**SOFTWARE ENGINEERING PROBLEM SPECIFICATION TABLE, IDENTIFYING THE FOLLOWING ITEMS**

|  |  |
| --- | --- |
| CLIENT | GreenSQA |
| USER | GreenSQA employees & Administrator |
| FUNCTIONAL REQUIREMENTS | RF0: Create a project.  RF1: End a project phase.  RF2: Register a capsule.  RF3: Approve a capsule.  RF4: Publish a capsule.  RF5: Inform the user how many capsules are registered based on their type.  RF6: Inform the user about the learnings of the capsules registered in the projects corresponding to a particular phase.  RF7: Inform the user of the project with the most capsules registered.  RF8: Inform the user if a partner has registered capsules at any time.  RF9: Inform the user of the situations and learning of the approved and published capsules. |
| CONTEXT OF THE PROBLEM | GreenSQA is a company focused on high-quality software assurance. There is a knowledge leak due to the frequent rotation and change of employees. To solve this problem the company plans to design software that allows employees to manage, consult and register information about the projects they are currently working on in capsules. These ones are texts where the employee describes situations, elements, or facts relevant to the project. This would allow the engineer’s work to be maintained. Every project taken by GreenSQA is divided into 6 phases and in each phase, the employees will write capsules in order to store their process. All the capsules should have words surrounded by hashtags since these will be important words that will have the possibility to be searched by other workers to gain knowledge about the project. |
| NON-FUNCTIONAL REQUIREMENTS | RF0: Since this is a test version, it will only count with the possibility of making 10 projects.  RF1: Since this is a test version, it will only be 50 capsules per phase.  RF2: The text inside the capsule must contain one or multiple words with hashtags. |

**Functional requirements analysis (Note: One table for each functional requirement)**

| Name or identifier | RF0: Create a project | | |
| --- | --- | --- | --- |
| Summary | When accepting a project from a client and creating it. The software should store:   * Name of the project * Name of the Client * Value corresponding to the budget of the project. * Names and phone numbers of the managers echarged for the project, from GreenSQA side, and from the client’s side.   Every time a project is created, its 6 phases must be created with it but only the start phase must be initialized.  Projects are divided into 6 phases: Start, analysis, designing, execution, closing and control, and maintaining the project. Each phase will have:   * A date of starting and ending (Planned) * A date of starting and ending (Current one) * Every time the capsule is approved, this information should be saved.   In order to know the planned dates and times, the software has to ask the user for how long each phase will last. | | |
| Inputs | Entry name | Data type | Selection or repetition condition |
| projectName | String |  |
| clientName | String |  |
| projectBudget | Double | It must be a positive number. |
| greenSQAManagersNames | String |  |
| clientManagersNames | String |  |
| startPlannedDate | Date | It has to be greater than today’s date |
|  |  |  |
| Result or postcondition |  | | |
| Outputs | Entry name | Data type | Selection or repetition condition |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |

| Name or identifier | RF1: End a project phase | | |
| --- | --- | --- | --- |
| Summary | In order to end a project phase and start the other one the system should register the approbation of the phase, and the current date of ending. Besides that, the phase changes from active to inactive and the next phase is started with the current date. | | |
| Inputs | Entry name | Data type | Selection or repetition condition |
| projectName | Project | It must be a project that is currently been working on. |
| Result or postcondition | The software will receive the project name which should change phases. It will save the approbation of the phase, and the current date of ending and it will switch the phase’s current state to Inactive and activate the next phase with the current date as a starting date. If the process was able to complete successfully, the software should display a message indicating so, on the other hand, display a message saying why it wasn’t possible to end the phase. | | |
| Outputs | Entry name | Data type | Selection or repetition condition |
| message | String | The user should see a message indicating whether the phase ended successfully or not. |

| Name or identifier |  | | |
| --- | --- | --- | --- |
| Summary |  | | |
| Inputs | Entry name | Data type | Selection or repetition condition |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
| Result or postcondition |  | | |
| Outputs | Entry name | Data type | Selection or repetition condition |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |

| Name or identifier |  | | |
| --- | --- | --- | --- |
| Summary |  | | |
| Inputs | Entry name | Data type | Selection or repetition condition |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
| Result or postcondition |  | | |
| Outputs | Entry name | Data type | Selection or repetition condition |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |

| Name or identifier |  | | |
| --- | --- | --- | --- |
| Summary |  | | |
| Inputs | Entry name | Data type | Selection or repetition condition |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
| Result or postcondition |  | | |
| Outputs | Entry name | Data type | Selection or repetition condition |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |

| Name or identifier |  | | |
| --- | --- | --- | --- |
| Summary |  | | |
| Inputs | Entry name | Data type | Selection or repetition condition |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
| Result or postcondition |  | | |
| Outputs | Entry name | Data type | Selection or repetition condition |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |

| Name or identifier |  | | |
| --- | --- | --- | --- |
| Summary |  | | |
| Inputs | Entry name | Data type | Selection or repetition condition |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
| Result or postcondition |  | | |
| Outputs | Entry name | Data type | Selection or repetition condition |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |

| Name or identifier |  | | |
| --- | --- | --- | --- |
| Summary |  | | |
| Inputs | Entry name | Data type | Selection or repetition condition |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
| Result or postcondition |  | | |
| Outputs | Entry name | Data type | Selection or repetition condition |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |

| Name or identifier |  | | |
| --- | --- | --- | --- |
| Summary |  | | |
| Inputs | Entry name | Data type | Selection or repetition condition |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
| Result or postcondition |  | | |
| Outputs | Entry name | Data type | Selection or repetition condition |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |

| Name or identifier |  | | |
| --- | --- | --- | --- |
| Summary |  | | |
| Inputs | Entry name | Data type | Selection or repetition condition |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
| Result or postcondition |  | | |
| Outputs | Entry name | Data type | Selection or repetition condition |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |