Functional Requirements Document

for the Gym Manager Website

1. General

**1.1. Project Description and Purpose**

The Gym Manager project will help coaches to have a better overview of their customers and their specific needs. The purpose of this project is to facilitate personal trainers with an intuitive, easy-to-use platform for managing their customers.

**1.2**  **Additional Tooling used**

About Cypress:

Is an automation web testing tool that's fast, easy, and reliable for testing the web components that run in a browser. It's an open-source test automation framework for testing JavaScript web and enables you to perform unit, integration, and end-to-end tests.

2. Page Elements:

**2.1. Head**

The site title will update to show what page the user is on, but will also always show the name of the website;

The site will also have a favicon.

**2.2. Navigation**

The navigation section will display all the possible navigation options within the application. This will include:

* Language Selector (RO/EN)
* Client List
* Appointment List
* Display the coach's name on the upper-right side of the page.
* Next to the name, there should be a log-out button.

**2.3. Body**

The body will be dynamically updated based on the page that is currently displayed; It is the only part of the webpage that will change when a different page is loaded.

**2.4. Footer**

The Footer will contain a copyright notice containing the name of the company and website, along with the year of creation - current year.

3. Website Pages and specifications

**3.1. Login Page**

The Login Page is the first page users will see when accessing the application. This is where the user will enter a username and password. The login form will be displayed in the **center** of the page. After logging in the user will be redirected to the Client List page.

Missing login buton

1. Username;

* Text field;
* Maximum characters: 32;
* Minimum characters: 2;

2. Password;

* Password field;
* Maximum characters: 32;
* Minimum characters: 5;

**3.2. Client List Page**

Upon a successful login, the coach/user can access his personal Client List page. This is where he can register and manage his gym users, and also schedule appointments/training sessions.

On the left side of the page, there will be a simple form for adding a new client which adds a new record in the database based on the inputted data and a list of clients. The list will display the client name. Upon selection of a client (left-click on a client from the list) the rest of the body content will be filled with gym profile of the client.

The add gym client form consists of:

1. Gym Client name.

* Input type: text;
* Max Characters 64;
* Min Characters 2;

2. Age.

* Input type: integer;

3. Height.

* Depending on the language (RO/EN) selection, this will display either one field (RO) and next to the field “CM” text - or two fields (EN) one for the feet, next to the field text “feet”, and one for the inches, next to the field text “inches”. For the EN selection, the height data will be parsed as - feet value + ‘ + inches value.

4. Weight.

* Depending on the language (RO/EN) selection, this will display next to the input field either “KG” (RO) or LBS “(EN)

5. Gender.

* Option list picker which will include: M/F/X as choices.

After filling in the details, if the user submits the inputted data by using the “Add client” button, the page will refresh and the new client will be added to the list. If the user tries to add a client without filling in one of the field, a popup telling the user that “all fields are mandatory to be filled” will appear.

The Gym Profile of a client contains.

1. Gym Client name.

* Centered (relative to the gym profile area) and displayed in bold.

2. Profile details table with two columns.

* It will display general information: age, height, weight, gender/sex
* The table will be organized in two columns. First column will be consisted of the “age”, “height”(RO/EN), “weight”(RO/EN) and “gender”. The second column will display the respective personal values of the gym client.

3. Schedule appointment.

* A date picker for which has a start date of today. An hour picker with half an hour intervals that has a hour start of the next hour.
* A list picker for the workout type (cardio, muscle, endurance).
* A button that upon clicking on it schedules an appointment based on the selected date, hour, workout type and gym profile.
* After pressing the button, a pop-up will appear to confirm the success or failure registering the appointment in the database.

**3.3 Appointment List Page**

This page will display a table with all the scheduled appointments made by the coach. The table will show the following appointment column details::

1. Date

* Displayed in the correct format based on the language selection MM/DD/YYYY - EN, DD/MM/YYYY - RO.

2. Hour

* Displayed in the correct format based on the language selection 12 Hour AM/PM - EN, 24 Hour - RO.

3. Gym Client Full Name

4. Workout type

If the coach does not have any appointments made, the page will display a “no appointments schedules” message inside the table.

4. Security

**4.1 Page Restrictions**

Only the login page will be open to anyone; In order to access any of the other pages, the user has to be logged in;

**4.2. New accounts and SQL injection prevention**

New account cannot be created by a user. Only a database administrator can do that.

Any form data will be sanitized before being sent to the server., this means that SQL injections will not represent a threat.

5.0 Application testing plan

The following will detail an application testing plan for the Gym Manage web application, including scope, criteria, and members included in the plan.

\*Aditional Tooling used: Cypress

**5.1 Testing scope**

The scope of the tests should cover basic functionality, with both manual and automated tests. Basic functionality scope includes the following:

* Login
* Register
* Add Client
* Add Appointment
* Internationalization (abbreviated as “i18n”)

These tests are necessary for ensuring a good user experience. The functionality tested here is the bare minimum necessary for the application. If testing time does not allow for implementation of automated test regarding a functionality, that functionality should **AT LEAST** be tested by the manual testing method.

**5.2 Testing criteria**

If any of the basic functionality tests fail, the application state will be considered as critical. The test result should be consulted and the application debugged immediately. Deployment of the application should not happen under the failure of any of the basic functionality tests, or any other class suite of tests. Other tests that are added later and not part of the basic functionality suite, should be treated as failed simple failed tests, and be fixed in due normal time of development with a normal priority (no need for calling outside of work hours).

**5.3 Team testing responsibility resources**

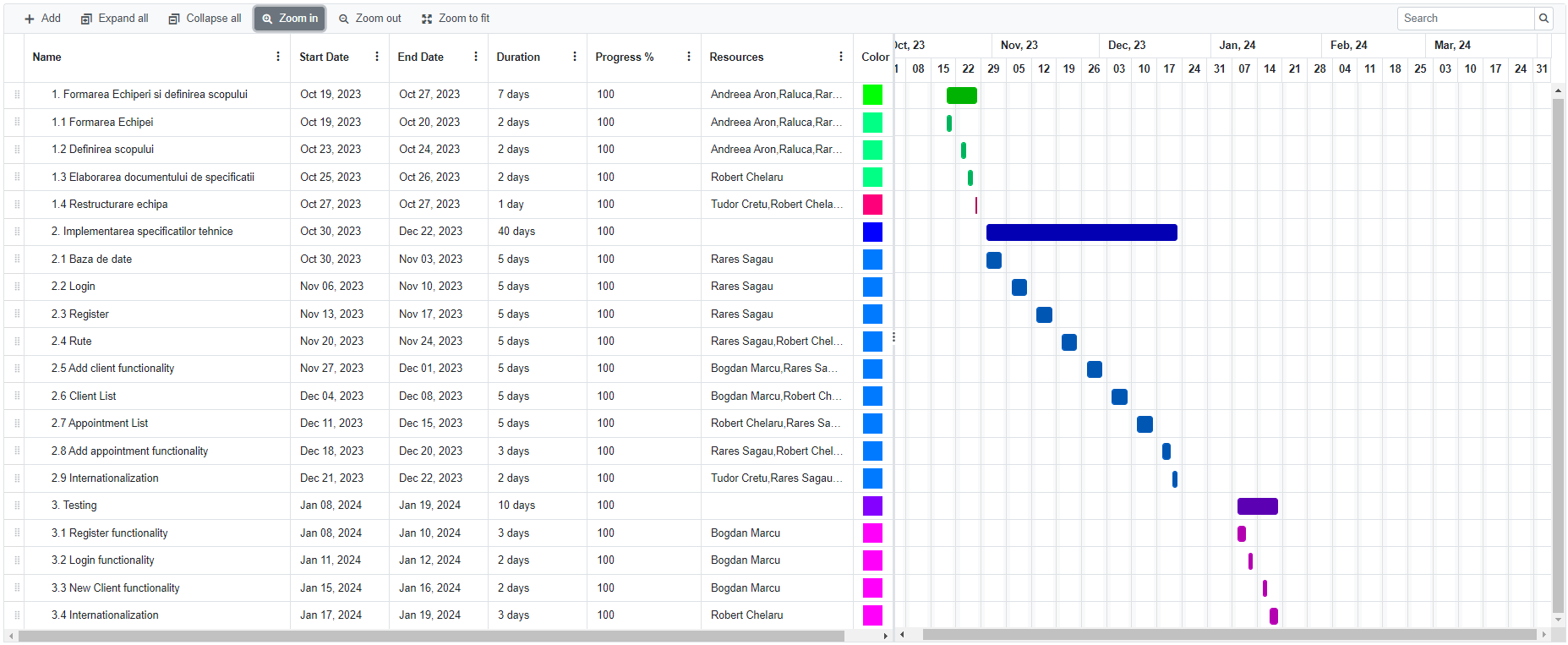
Sagau Rares - Setup testing application expert

Bogdan Marcu - Automated Testing and Manual Testing

Robert Chelaru - Automated Testing and Test case scenario

Tudor Cretu - Manual Testing

6.0 Gant planification, team members responsability and roles



Rares Sagau - Tech Lead and Product Owner, testing and setup

Bogdan Marcu - Consultant in development, testing and deployment

Robert Chelaru - Team Lead, developer, testing

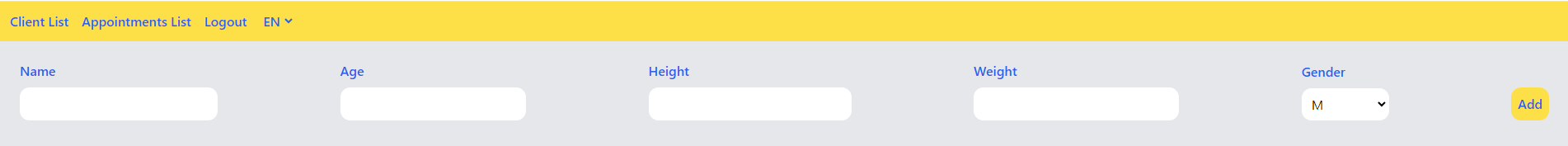
Tudor Cretu - Design and Manual Testing

7.0 Test Case example scenario

The following will detail and example Manual Test case that will serve as example but also as a first Manual Test Case that will be executed before deployment.

**7.1 Add Appointment**

Ensure that the “*Add Appointmen*t” functionality works by performing a complete add appointment happy-flow.  
  
The manual test was performed by following the next steps:

1. Authentication of the trainer inside our web application using the right credentials.
2. Make sure that you do NOT have an empty client list in the dashboard.
   1. *\*Optional: if the client list is empty you can create one by filling in the new client form displayed at the top of the dashboard.*
3. 
4. Select a desired client for an up coming appointment (expected behavior: the web-app should load a client details component and a form)
5. Send an empty form (expected behavior: application error)
6. Fill in the required form data:
   1. Date: both and future and a past date (expected behavior: for the future appointment, an redirect to the appointment-list URL should be made, for the past one and error page should be loaded)
   2. Hour
   3. Workout Type
7. In the *“Appointment-List”*  section you should expect the newly added appointment.

**7.3 Automated Tests scenarios**

|  |  |  |
| --- | --- | --- |
| Automated Tests scenarios | | |
| Test Case Number | Description | Expected Result |
| 1 | Verify the login button displays 'Autentificare' (RO) when the language is set to Romanian. | Login button displays 'Autentificare' |
| 2 | Check that changing the language to English (EN) on the login page changes the login button text to 'Sign In.' | Login button text changes to 'Sign In' |
| 3 | Confirm that user heights are displayed as '190cm' in the default language (RO) and '6 feet 3 inches' in English (EN). | User height displayed correctly in both languages |
| 4 | Verify that user weights are displayed as '73 kg' in the default language (RO) and '160 lbs' in English (EN). | User weight displayed correctly in both languages |
| 5 | Test the creation of a new user 'client4' with specific attributes (age, height, weight, gender) and verify successful creation. | New user 'client4' is created and verified. |
| 6 | Test login functionality with correct credentials (username: bogdan, password: 12345678) and verify redirection to '/client-list'. | Successful login and redirection to '/client-list'. |
| 7 | Test login functionality with incorrect credentials (username: lmao, password: parola) and verify redirection back to '/login'. | Failed login and redirection back to '/login'. |
| 8 | Test user registration with the button labeled either 'Înregistrare' or 'Sign Up', using new credentials (username: newUser2, password: 12345678), and verify redirection to '/client-list'. | New user 'newUser2' is registered and redirected to '/client-list'. |
|  | | |

Github-repo: https://github.com/Reynk/gym-manager