* ﻿﻿Presentation of all project members and team organization (your implementation of SCRUM, roles, who is doing what at the current stage...)
* ﻿﻿High-level overview of the requirements - why do we need this project
* ﻿﻿High-level overview of the solution - high-level system architecture, what elements would the solution contain, what technologies are (considered) to be used
* ﻿﻿Quality assurance procedures (defined policies, such as code repository usage policy, coding convention policy ...)
* ﻿﻿Risk assessment analysis
* ﻿﻿Communication (what forms of contact will you be using, internal communication routines)
* ﻿﻿What the project is NOT going to address
* ﻿﻿(Short) presentation of Sprint #0 report
* ﻿﻿What was done in this one-week sprint
* ﻿﻿How much time did each member spend (including meetings, research, etc.)
* ﻿﻿Other issues and problems encountered if

It remains:

- Quality Assurance procedures

- Risk assessment

- High-Level Overview

of the Solution

Tech-stack will be decided after consultation with the customers whether they have a specific preference or not. The core features and out-of-scope points are as follows:-

### **High level overview…….**

### **Core Features:**

1. Automated data collection from university websites *(Need to check with the customer whether we need a scraping mechanism or not*)
2. Course and programme analysis, including learning outcome categorization
3. Interactive graphical representation of programmes
4. Requirement validation for courses and degrees
5. User-friendly interface with programme tree view and course management tools
6. Advanced analysis features for programme consistency and requirement checks
7. Technical integration with university systems and data processing algorithms
8. Focus on performance, scalability, and usability
9. Requirement tracking and visual planning capabilities
10. Future-ready design with potential for system integration and advanced analytics

### **Out of Scope:**

1. Student enrollment management
2. Financial aspects of programme planning
3. Curriculum content creation
4. Student performance tracking or grading systems
5. Integration with external accreditation bodies
6. Direct modifications to university websites or databases

### **Quality assurance and Risk Assessment Analysis**

### **Three Environments: Development, QA, and Production** ➝ The application will be deployed and tested across three environments: Dev for ongoing development, QA for testing and validation, and Prod for live, stable releases. This setup will ensure smooth transitions between stages and minimize the risk of issues in production.

**Continuous Integration (CI)** in Pull Requests ➝ CI will run automatically on every pull request to check for linter errors (code style consistency) and type errors (correct data type usage). This will ensure that code quality and integrity are maintained before merging.

**Pull Request Approval Process** ➝ To merge any pull request into Dev, QA, or Prod, at least one approval will be required from a team member. This will ensure code is reviewed for quality and correctness at each stage.

**Unit + E2E Testing** ➝ Unit tests will be executed as part of the CI pipeline to validate individual functions or components, ensuring that the new code doesn’t introduce bugs and performs as expected, E2E allow testing the user interaction.