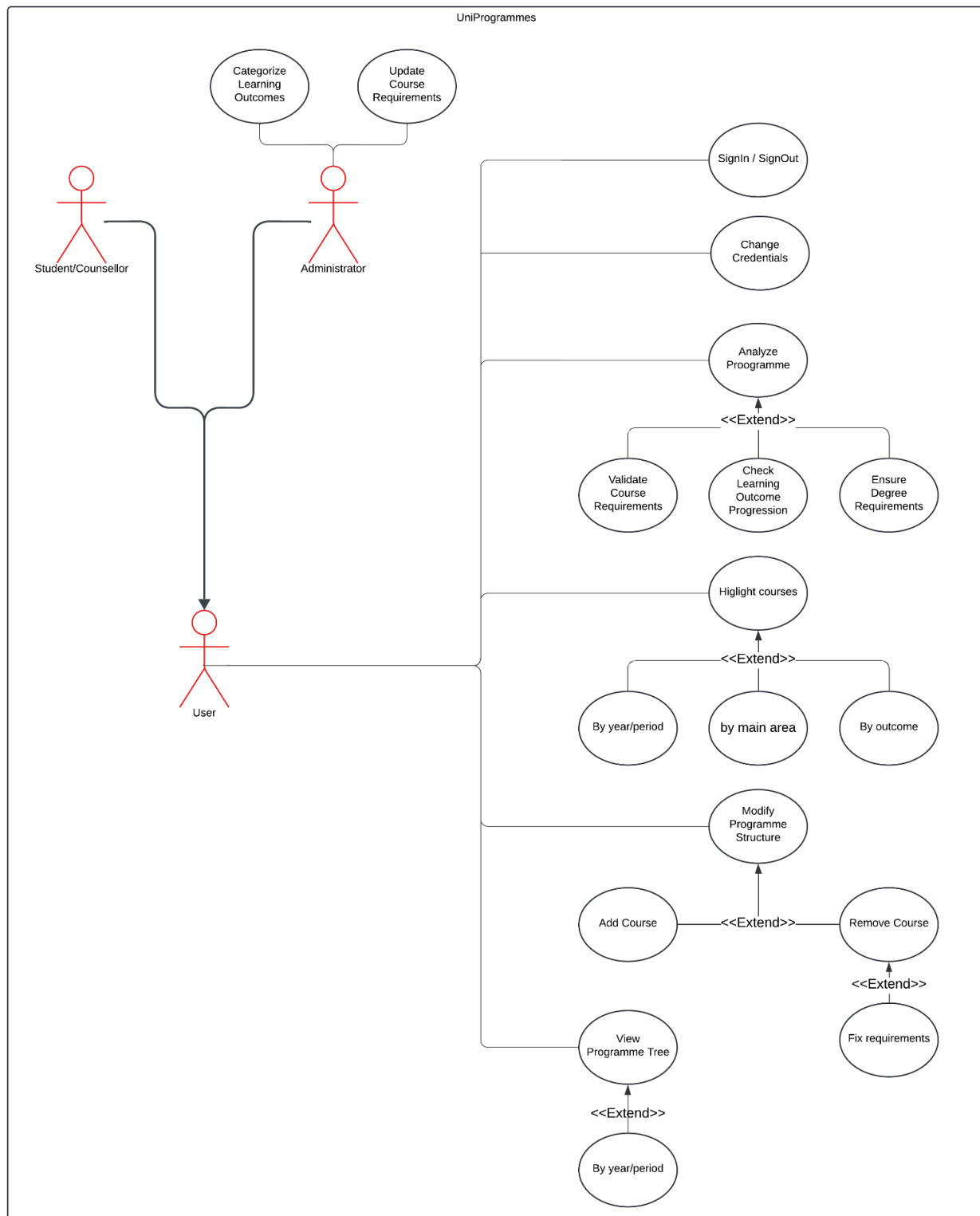
Description: The system must ensure that updates to software components, including patches, upgrades, or new feature integrations, do not disrupt service availability for end-users. Updates should be scheduled and implemented in a way that minimizes or eliminates downtime.

Priority: High

Acceptance Criteria: Updates are completed with minimal service interruption, and any planned downtime is communicated to users in advance, ensuring uninterrupted core functionality for the majority of users.

5. Use Case Diagram

Use-Case Diagram



6. User Stories

6.1 User Stories for the User Role

1. **Login/Logout**
 - As a user, I want to log in and log out of the system, so I can access and secure my data.
2. **Change Credentials**
 - As a user, I want to change the credentials of the system, so I modify my password or email.
3. **Analyze Program**
 - As a user, I want to analyze the program, so I can get an overview of the study path.
4. **Validate Course Requirements**
 - As a user, I want to validate course requirements, so I can ensure I meet the necessary prerequisites to progress in the program.
5. **Ensure Degree Requirements**
 - As a user, I want to ensure that degree requirements are met, so I can complete the program successfully.
6. **Check Learning Outcome Progression**
 - As a user, I want to check my progress toward learning outcomes, so I can track my advancement and complete the program.
7. **Highlight Courses (by Main Area, Year/Period, or Outcome)**
 - As a user, I want to highlight courses by main area, year/period, or outcome, so I can easily identify relevant courses in my program.
8. **Modify Program Structure**
 - As a user, I want to modify the program structure to add or remove courses according to my needs.
9. **View Program Tree**
 - As a user, I want to view the program tree by year or period, so I can see a clear representation of the program structure.
10. **Fix Requirements When Removing a Course**
 - As a user, I want to adjust requirements whenever a course is removed to maintain program integrity and consistency.

6.2 User Stories for the Administrator Role

11. **Update Course Requirements**
 - As an administrator, I want to update course requirements, so I can ensure they are up-to-date and aligned with the curriculum.
12. **Categorize Learning Outcomes**

- As an administrator, I want to categorize learning outcomes, so I can organize and manage learning objectives effectively within the system.

7. Future Considerations

Advanced Analytics: The system could incorporate advanced analytics tools that analyze historical data, such as student success rates, to suggest the optimal arrangement of courses within a programme. This would help planners improve curriculum design based on past performance.

Collaborative Features: Future versions of the system could introduce real-time collaboration, allowing multiple users (e.g., administrators, faculty members) to work together on programme planning and adjustments. This would streamline teamwork and improve efficiency.

Device Compatibility: The system could be further developed to ensure compatibility with mobile and tablet devices, providing users with the flexibility to access the platform on different devices and maintain productivity on the go.

Predictive Analytics: AI-based predictive analytics could be introduced to forecast the impact of course arrangements on student outcomes. This feature would allow the system to recommend improvements to programme structures, enhancing academic success.

Integration with External Systems: The system could integrate with learning management systems (LMS) or student information systems (SIS) in the future, enabling seamless data sharing and synchronization, reducing manual data entry, and improving operational efficiency.