**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 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 the GreenSQA side, and 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 how long each phase will last. | | |
| Inputs | Entry name | Data type | Selection or repetition condition |
| projectName | String | REQUIRED |
| clientName | String | REQUIRED |
| projectBudget | Double | It must be a positive number. |
| ManagersNames | String | REQUIRED |
| durationInMonths | Array | It has to be an array with 6 positive integers. |
| ManagersPhoneNumber | String | It has to be a string containing only numbers and special characters |
| Result or postcondition | When the project is accepted, the software will ask the user for the name of the project, the name of the client, the budget of the project, and the names and phone numbers of the managers in the project whether they are from greenSQA or the client. The software then will set the actual date as the starting date and calculate the ending date based on how long the phases last. The software will validate the information and will output a message of confirmation. If any of the information inserted by the user is not valid, the software will show a message showing the error. | | |
| Outputs | Entry name | Data type | Selection or repetition condition |
| Message | String | It will be a message saying if the process was successful or not and why. |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |

| Name or identifier | RF1: End a project phase | | |
| --- | --- | --- | --- |
| Summary | 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 |
| indexOfTheProject | int | It must be a positive numbers corresponding to one of the projects in the system. |
| Result or postcondition | The software will display a menu showing all the available projects at the moment, after that, the software will receive the project index 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. |
| MenuOfProjects | String | It will be a menu with the index of the project and it’s name. |

| Name or identifier | RF2: Register a capsule | | |
| --- | --- | --- | --- |
| Summary | In every phase, the employees will generate capsules of knowledge. A capsule will have:   * ID * A description of the situation in which the employee wants to register. * A type of capsule (technique, management, domain, and experiences.) * Name of the employee * Position of the employee * Knowledge generated with that situation.   On each generated capsule, text must contain some words surrounded by hashtags indicating than that word is a keyword.  All the capsules are under revision, and it can be determined to approve them making them public. | | |
| Inputs | Entry name | Data type | Selection or repetition condition |
| description | String | It must contain hashtags. |
| kindOfCapsule | String | It will be one of the following values: (technique, management, domain, and experiences) |
| projectsName | String | It has to correspond to any of the projects already created. |
| nameEmployee | String | REQUIRED |
| knowledgeGenerated | String | It must contain hashtags. |
| Result or postcondition | The software will have the option to generate a capsule, each capsule must contain an ID (generated by the application), a description of the situation, a type of capsule (technique, management, domain, and experiences), the name, and position of the employee that is registering the capsule and the knowledge generated with the experience.  Besides this, the software has to make sure that the capsule has keywords (words surrounded by hashtags) and if all the requirements are fulfilled, it will display a message indicating so. If some of the information is wrong, the software will display a message indicating what information was missing. | | |
| Outputs | Entry name | Data type | Selection or repetition condition |
| Message | String | It will be a message saying if the process was successful or not and why. |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |

| Name or identifier | RF3: approved a capsule. | | |
| --- | --- | --- | --- |
| Summary | The software will ask for which capsule to approve, and it will register the date of approval. | | |
| Inputs | Entry name | Data type | Selection or repetition condition |
| IDofCapsule | String | It will be the ID of the capsule that wants to be approved, it must match the current capsules in the system. |
| Project | String | It will be a String representing which project the capsules that you want to approve are stored. The project must be active |
| Result or postcondition | The software will display a list indicating the projects that you can approve capsules from, and a list of what capsules are under revision, then the user can choose from one of the capsules and the application will automatically save the actual date of approval. Then it will show a message indicating the success of the transaction or not. | | |
| Outputs | Entry name | Data type | Selection or repetition condition |
| Message | String | It will be a message saying if the process was successful or not and why. |
| MenuOfCapsules | List | It will be a list containing all the capsules under revision |

| Name or identifier | RF4: Publish a capsule | | |
| --- | --- | --- | --- |
| Summary | The capsules that are interesting for the whole organization can be published on the web for everyone to see them. For this reason, the manager can select which of the already approved capsules he wants to publish. | | |
| Inputs | Entry name | Data type | Selection or repetition condition |
| IDofCapsule | String | It will be the ID of the capsule that wants to be published, it must match the currently approved capsules in the system. It has to be an approved capsule. |
| Project | String | It will be a String representing which project the capsules that you want to approve is stored. The project must be active |
| Result or postcondition | The software will display a list of possible capsules that were already approved to be published, it will receive the ID of the capsule that wants to be published and it will generate an URL link to the capsule. I the process was successful it will display a message indicating so. | | |
| Outputs | Entry name | Data type | Selection or repetition condition |
| URL | String | It will be an URL representing the capsule in an HTML file |
| Message | String | It will be a message saying if the process was successful or not and why. |
| MenuOfCapsules | List | It will be a list containing all the capsules that can be published. |

| Name or identifier | RF5: Inform the user how many capsules are registered based on their type. | | |
| --- | --- | --- | --- |
| Summary | The user will have the possibility of knowing how many capsules are registered based on their type (technique, management, domain, and experiences). To do this the user must input which type he wants to see. And the software will display all the capsules with that type | | |
| Inputs | Entry name | Data type | Selection or repetition condition |
| typeOfCapsule | String | It will be one of the following values: (technique, management, domain, and experiences) |
| Project | String | It will be a String containing the projects that the user can information from. |
| Result or postcondition | The software will receive a type of capsule and a project, then display all the capsules with that type registered to them. | | |
| Outputs | Entry name | Data type | Selection or repetition condition |
| MenuOfCapsules | List | It will be a list containing all the capsules of the same type. |

| Name or identifier | RF6: Inform the user about the learnings of the capsules registered in the projects corresponding to a particular phase. | | |
| --- | --- | --- | --- |
| Summary | The user will be able to see all the information on the capsules related to a phase. For this reason, the software will receive the project and then the phase. With this information, the software will show a list of the content of the capsules | | |
| Inputs | Entry name | Data type | Selection or repetition condition |
| projectName | String | It has to be a string that matches any of the current projects |
| phase | String | It has to be any of the 6 phases of each project |
| Result or postcondition | The software will first get the project name and the specific phase that he must display information about. After that, the software will display the content of all the capsules regarding a specific phase in a specific project. | | |
| Outputs | Entry name | Data type | Selection or repetition condition |
| learningsList | List | It will be a list containing all the learnings for a specific phase in a specific project. |

| Name or identifier | RF7: Inform the user of the project with the most capsules registered. | | |
| --- | --- | --- | --- |
| Summary | When the user wants to see the project with the most capsules registered the software will display it for him. It will be just an option in the menu. | | |
| Inputs | Entry name | Data type | Selection or repetition condition |
| N\A | N\A | N\A |
| Result or postcondition | When this option is selected on the menu, the software will go and look from all the projects which is the project that has the most amount of capsules registered without regarding each induvial phase. | | |
| Outputs | Entry name | Data type | Selection or repetition condition |
| projectName | String | It will be the project with the most amount of capsules registered. |
|  |  |  |

| Name or identifier | RF8: Inform the user of the project with the most capsules registered. | | |
| --- | --- | --- | --- |
| Summary | The software has to be able to receive the name of the partner or employee and display if it has made any capsules and when it did. | | |
| Inputs | Entry name | Data type | Selection or repetition condition |
| employeeName | String | It has to be an employee in the company database. |
| Result or postcondition | The software will receive the name of the employee and validate the information. If the name exists in the database it will display all the capsules that the employee has written. If not, it will display an error message. | | |
| Outputs | Entry name | Data type | Selection or repetition condition |
| Message | String | It will be a message saying that the name doesn’t exist in the database. |
| capsulesEmployee | List | It will be a list containing all the capsules that the employee has made. |

| Name or identifier | RF9: Inform the user of the situations and learning of the approved and published capsules. | | |
| --- | --- | --- | --- |
| Summary | The software has to be able to show the user the situations and learnings related to a capsule. This capsule will be searched by a text that the user inputs and then the software will display the information about the capsule | | |
| Inputs | Entry name | Data type | Selection or repetition condition |
| search | String | It has to be a word in any of the capsule’s learnings. |
| Result or postcondition | The user will input keywords or text and the software has to show the user all the information related to that text or keyword. If there is no match in the database the software will show an error message. The information must have been approved and published to be shown. | | |
| Outputs | Entry name | Data type | Selection or repetition condition |
| learning | String | It will be a list with all the learnings in capsules containing the search String. |
| errorMesagge | String | It will be a message indicating to the user that the information he was looking for does not exist in the database. |