<inteliQ>

Software Requirements Specification

Version <1.1>

Revision History

Date	Version	Description	Author
<MM/DD/YYYY>	<version>	<description>	<author>
12/17/2022	1.0	SRS	Team 22-17
02/14/2023	1.1	SRS of renewed project	Team 22-17

Table of Contents

1. Introduction

Our software aims at creating an intelligent forms' answering application where each selected option of a question leads to a specific next question. Our software consists of a frontend application for the users of our application (participants) and a CLI, as well as a frontend application, that can be used by an administrator (admin) to manage the system.

The participants can view existing surveys and answer them, while the admin can create, edit or delete a questionnaire, as well as retrieve statistical data concerning the answers given or execute healthchecks of the system.

1.1 Overall Description

1.1.1 Problem Statement

Creation of a questionnaire that can be dynamically configured as each participant is answering it.

1.1.2 Objectives

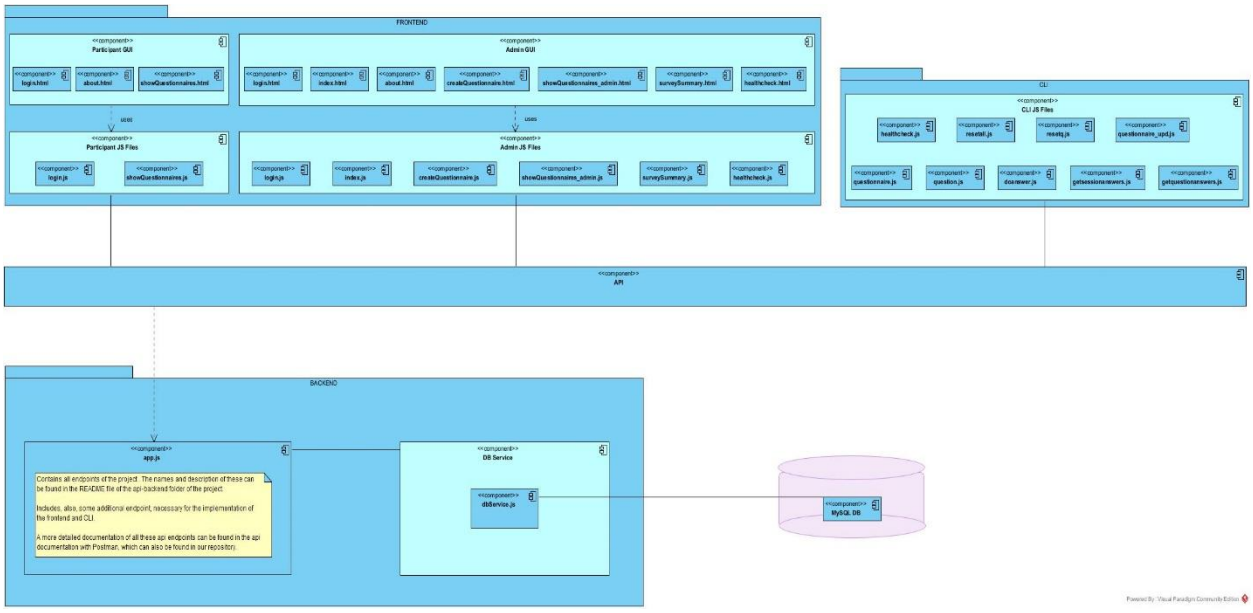
A convenient and easy to use User Interface in order to answer questionnaires and a CLI so that the System Administrator can easily interact with the data (questionnaires and their questions and answers) and overall manage the application.

1.2 Interfaces

1.2.1 Interfaces with internal systems

- A Javascript Web Server used for the frontend
- A Rest API used for the communication between the backend and the frontend or the CLI
- MySQL Database used for storing data.

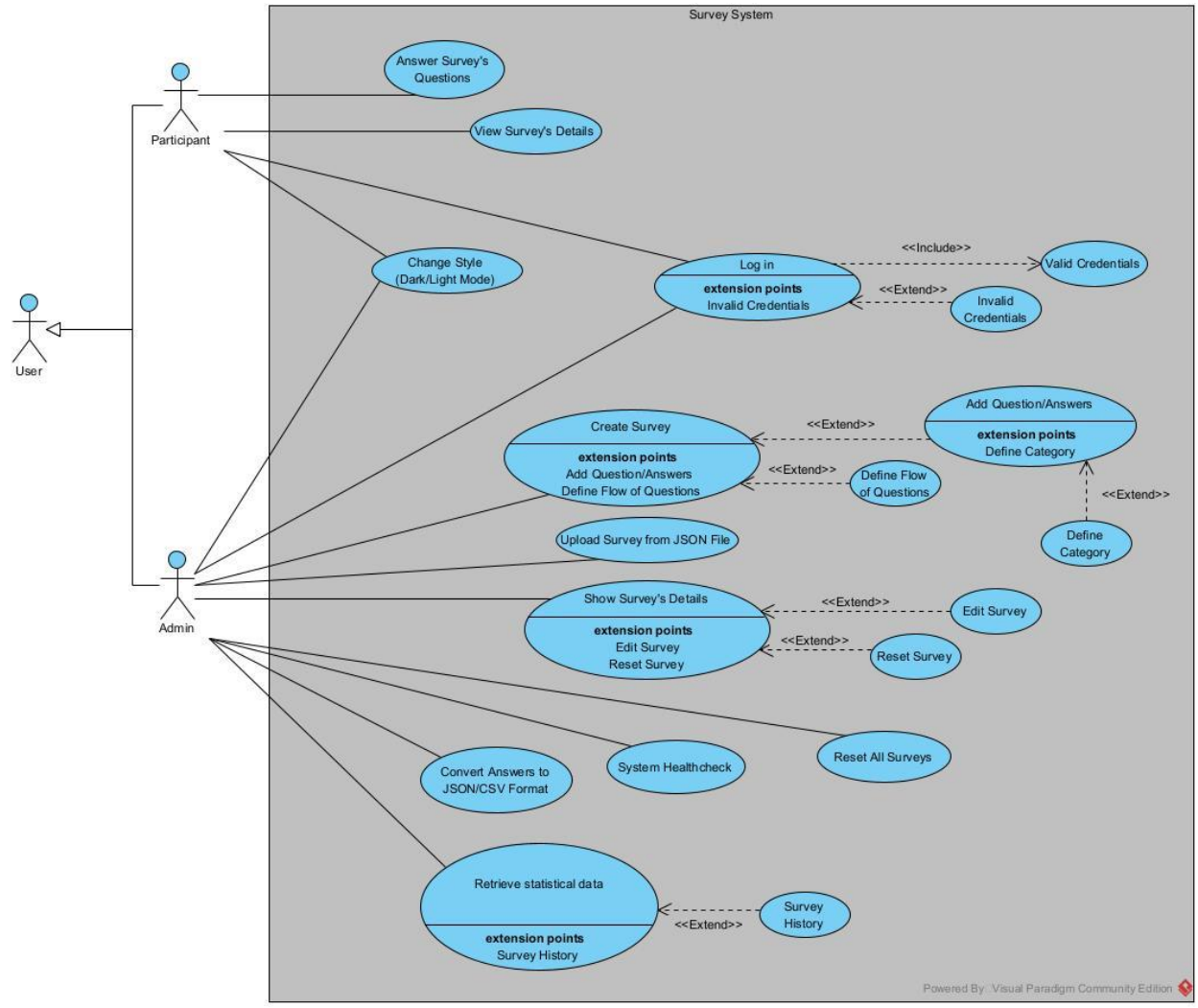
Component Diagram



1.2.2 Interfaces with Users

Below we can see the Use Case Diagram which describes all possible use cases with the respective roles/actors.

Use Case Diagram



1.3 References

N/A

2. Functional Requirements – Use Cases

Below, we will describe two possible use cases.

2.1 Use Case 1: Answer Survey's Questions

2.1.1 Roles

Participant in a specific questionnaire.

2.1.2 Prerequisites
Logged in as a user

2.1.3 Execution Environment
The participant can answer the questionnaire of his selection using the frontend Web Server which in its turn communicates with the backend server and the Database to render the Website and provide the user with the necessary data. In this case, no extra communication with an External Server is needed.

2.1.4 Input Data
Each User chooses the desired answer in each question and then presses the <Next> Button in order to answer the next question his answer will lead to, until the questionnaire is fully filled.

2.1.5 Parameters
After the participant begins answering the questionnaire, the following parameters must be transported from the frontend to the backend for further processing:

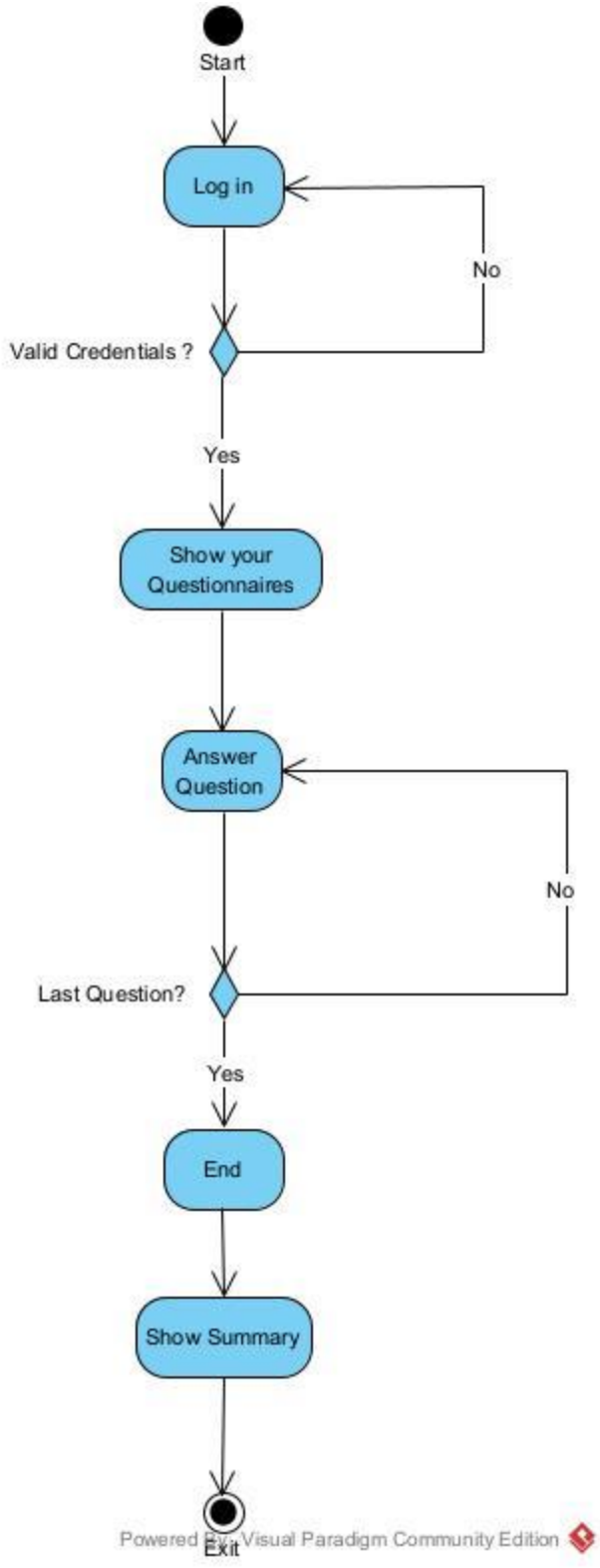
- Questionnaire Information: Transfer the ID of the questionnaire the user is currently answering
  - ✓ questionnaireID: String
- Question Information: Transfer the chosen's question's details to the backend to be further transferred to the database.
  - ✓ questionID: String
  - ✓ answerID: String
  - ✓ sessionID: String

2.1.6 Sequence of Actions – Sequence/Activity Diagram
Below we can see the Sequence and Activity Diagrams for this Use Case.

(The Sequence Diagram contains of course several Use Cases, among which is also the aforementioned)

Participant - Sequence Diagram

Answer Survey's Questions



2.1.7 Output Data

After each question has been answered, our application must render the next question in the Web Page (i.e. next_quesionID, question_text, options…) and finally when the whole questionnaire is answered a summary of all selected answers is shown to the user.

2.2 Use Case 2: Create a Survey

2.2.1 Roles

Administrator of a specific questionnaire.

2.2.2 Prerequisites

Logged in as an admin (in case of using the frontend application)

2.2.3 Execution Environment

The administrator can create a questionnaire, either by using the CLI and its respective commands or by using the admins frontend application. In both cases, a communication with the backend server is followed, which in its turn communicates with the Database to post the newly created questionnaire's data.

2.2.4 Input Data

- CLI: The admin has to use a specific CLI command (questionnaire_upd –source – format) in order to post the questionnaire from a JSON file (which requires a specific form to be correctly read – where the questions, answers and the flow is determined)
- Frontend: The admin has to name the questionnaire and optionally a keyword following the input of the questions, their category, whether they are profile questions, whether they are required, and their respective answers and ultimately they determine the flow of the questions.

2.2.5 Parameters

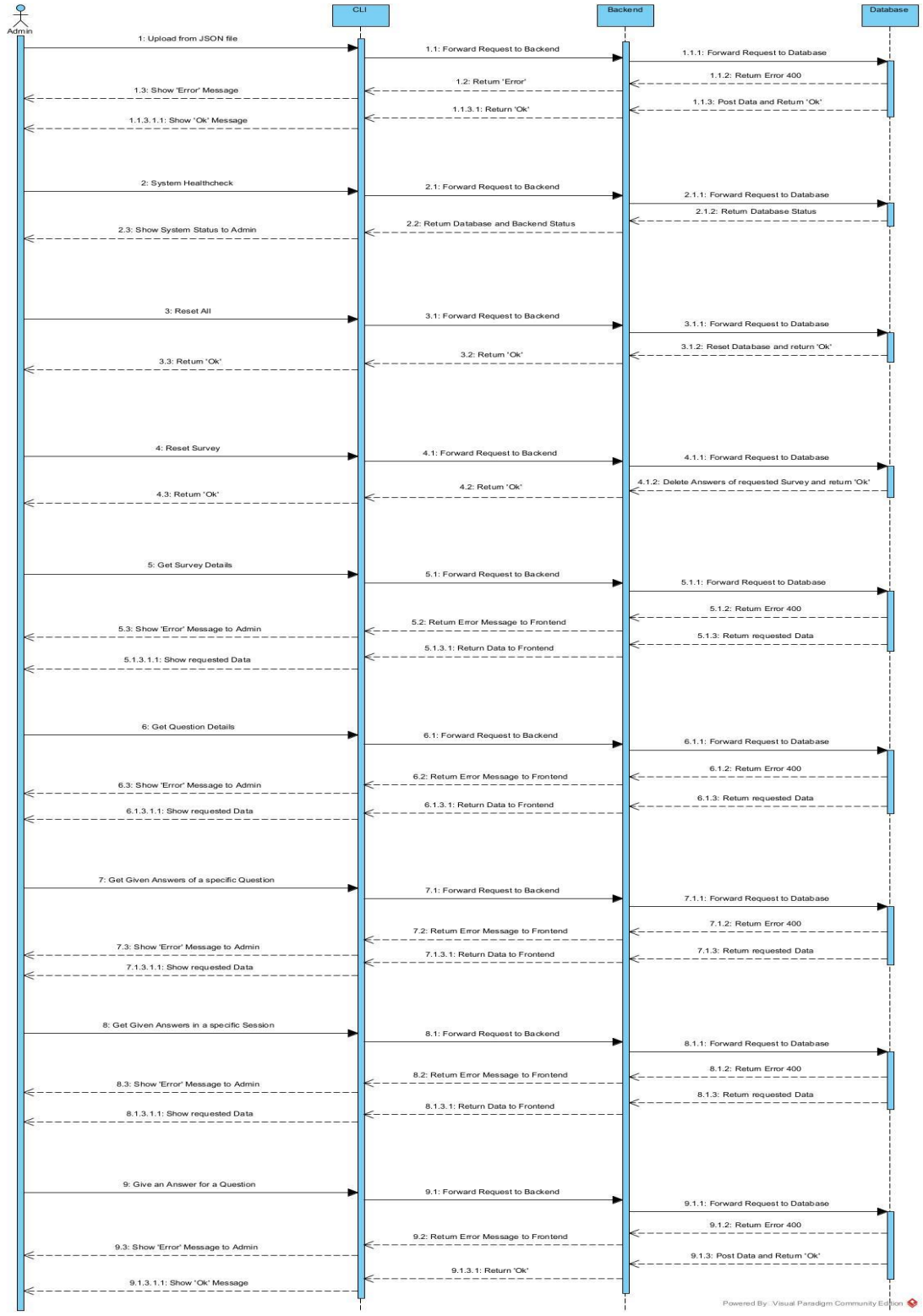
- Questionnaire Information: Transfer of the ID of the questionnaire the admin creates
  - ✓ questionnaireID: String
  - ✓ questionnaireTitle: String
  - ✓ keyword: String
- Question Information: Transfer the questionnaire's details to the backend to be further transferred to the database.
  - ✓ questionID: String
  - ✓ questionTitle: String
  - ✓ required: Bool
  - ✓ type: Bool
  - ✓ options: List

2.2.6 Sequence of Actions – Sequence/Activity Diagram

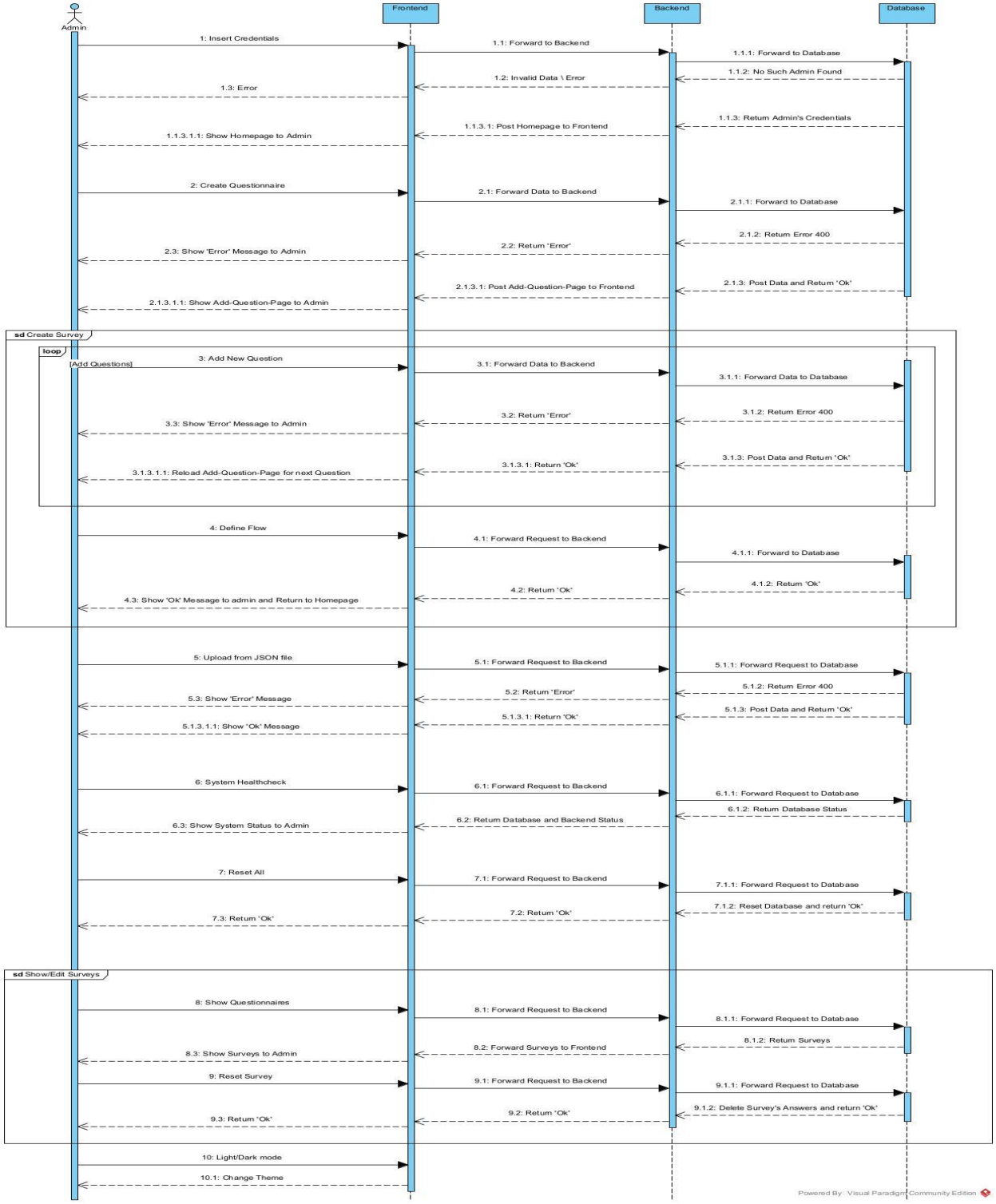
Below we can see the Sequence and Activity Diagrams for this Use Case.

(The Sequence Diagram contains of course several Use Cases, among which is also the aforementioned)

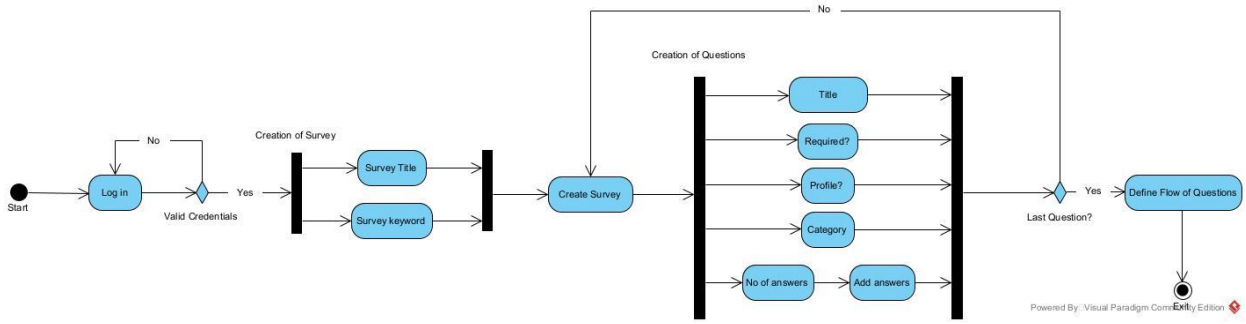
Admin - Sequence Diagram (CLI)



Admin - Sequence Diagram (frontend)



Create Survey (frontend)



(The CLI's activity diagram is trivial; it only requires uploading the JSON file through the command line)

2.2.7 Output Data

Produce Success or Failure Message

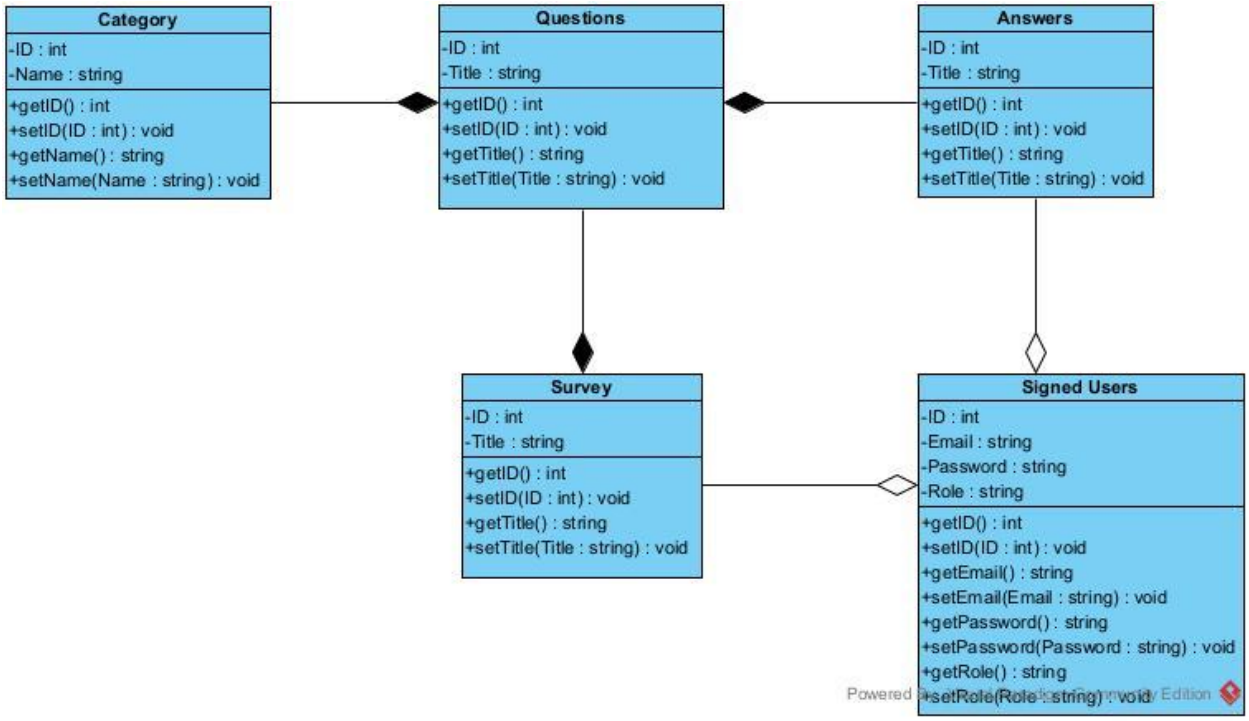
2.3 Performance Requirements

Our software will be available for use throughout each day and we expect for our software to be able to deal with users' requests (log in or data get/post) without significant delay.

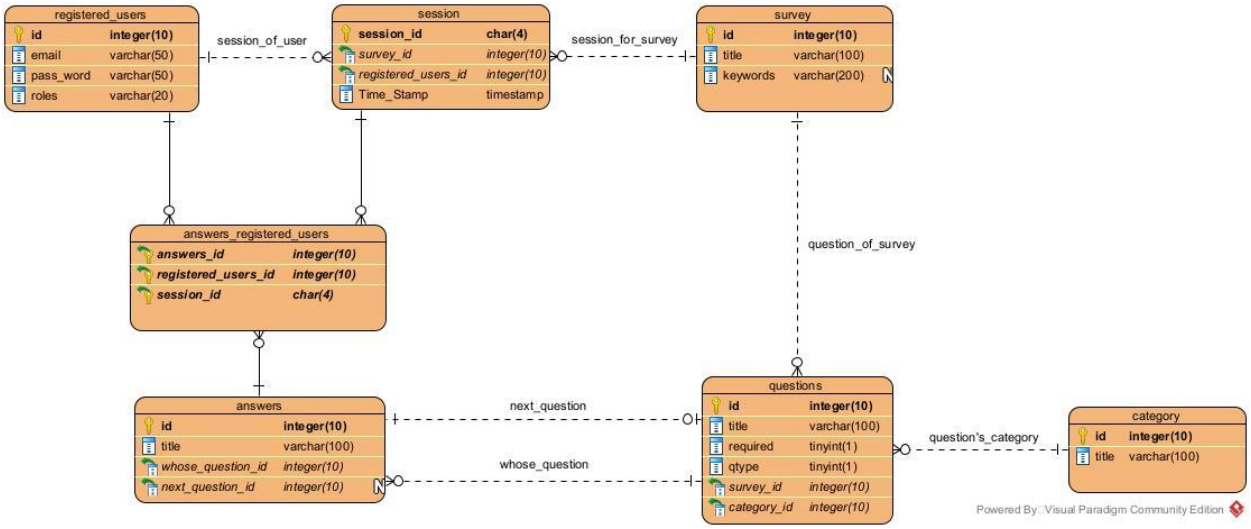
2.4 Data Organization Requirements

Below, we can see the Class Diagram of our System and the Entity Relationship Diagram of the Database.

Class Diagram



Entity Relationship Diagram



2.4.1 Data Access Limitations

Users have access to only specific parts of the Database, such as questionnaires, questions and answers and not personal information of other users. Only the System administrator is able to handle data concerning usernames and passwords.

2.5 Other Requirements

2.5.1 Software Availability

The software has to be available for use throughout the day.

2.5.2 Security

<We wish>

2.5.3 Maintenance

Maintenance of our system needs to be performed regularly and is subject to the number of users our system hosts.