

=====

Assignment 2:

Core features to implement by the end of Milestone 2, due by end of A2:

Priority:

- Everything except manual testing. These are all essential features of the app. Without these features, then we wouldn't be able to do the basics of the app. DB isn't top priority, because we can write the manual testing using mock data or a local db.
- Manual testing. Unit tests provide additional validation
 - Create db/schemas and populate our downloaded guideline definitions (James)
 - Download a zip file of guidelines in json format from <https://build.fhir.org/ig/HL7/cqf-recommendations/definitions.json.zip>
 - Create a new database called "guidelinesdb"
 - Creates schemas
 - Create two collections for "plandefinitions" and "activitydefinitions"
 - Insert all of the plan/activity definitions from the zip file into the db
 - Verify that the db works by reading a few of the elements.
 - *AC: I can write a program that can add data to the plan or activity definition collection or read any definition data from the 'guidelinesdb' database.*
 - We should be able to read a list of all guidelines
 - Create a get express endpoint which will return all specified records in JSON format
 - *AC: I should be able to use the /listDefinition endpoint to retrieve a list of all definition ids.*
 - Create a unit test to verify its functionality. (Henry)
 - *AC: I should be able to run the tests using docker, which runs all of our unit tests. The unit tests should include at least one test case for each "typical case" and around 2-3 "edge cases" where appropriate. The group should review and accept all TCs at the end.*
 - We should be able to retrieve a specific guideline
 - Create a get express endpoint which will return a specific records given an id in JSON format
 - Creating different router endpoints depending on which guideline type is requested
 - *AC: I should be able to use the get /<resourceType>/<id> endpoint to retrieve the specific guideline of <id>.*
 - create a unit test to verify it's functionality. (Daniel)
 - *I should be able to run the tests using docker, which runs all of our unit tests. The unit tests should include at least one test case for*

each “typical case” and several “edge cases” where appropriate. The group should review and accept all TCs at the end.

- We should be able to add a new guideline to the db
 - Create a unit test to verify it's functionality. (Stew)
 - *AC: I should be able to use the <endpoint> to retrieve the specific guideline of <id>.*
 - Create a router endpoint to add a new guideline to the db.
 - *AC: I should be able to run the tests using docker, which runs all of our unit tests. The unit tests should include at least one test case for each “typical case” and several “edge cases” where appropriate. The group should review and accept all TCs at the end.*
- We should additionally verify the above functionality using manual testing with Postman (Henry/Daniel/Stew)
 - *AC: Using postman, I can try requesting all of the endpoints (with several different varying inputs) using manual testing and receive the expected response.*

All of the provided AC for milestone 2 have been reviewed and confirmed by our industry partner. Meeting all the AC for each milestone will be our measure of success for this project. Our partner also confirmed that the contents of our milestones are exactly within expectations and that we're on track.

Next steps to pursue:

- Milestone 3: Additional functionality + unit tests, integration tests, documentation (done by: A3 deadline)

Priority:

- 1 and 2, these are core features of the app and also blocks remaining tasks
 - 5, ensures end to end functionality of the app
 - 6, provides documentation for our app's users
 - 3 and 4, additional testing to ensure app quality
 - 7, a stretch goal if we are ahead of schedule
1. We should be able to update a guideline (Daniel, halfway between A2 and A3)
 - Determine what fields needs to be updated (ex. version number)
 - Create a router endpoint to update the json of a specified guideline.
 - *AC: I should be able to use the <endpoint> to update the specific guideline of <id>.*
 - Create a unit test to verify its functionality.
 - *AC: I should be able to run the tests using docker, which runs all of our unit tests. The unit tests should include at least one test case for each “typical case” and about 2- 3 “edge cases” where*

appropriate. The group should review and accept all TCs at the end.

2. We should be able to delete guidelines (James, halfway between A2 and A3)
 - Create a router endpoint to delete a specified guideline.
 - This should be a soft-delete, and shouldn't be removed from the DB.
 - *AC: I should be able to use the <endpoint> to delete the specific guideline of <id>.*
 - Create a unit test to verify it's functionality.
 - *AC: I should be able to run the tests using docker, which runs all of our unit tests. The unit tests should include at least one test case for each "typical case" and about 2 -3 "edge cases" where appropriate. The group should review and accept all TCs at the end. When "deleted" the guideline is still in the db but can no longer be accessed (we can check manually).*
3. We should additionally verify the above functionality using manual testing with Postman (Daniel/James , End of A3)
 - Create get requests like the unit tests and ensure that all of them arrive with the expected response.
 - *AC: Using postman, I can try at least all of the test cases (with several different varying inputs) using manual testing and receive the expected response.*
4. Our test coverage at the end is $\geq 75\%$ (Daniel/James, End of A3)
 - Run jest check the coverage.
 - If it's less than 75%, then write more unit tests until it reaches the threshold.
 - *AC: I can run jest with the --coverage flag and the result is $\geq 75\%$ code coverage.*
5. We have written integration tests to verify the following functionality (Henry, End of A3):
 - Should verify below for both activity and plan definition:
 - adding a new guideline to DB
 - reading a list of all available guidelines
 - updating guideline's json
 - retrieve the guideline and verify contents
 - delete the guideline.
 - *AC: The test case does exactly as we specified above. We should attempt 2-3 "atypical" edge cases similar to the respective unit tests.*
6. We have either a website or readme in our github repo (undecided), which documents the following (Stew, End of A3)
 - What our project is about.
 - What the various API calls do, how to use them and an example of each.
 - *AC: I can access the website or readme, which clearly states the purpose of the project and documents all API calls, how to use them and has an example.*

7. Stretch goal: We should be able to get the activity definition referenced by a plan definition (Whoever finishes early, No due date)
- Create a router endpoint to retrieve an activity definition referenced by a plan definition.
 - This data can already be retrieved by getting the json data of the specified definition, which is why it's a stretch goal.
 - *AC: I should be able to use the <endpoint> to retrieve an activity definition for a specified plan definition.*
 - Create a unit test to verify its functionality.
 - *AC: I should be able to run the tests using docker, which runs all of our unit tests. The unit tests should include at least one test case for each "typical case" and about 2- 3 "edge cases" where appropriate. The group should review and accept all TCs at the end.*

Demonstrated progress towards one or two of your next milestones. Demonstrating progress includes:

- Clarity from your industry partner about what constitutes success in this milestone, in a paragraph.
- Each team member's responsibilities for reaching the milestone, with a status (complete, in progress, not started)
- Your validation process: how will you know that you have achieved this milestone?

What constitutes success?

All of the provided AC have been reviewed and confirmed by our industry partner. Meeting all the AC for each milestone will be our measure of success for this project. Our partner also confirmed that the contents of our milestones are exactly within expectations and that we're on track.

Our members are responsible for reaching the acceptance criteria of their respective features:

- Create db/schemas and populate our downloaded guideline definitions (James) [In progress]
- We should be able to read a list of all guidelines [Completed]
- We should be able to retrieve a specific guideline [Completed]
- We should be able to add a new guideline to the db [In progress]
- We should additionally verify the above functionality using manual testing with Postman (Henry/Daniel/Stew) [In progress]

Validation process: We will achieve the milestone if we satisfy all of the acceptance criterias for our features.