Technological Institute of Costa Rica

Computer Engineering School

Inspection Report

Software Quality Assurance

I-2024

Professor: Erick Hernández Bonilla

Members:

María Paula Bolaños Apú

Marco Herrera González

Jeffrey Leiva Cascante

Content

Revision Log

Date	Version	Changes
April 18, 2024.	1.0.	Report produced by the Inspection Recorder.

Inspection Report

Introduction

The following report documents the inspection process, decisions, and results regarding the inspection of the source code of the project. This inspection corresponds to the execution of the code inspections indicated by the test model specification document, for test models #1 and #2.

Inspection Preparation

The Inspection Team was formed by Marco Herrera González, who served as the Inspection Leader and Recorder; María Paula Bolaños Apú, who served as an Inspector; Jeffrey Leiva Cascante, who served as an Inspector; and members of the Development Team, who served as Authors and Readers. Each member of the Inspection Team used one hour of time to prepare for the inspection meeting.

The product that was reviewed was the source code of the entire project. The source code was the only input for the inspection.

Inspection Objectives

The objectives defined for the inspection were:

- Verify that all functions in the code are mapped to a functional requirement, so that the source code contains only the necessary code to implement the requirements, and there are no additional features implemented.
- Evaluate the time complexity of the source code to identify potential areas that may affect performance.
- Report any anomalies, errors and recommendations contained in source code of the project.

Completion Criteria

The item under inspection will be considered as approved if there are no critical or catastrophic anomalies found. If there are critical or catastrophic anomalies, then the item will be rejected, and another inspection will have to be conducted on the fixed item. If there are negligible and marginal anomalies, the item will be partially approved until anomalies are fixed, when the item will be totally approved.

Inspection Meeting

The inspection meeting was performed on April 18, 2024, with a total duration of two and a half hour. All members of the Inspection Team were present. During the meeting, the entire document of the software requirements was inspected. Findings of the inspection are specified in the following sections.

Anomaly List

The anomalies found by the inspectors in the source code, discussed during the inspection meeting are:

- In the CalendarDAO.ts file, in the *overlaps*() function, the function is getting all events registered in the database to verify if any of those events overlaps with the event the function gets as a parameter. This is inefficient, as it is not necessary to bring all events of the database for comparison with the input event, and the code logic itself is unnecessary as the checking can be done with a single database query. This function represents a performance issue that can arise when the quantity of events registered is considerable and many events are getting registered. This is a marginal anomaly, of the risk-prone type.
- In the CategoryDAO.ts file, in the *getCategories*() function, the way of getting the subcategories for each function is inefficient. For each category, the function *getSubCategories*() (that makes a query to the database) is getting called to bring the subcategories of the category. This can make queries slow when the quantity of categories is considerable, considering that this function will have a high usage rate by the users. This is a marginal anomaly, of the risk-prone type.
- On the NotificationButton.js file, the useEffect() function is requesting the unread messages of the user every five seconds. This approach is inefficient, and instead, services like web sockets or APIs specialized for sending messages from the backend server should be used. On a real scenario, the number of requests sent to the server for the notifications would definitely slow down the system. This is a critical anomaly, of the risk-prone type.

Final Considerations

All anomalies found were discussed and reviewed during the meeting, all the Inspection Team members agree on the anomalies list presented on this document. The objectives of the inspection were accomplished, anomalies were reported and sent to the Development Team for rework and corrections, ensuring the quality of the product. Given the completion criteria, the item under inspection is not approved as there is a critical anomaly that should be fixed in order for approval. When anomaly is fixed, another inspection should be made on the fixed issue for approval.