

# Testing of the report management system

By:

Dembegiotis, Callista	s5822637
Nieuwenhuis, Demian	s5163609
Giummarra, Alessio	s5793181
Mura, Maria	s4778898
Willmann, Leo	s5253144

Faculty of Science and Engineering, University of Groningen  
WBCS017-10: Software engineering

June 2, 2024

<b>Introduction.....</b>	<b>3</b>
<b>Traceability matrix.....</b>	<b>3</b>
<b>Acceptance testing.....</b>	<b>5</b>
US-1 Management login.....	5
US-2 Report submission.....	6
US-3 View reports.....	6
US-4 Update status and priority.....	7
US-5 Manage questionnaire.....	7
US-6 Employee assignment.....	8
US-11 System environment.....	8
US-12 Deployment instructions.....	9
US-13 Maintenance.....	9
<b>Progress.....</b>	<b>10</b>

# Introduction

This document is one of three parts that will discuss the report management system developed for Gomibo. This document will discuss the testing of the application.

## Traceability matrix

The traceability matrix will connect every test to a requirement. Each test will be given a unique test ID and can be found in the test files and commit log. The test files are stored in the `backend/tests/Feature/` package and the matrix shows which files are tested.

The test ID tag is composed as follow:

{test type}-{test content tag}{test number}

Where

- {test type} can be UT for unit test, FT for feature test and IT for integration test. Though for our use case, we are testing the API endpoints and the controllers, which by laravel convention are feature tests.
- {test content tag} is an abbreviation for the test content. Examples are AUTH for authentication, QE for questions endpoint, RE for report endpoint, AE for answers endpoint and FE for file endpoint.
- {test number} is the nth test of the test file.

Requirement	Package	Files tested	Test file	Test ID	Passed?
<b>US1-M1:</b> There is a login and authorization system for the role of 'report manager' whenever a user logs into the web service.	backend/app/Http/Controllers	AuthController.php	AuthenticationTest.php	FT-AUTH1 FT-AUTH2 FT-AUTH3 FT-AUTH4	Yes
<b>US2-M1:</b> The API must be able to retrieve the questionnaire from the database.	backend/app/Http/Controllers	QuestionsController.php	QuestionsTest.php	FT-QE5 FT-QE6 FT-QE7 FT-QE8	YES
<b>US2-M2:</b> The API must be able to create the report with responses in the database.	backend/app/Http/Controllers	ReportsController.php	ReportsTest.php	FT-RE5 FT-RE6 FT-RE7 FT-RE8 FT-RE9	YES
<b>US2-M4:</b> The API must be able to create answers in the database to answer open	backend/app/Http/Controllers	AnswersController.php	AnswersTest.php	FT-AE1 FT-AE2	YES

questions.					
<b>US2-S6:</b> The API should store uploaded files and create file entries in the database.	backend/app/Http/Controllers	FileController.php	FileTest.php	FT-FE1 FT-FE2 FT-FE3 FT-FE4 FT-FE5	YES
<b>US3-M2:</b> The API must be able to retrieve a single report with all its responses from the database.	backend/app/Http/Controllers	ReportsController.php	ReportsTest.php	FT-RE1 FT-RE2 FT-RE10	YES
<b>US3-M3:</b> The API must be able to retrieve all report entries with search filters such as 'by priority' or 'by status' from the database through the backend.	backend/app/Http/Controllers	ReportsController.php	ReportsTest.php	FT-RE3 FT-RE4 FT-RE13 FT-RE14	YES
<b>US3-M4:</b> Only a logged in 'report manager' must be able to retrieve reports through the API.	backend/app/Http/Controllers	ReportsController.php	ReportsTest.php	FT-RE15 FT-RE16	YES
<b>US3-S6:</b> The API should be able to download file entries from the stored file path in the database.	backend/app/Http/Controllers	FileController.php	FileTest.php	FT-FE6 FT-FE7	YES
<b>US4-M1:</b> The API must be able to update report entries' status and priority in the database.	backend/app/Http/Controllers	ReportsController.php	ReportsTest.php	FT-QE13 FT-QE14	YES
<b>US4-M2:</b> Only a logged in 'report manager' must be able to update report entries' status and priority through the API.	backend/app/Http/Controllers	ReportsController.php	ReportsTest.php	FT-RE17	Yes
<b>US5-M2:</b> The API must be able to create questions with its belonging multiple choice options in the database.	backend/app/Http/Controllers	QuestionsController.php	QuestionsTest.php	FT-QE1 FT-QE2 FT-QE3 FT-QE4	YES

<b>US5-M5:</b> The API must be able to update the <code>is_active</code> field of questions in the database.	backend/app/Http/Controllers	QuestionsController.php	QuestionsTest.php	FT-QE9 FT-QE10	YES
<b>US5-M6:</b> Only a logged in 'report manager' must be able to create and update questions through the API.	backend/app/Http/Controllers	QuestionsController.php	QuestionsTest.php	FT-QE11 FT-QE12	Yes
<b>US6-S2:</b> The API must be able to update report entries in the database to assign 'report manager' to the report.	backend/app/Http/Controllers	ReportsController.php	ReportsTest.php	FT-RE12	Yes

The coverage score of our tests over the controller, resource and requests package yields 93.4% with the breakdown below. This score is gained by executing the `php artisan test --coverage` command.

```

Controllers/AnswersController ..... 100.0%
Controllers/AuthController ..... 100.0%
Controllers/Controller ..... 100.0%
Controllers/FileController ..... 59 / 96.0%
Controllers/QuestionsController ..... 100.0%
Controllers/ReportsController ... 41..42, 64, 70, 72, 124, 127, 131, 178..181 / 86.9%
Requests/LoginRequest ..... 100.0%
Resources/UserResource ..... 100.0%

Total: 93.4 %

```

## Acceptance testing

This section discusses the client response and remarks to our project. Testing is done per use case of the User Stories.

Note: The acceptance testing was done before the file upload and download functionality was implemented. Thus the client could not verify the functionality and are thus not included in the section below.

### US-1 Management login

**As a** maintenance personnel, **I would like** to be able to login to the report management system, **so that** only maintenance personnel can manage malfunction reports.

Persona	Maintenance personnel
---------	-----------------------

Task	Login to the system with correct credentials
Worked as intended?	Yes
Remarks	Easy way to log in. With this solution you can also keep information familiar, when necessary.

Acceptance criteria:

- Given: a maintenance personnel tries to log into the system.
- When: they enter their credentials.
- Then: the system checks their credentials and gives them the authorization of a 'report manager'.

## US-2 Report submission

**As a** reporter, **I would like** to report a malfunction, **so that** it gets fixed as soon as possible.

For use case submit report:

Persona	Reporter
Task	The reporter will use the web application to submit a report.
Worked as intended?	Yes
Remarks	Mobile view looks good

Acceptance criteria:

- Given: a reporter is on the reporting page
- When: the reporter fills in the current questionnaire, and submits it.
- Then: through the API, a report submission is stored in the database.

## US-3 View reports

**As a** maintenance personnel, **I would like** to be able to view the submitted reports, **so that** I know what equipment needs maintenance.

Persona	Maintenance personnel
Task	Go to the dashboard and view the list of unresolved reports.
Worked as intended?	Yes

Remarks	The overview looks good. The information is clearly processed in the overview.
---------	--------------------------------------------------------------------------------

Acceptance criteria:

- Given: a report manager is logged into the system.
- When: the user opens the corresponding page to view reports.
- Then: the user sees a list of the most recent unresolved reports (at default), further filterable with additional settings.

## US-4 Update status and priority

**As a** maintenance personnel, **I would like** to set the priority, and status of the submitted reports, **so that** fellow maintenance personnel know the priority and status of the report.

Persona	Maintenance personnel
Task	On the dashboard page, click on a report to view and edit it.
Worked as intended?	Yes
Remarks	It works and you can choose how many priorities you prefer.

Acceptance criteria:

- Given: a logged in maintenance personnel and a report.
- When: the user opens the corresponding page to edit the report status / priority.
- Then: through the API, the new status / priority is stored in the database.

## US-5 Manage questionnaire

**As a** maintenance personnel, **I would like** to manage the questionnaire asked in a report submission, **so that** any change in the workspace, equipment or errors in the questionnaire can be addressed.

Persona	Maintenance personnel
Task	Initialize the questionnaire with a set of questions both multiple choice and open questions.
Worked as intended?	Yes
Remarks	You can easily add (in-depth) questions as personnel which is a valuable option. This makes it easy to specify.

Acceptance criteria:

- Given: a report manager is logged into the system and on the corresponding page to edit the questions.
- When: the user edits the questions and submits them.
- Then: through the API, the list of questions is created and the old questions are updated to `is_active = false` in the database.

## US-6 Employee assignment

**As a** maintenance personnel, **I would like** to be able to assign myself and others to reports, **so that** fellow maintenance personnel know who is attending to it.

Persona	Maintenance personnel
Task	Assign the maintainer to a submitted report on the dashboard page.
Worked as intended?	Yes
Remarks	You can assign malfunctions to persons.

Acceptance criteria:

- Given: a report manager and a report.
- When: a report manager assigns a report manager to a report.
- Then: through an API call it is saved in the database.

## US-11 System environment

**As a** developer, **I would like** to have the system be developed in familiar environments, **so that** any eventual system maintenance or changes do not require the learning of new frameworks or languages.

Persona	Developer
Task	The developer checks if the software is using the Laravel, React and the MySQL database.
Worked as intended?	Yes
Remarks	

Acceptance criteria:

- Given: The system follows requirements of US12.



- When: a developer wants to do anything with the software.
- Then: the developer can improve productivity as they do not need to learn a new framework or language.

## US-12 Deployment instructions

**As a developer, I would like** to have an instruction set, **so that** it is easier to deploy the system.

Persona	Developer
Task	The developer will download the application and follow the instructed steps in the <b>README .md</b> file at the root of the repository.
Worked as intended?	Yes
Remarks	

Acceptance criteria:

- Given: there exists a **README .md** file at the root directory of the repository.
- When: a developer follows the instruction to deploy the system.
- Then: the developer has the system deployed.

## US-13 Maintenance

**As a developer, I would like** to receive a system that has high code quality, **so that** it is easier to extend and maintain the software.

Persona	Developer
Task	The developer will have a quick look through the repository to see if it is well structured and will check on the test coverage of the application.
Worked as intended?	Yes
Remarks	

Acceptance criteria:

- Given: the software has high code quality and is structured.
- When: a developer wants to extend and maintain the software.
- Then: the developer has an easy time implementing their changes.

## Progress

Who	When (2024)	Where	What
Leo	13 May	Traceability matrix; Acceptance testing	Initial table templates
Leo	15 May	Acceptance testing	Filled in acceptance testing table.
Leo	20 May	Acceptance testing	Reword acceptance testing layout.
Leo	24 May	Traceability matrix	Increased table size and included requirements with ID. Populated the table with testable requirements.
Leo	24 May	Acceptance testing	Reworked acceptance testing layout and added all User stories.
Leo	26 May	Traceability matrix	Added all API functionality testing
Leo	29 May	Acceptance testing	Added in client responses and remarks
Leo	2 June	All	Some text to the sections and the file testing.