PROJECT ESTIMATION

User-Story: As a developer,

I want to see a new chart on my dashboard (the page already exists). This chart will show the

amount of work (SP) assigned to me by status.

Design:

Chart, pie chart

Description automatically generated

Context: In our current codebase, there exists an entity called "Issue". Each Issue has the following values:

● Id (UUID) ● Title (text) ● Description (text) ● Story points (number)

● Developer (relation with user entity) ● Status (non-nullable enum with one of the following states: To Do, In Progress, Under Review, Done)

The chart will show the total number of story points of the issues grouped by status and where

the developer is the logged user.

SCOPE

Definitions : 1: 1-2 hrs, 2:3-4hrs, 3: 5-10hrs.

**Front end**

*Design components(2)* – design components that will be used in the dashboard.

*State management(2)* – decide where state for the items in the dashboard should be managed.

*Develop JS logic (3)* – implement state management logic & fetch calls to retrieve developer information and store in our app, passing data to our components.

*Folder Architecture(1)* – decide on folder architecture for readability and structure.

*Style Components(2)* – create css for components.

**Back end**

*Routing(1)* – create a route to receive a custom parameter to identify developer.

*Data Fetching(2)* – implement data fetching logic to Issue entity, returning all issues where the assigned developer matches our developer route param.

*Error Handling(2)* – implement error handling logic for routes .

*Utils & Functions(3)* – create our utils and functions to handle calculating the percentage of each task status.

*Folder architecture(1)* – decide on folder architecture for readability and structure.

**Tests**

*Write performance tests for entire stack(3)* – write tests (including regression) on our entire stack.

*Run performance tests on stack/utils (1)* – run our tests on our entire stack.