Angular Learning & Development Path

Angular Project

Bug Reporting System

October 2022

1 Introduction

1.1 Prerequisites

Initialize a new Angular project following the standard practices when it comes to what has been taught in class. Proceed in developing the project by following the order of the requirements steps as they are depicted in the following paragraphs.

Next, create a personal GitHub account or use your existing one. Create a new public repository to push the project. Send the link of your repos to idaniil@athtech.gr.

Study the file *db.json* provided to understand the structure of the data stored (json format).

Note: For each step, you are free to implement the requirements in any way you want. However, keep in mind that there are good implementations and bad implementations. So, make sure that your solution follows the best practices for matters such as code quality and others.

1.2 Technology Stack

The technology stack you have at your disposal consists of the following components:

- Git
- Angular

The development IDE of our choice is Visual Studio Code. Nevertheless, if you consider using another IDE will help, you are free to use that IDE.

You are free to use any library for the development of the project.

1.3 Backend Server

You have to install (if not already installed) the *json-server* npm module to simulate your backend. Copy the *db.json* file provide in the appropriate place and start the server:

```
json-server --watch db.json
```

For more information for the server and the way it exposes the API check: JSON Server (Github)

2 Functional Requirements

2.1 Product Perspective

All systems, regardless of their size, have errors that cause them to not function correctly. Those errors are called bugs and some of them are very critical and require immediate attention while others are minor and require little attention.

The bugs are resolved by the DEV (development) team and are often reported by the QA (Quality Assurance) team who is responsible for the overall quality of the product.

To report the bugs, we need to have a system that both the DEVs, QAs, and even POs (product owners) can access.

The goal of the project is to create a Bug Reporting System that will be used by QAs, DEVs, and POs for the submission and review of bugs.

If you need more fields/properties add them

The product needs to be a Single Page Application (SPA). More specifically, it will be an Angular Application that will communicate via HTTP with a backend API.

For quick reference, here is a list of all the available endpoints:

Domain	https://localhost:3000/
GET	/bugs
POST	/bugs
PUT	/bugs/:id
DELETE	/bugs/:id
GET by Id	/bugs/:id
Get sorting - pagination	/bugs?_sort=title,desc&page=0&size=10&title=bug&priority=1&reporter=QA&status=Done

2.2 Functionality

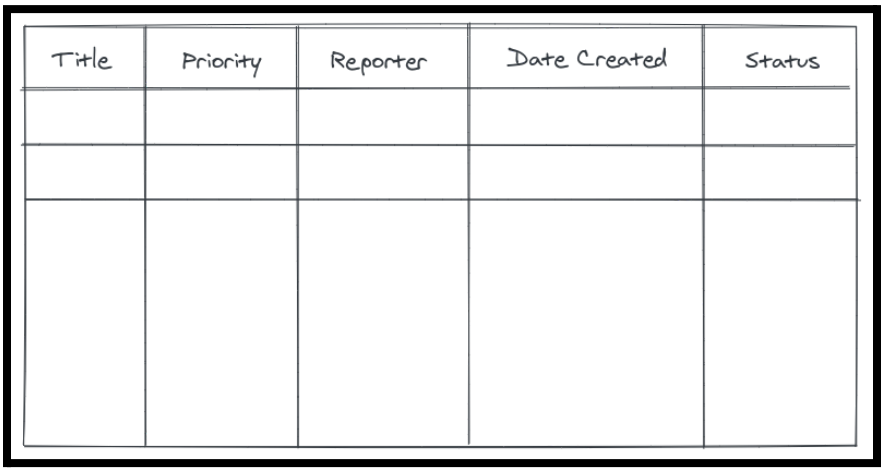
2.2.1 Description

The project is required to have the following user stories implemented:

1. As a QA user, I want to see a list of bugs, so that I can identify what has been solved, what has been created and what is the priority.

Acceptance Criteria:

- Display on an HTML table the following fields:
    - Title,
    - Priority,
    - Reporter,
    - Date Created,
    - Status.
- The table should be sortable by all of its fields in a bi-directional way. For example, when clicking on the Priority header, the data should be sorted based on higher priority. If clicked again, the data should be sorted based on lower priority.

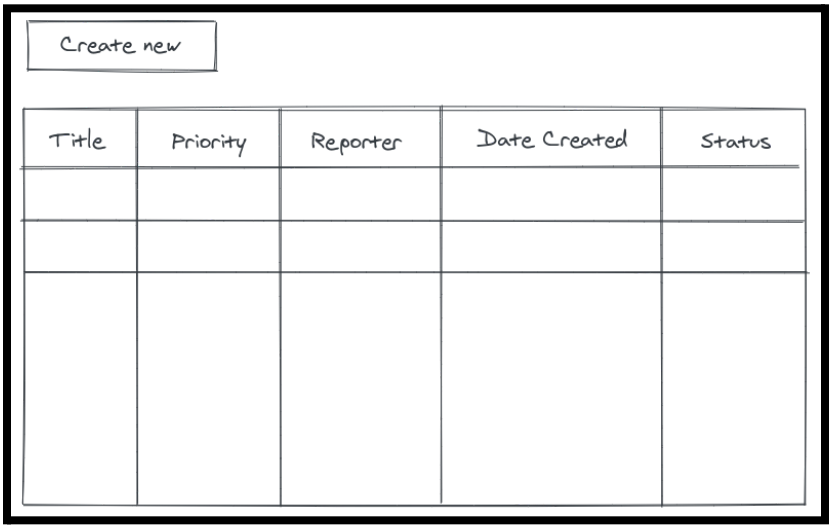


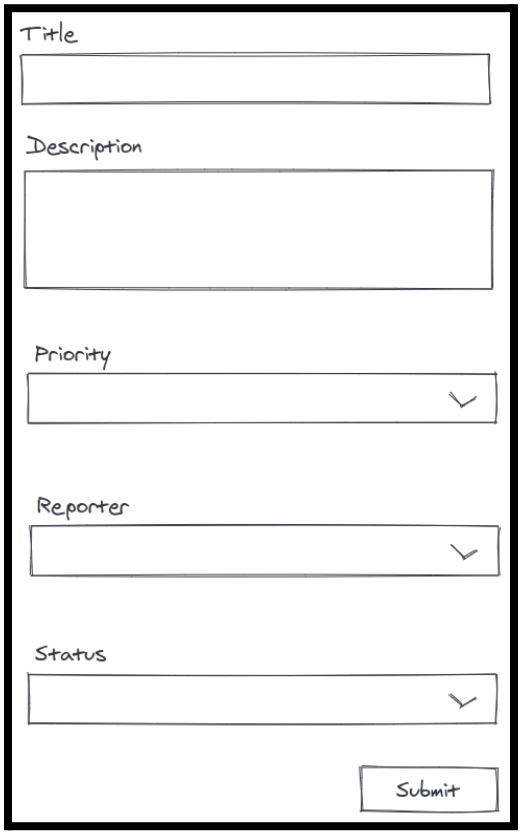
2. As a user, I want to report a bug, so that it can be solved by the development team.

Acceptance Criteria

- Place a button on top of the list of bugs table, which redirects us to a new page.
- The new page should have the following fields:
    - Title (title of bug),
    - Description (explanation of bug),
    - Priority (Minor, Major, Critical),
    - Reporter(QA, PO, DEV),
    - Status(For Review, Done, Rejected).

  The following should apply**:**
    - If the reporter is QA, then the status by default becomes mandatory.
    - All fields are required, except for status which is conditional (it can be empty/null).
- When the bug is created, it should be visible in the list of bugs table.



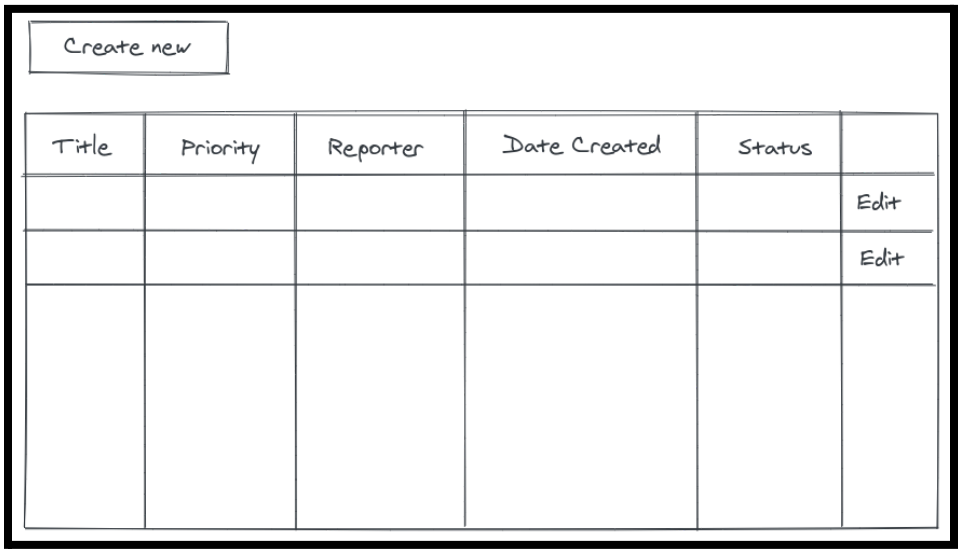


3. As a QA user, I want to edit a bug, so that I can update all its fields.

Acceptance Criteria

- Each item (bug) in the list of bugs table should have an edit button.

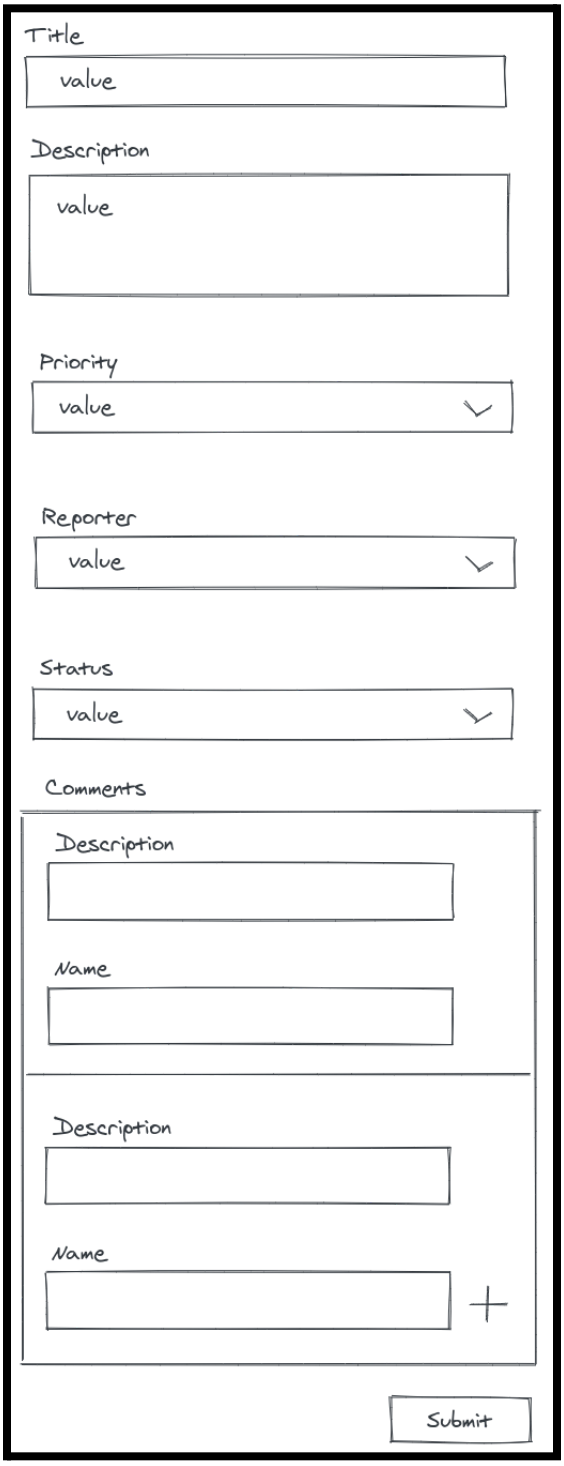
- When clicked, the user should be redirected to a new page where he/she can edit the bug. All of the fields are expected to be filled with the existing values.
- When the user finishes editing the bug and saves it, the updated item should be visible in the list of bugs table.



4. As a PO/DEV user, I want to add comments to a bug, so that I can express my thoughts to the QA team.

Acceptance Criteria

- On the edit page of a bug, a new section should exist with the following:
  - A comment list,
  - The following fields which can be filled:
    - Comment (free text),
    - Reporter Name.
- When the user clicks on the submit button, any new comment should be added to the comment list.

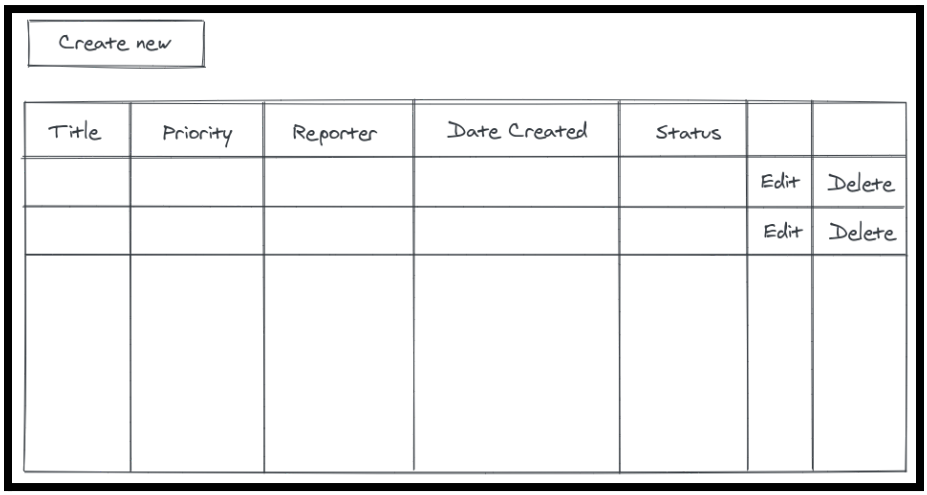


- At the bottom of the list, add two buttons (left arrow & right arrow) which increases or decreases the pages.

5. As a user, I want to delete bugs.

Acceptance Criteria

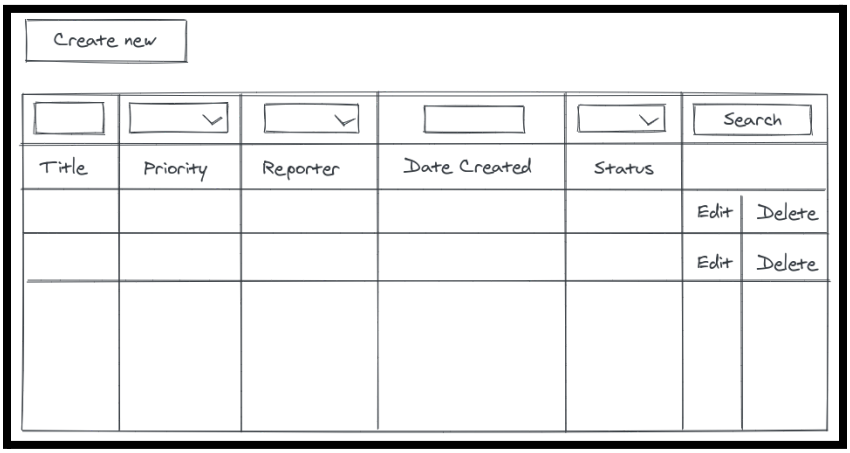
- Each item (bug) in the list of bugs table should have a delete button.
- When clicked, the bug should be deleted from the list of bugs table.



6. As a QA user, I want to be able to search for bugs, so that I can find faster what I'm looking for.

Acceptance Criteria

- The list of bugs table should have a new row above its headers which is used for searching. The searching fields are the following:
    - Title,
    - Priority,
    - Reporter,
    - Status.
- A "search" button should exist at the end of this new row.
- When the user clicks the button, the filtered results are fetched from the server.



3 Deliverables

3.1 Delivery Date

The project's delivery date is **10/11/2022**.

3.2 Source Code Delivery

The Github repository hosting your source code is the actual delivery. Although technically, you can push the entire source code at once, we urge you to have multiple commits as this approach will allow us to track your thinking and the approach you followed to develop the final product.

Once ready, you should send an email to:

- i.daniil@codehub.gr

containing:

1. Your full name,
2. GitHub repository URL.