Project in Software Engineering Course

School of Electrical and Computer Engineering
National Technical University of Athens

Fall Semester 2022-2023

Prof. V.Vescoukis

Subject Area

The project's subject area is the management of "intelligent questionnaires" for conducting surveys of any kind. A questionnaire is defined as "smart" when each subsequent question and its answers can be determined by the answer of the previous one. Both questions, answers and transition rules must be parametrically defined, i.e. the user can provide the options and paths and the software can be initialized and run through an appropriate interface to collect the data. There are several web services that function in this particular way, like Google forms, SurveyMonkey, typeform, kwiksurveys, surveyplanet, etc.

Two question categories are defined; those concerning the profile of each individual participating in the survey and those which are the pure content of the survey. The answers to the questions of both categories are available to the user of the system in a machine-processable format that can be easily handled by Python, R, Matlab, or even Excel, PowerBI, SPSS, etc.

Description

The objective of this project is the development of a software application for the definition, configuration and execution of "intelligent questionnaires", called "intelliQ". IntelliQ will be used to aggregate data from such questionnaires in various surveys. In the context of this particular project, students will do the identification and specification of requirements, the design, the definition of the architecture, implementation of a selected limited set of features, testing and, last but not least, generation of the documentation of their work. Requirements specification and design, code management, testing and project management will be done by using appropriate tools, as will be discussed in the sequel. The documents to be delivered will be produced automatically by these tools, so there will be no manual document editing of any kind.

The application consists of:

1. A back-end subsystem, which will implement functions of managing the structure of questionnaires as well as downloading and exporting answers. These functionalities will be available through a REST API, the specifications of which will be provided to you.

2. A CLI (Command Line Interface) application for the above operations, which will act as a REST API client of the back-end subsystem, offering its user the ability to perform operations through an OS shell.

Teams

The project will be done in teams of 4-5 students, who will implement the full system development cycle (requirement analysis, specification writing, design and architecture, implementation and acceptance checks, installation and operation). It is advisable Erasmus students to make their own team.

Technical Requirements - Tools

- **GitHub** as a code management environment. Erasmus students need to create their own team's repository and send the link to the course's staff at softeng@courses.ntua.gr.
- **Visual Paradigm** as a UML diagram generation tool. Either Visual Paradigm Enterprise trial version (free for one month) or the "Community Edition" can be used.
- **GitHub** for project management, using the relevant functionality offered by the platform.

Development stack

You may select your own stack, however you are strongly advised to consider the following:

- **Code implementation**: Python, JavaScript (NodeJS/Express)
- **Data Management**: one of MySQL, MariaDB, PostgreSQL, Mongo.
- **Frontend:** no frontend needs to be implemented by Erasmus students.

Deliverables

Deliverables List

Deliverable	
Documentation	
ER Diagram or NoSQL JSON schemas	YES ✓
UML Activity / State Diagrams	as needed
UML Class/API Diagrams	YES
UML Sequence Diagrams	YES
UML Deployment Diagrams	YES
UML Component Diagrams	YES
Implementation	
Back-End functionalities	YES ✓
Database dump (sql or json)	YES ✓
RESTful API	YES ✓
API documentation - demo	YES ✓
Command line interface (CLI)	for teams of 3 or more persons
Executable format	YES
Testing	
API functional tests	YES

Deliverable formats

The required format of the deliverables is shown in the table below.

Deliverable	Format	Deliverable file name	Comment
Documentation			
StRS - Stakeholder Req. Specification	docx or pdf automatically produced from Visual Paradigm	strs-softeng-XX.zip	The documents should also be present in the VPP file
SRS - Software Req. Specification		srs-softeng-XX.zip	
ERD or NoSQL JSON schemas	Single Visual Paradigm file (.vpp)	softeng22-XX.vpp	**Important!** One (1) single VPP file for all. Diagrams from design software or more than one VPP file are not accepted.
UML Activity / State Diagrams			
UML Class/API Diagrams			
UML Sequence Diagrams			
UML Deployment Diagrams			
UML Component Diagrams			
Implementation			
Import, manage and access data (backend)	Src code in the implementation language	(according to the development environment)	A README.md file should be included, providing a brief description of the application's components and the steps required to set up the development environment (libraries, etc).
Database dump (SQL or JSON)	SQL ή json		
RESTful API	Src code in the implementation language		
API documentation - demo	Postman scripts		
Command line interface (CLI)	Src code in the implementation language		
Executable format (build & deploy your code from source)			
Testing			
API functional tests	test scripts	(according to the development environment)	

Deadlines

1st deliverable: Includes the entire work. The deadline is until **midnight the day before the exam**.

Grading

The course grade is 100% derived from the assignment. The assignment is graded as follows:
- Documentation 35%
- Implementation 35%
- Testing 15%
- Use of tools 15%
- General image of your work 10%