*Florida International University*

*School of Computing and Information Sciences*

Software Engineering Focus

Feature Document

User Story #**271**

**Team Member:**

Gabriel Fernandez

**Product Owner(s)**:

Mohsen Taheri

**Mentor(s)**:

Mohsen Taheri

**Instructor**: Masoud Sadjadi

**User Story** Implement new progress algorithm

* As a front-end developer, I want to receive the smart progress status for all of the logged in user’s assignments, so that the user knows what assignments need work.
* **Acceptance Criteria**
  + There exist a backend endpoint that returns the smart status of the logged in user’s assignment
  + The endpoints return extra relevant data so that the front end doesn’t have to send multiple requests
  + The collaboration endpoint is updated with the new algorithm
  + The algorithm is implemented according to the business rules
  + Assignments should now have weights

**Use Case** #**001 – Fetch Assignment Smart Status**

**Actors**

Front-end developer

**Entry Conditions**

Front-end developer has access to a terminal or an api to send http requests

Front-end developer has read the documentation and understands which api endpoints to hit and what data to send

**Flow of Events**

1. Use case starts when front-end developer sends an http request to the assignment progress endpoint with the assignment’s primary key
2. The backend authenticates the user
3. The backend queries the database for assignment & assignment roster data
4. The backend queries the database for all of the user’s task in the given assignment
5. The backend calculates the smart status based on the tasks’ weights, the time left until the assignment due date, and the business rules
6. The backend returns a response with the smart status to the front-end developer and the use case ends

**Alternate Flow of Events**

* 2a.
  + The backend determines the token does not belong to an existing user and the use case ends.
* 3a.
  + The backend determines the user is not enrolled in the assignment or that the assignment does not exist, and returns a response with the error and the use case ends.

**Use Case Diagram**



**Sequence Diagram**



**Class Diagram**



**Unit Test**

**Test Case 1 (Sunny Day)**

**Purpose**

* Ensure backend returns smart status for the given user and assignment

**Precondition**

* Http request is made to the server at the assignment progress endpoint

**Input**

* User token, assignment pk

**Expected Result**

* Smart status for the user at the given assignment & extra information about the algorithm

**Actual Result**

* Smart status for the user at the given assignment & extra information about the algorithm

**Test Case 2 (Rainy Day)**

**Purpose**

* Ensure that the server sends a proper error message when the smart status isn’t available

**Precondition**

* Http request is made to the server at the assignment progress endpoint

**Input**

* User token, assignment pk for which the user doesn’t belong to

**Expected Result**

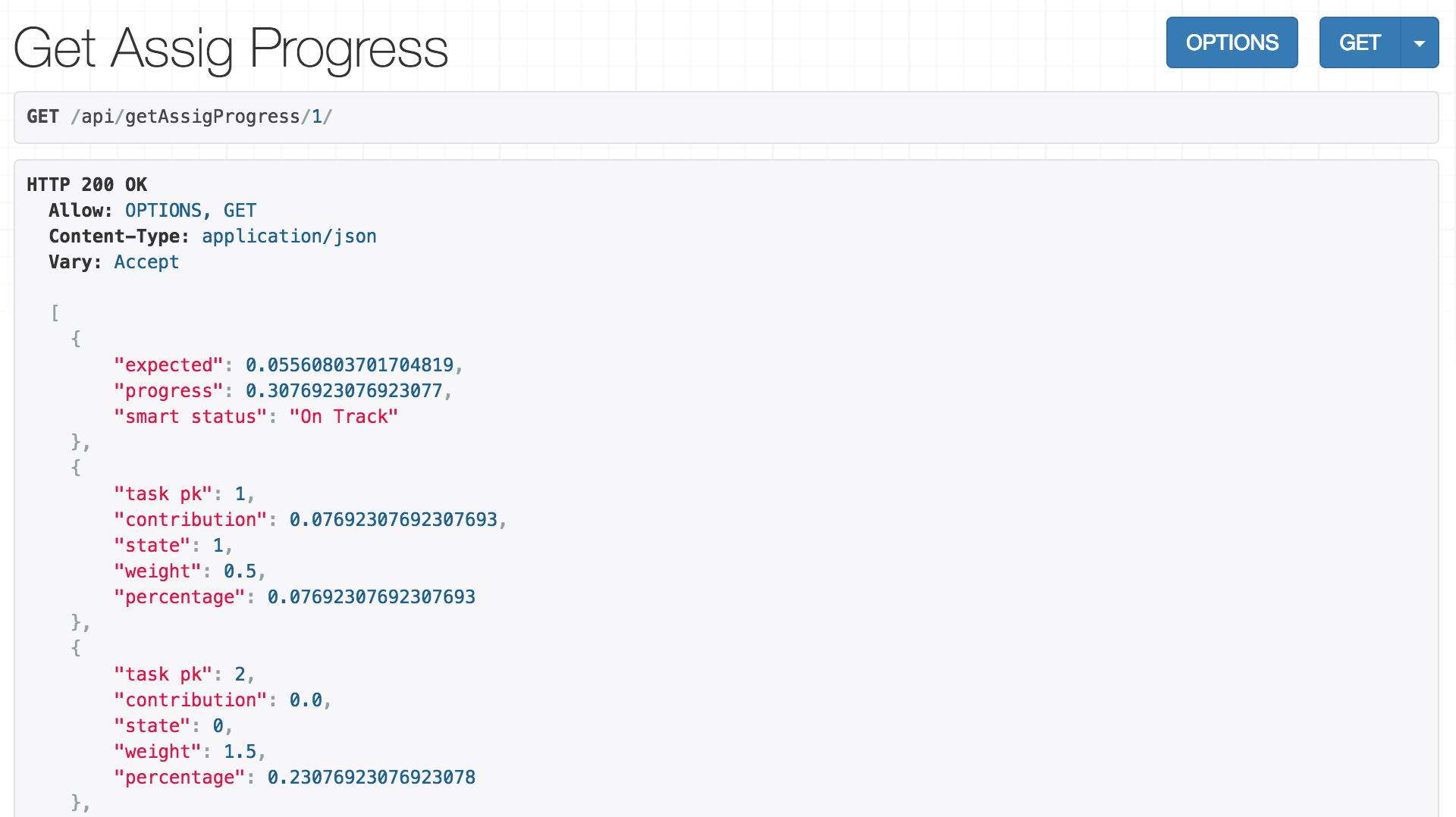
* Error message saying that the user doesn’t have any tasks in this assignment

**Actual Result**

* Error message saying that the user doesn’t have any tasks in this assignment

**Visual User Guide**

Individual assignment student progress & relevant information



Collaboration page updated to include the class average for the given assignment

