⊟ U1T1: Organization Model

in list <u>Done</u>

MEMBERS   STORY POINTS

JM   1

☰ Description <u>Edit</u>

Organization(**id**, name, country, administrative_area, locality, postal_code, thoroughfare, premise)

Storing an address in a database

> country => Country (always required, 2 character ISO code)
> name_line => Full name (default name entry)
> first_name => First name
> last_name => Last name
> organisation_name => Company
> administrative_area => State / Province / Region (ISO code when available)
> sub_administrative_area => County / District (unused)
> locality => City / Town
> dependent_locality => Dependent locality (unused)
> postal_code => Postal code / ZIP Code
> thoroughfare => Street address
> premise => Apartment, Suite, Box number, etc.
> sub_premise => Sub premise (unused)
>
> Source <u>https://www.drupal.org/project/addressfield</u>

Creating a Model in Spring Boot

> *Model*: in MVC represents the data, and does nothing else. The model does NOT depend on the controller or the view.
>
> Source <u>https://www.tomdalling.com/blog/software-design/model-view-controller-explained</u>

Note: In the guide, stop when you get to creating a repository

<u>https://spring.io/guides/gs/accessing-data-jpa</u>

⧉ Attachments

Organization Model EER.jpg ⧉

Added Oct 17 at 7:27 PM - Comment - Delete

🖵 Remove Cover

Add an attachment…

☑ Checklist

Hide completed items   Delete.

100% ▭

☑ *Create EERD and attach screenshot to this card*

☑ *Create Java classes for* ~~Organization~~ *entity (model)*

☰ Activity

Show Detai

JM **Jacob McMorrow** Oct 17 at 7:27 PM

Organization Model EER.jpg

- Reply - Delete

U1T2: Organization Repository

in list Done

MEMBERS   STORY POINTS

CZ   1

Description Edit

Create a database helper method that inserts an Organization into the MySQL Database

Since this is the first time we interact with the database programmatically, we need to configure Spring JDBC.

Use the `Organization` model created in the task that this card depends on to create an `OrganizationRepository` for database CRUD operations.

Creating a Repository in Spring Boot

> *Repository*: provides an abstraction of data, so that your application can work with a simple abstraction that has an interface approximating that of a collection. Adding, removing, updating, and selecting items from thi collection is done through a series of straightforward methods, without the need to deal with database concerns like connections, commands, cursors, or readers.
>
> Source https://deviq.com/repository-pattern

Note: In the guide, skip creating entities (they are already created), stop when you get to creating a controller

https://spring.io/guides/gs/accessing-data-mysql

☑ Dependencies

Hide completed items   Delete.

100% ▕▒▔▔▔▔▔▔▔▔▔▔▔▔▔▔▔▔▔▔▔▔▔▔▔▔▔▔▔▔

☑ 📘 *U1T1: Organization Model*

☑ Checklist

Hide completed items   Delete.

100% ▕▒▔▔▔▔▔▔▔▔▔▔▔▔▔▔▔▔▔▔▔▔▔▔▔▔▔▔▔▔

☑ *Create* `OrganizationRepository` *repository*

☰ Activity

Show Detai

▤ U1T3: Organization Controller

in list <u>Done</u>

MEMBERS   STORY POINTS

1

≡ Description <u>Edit</u>

Create an `OrganizationController` leave it empty (do not create any route mappings).

☑ Checklist

<u>Hide completed items</u> <u>Delete.</u>

100%

☑ ~~Create~~ `~~OrganizationController~~` ~~controller~~

≡ Activity

<u>Show Detai</u>

U1T4: Organization Creation View

in list <u>Done</u>

MEMBERS	STORY POINTS
DR	2

Description <u>Edit</u>

Create an HTML Form that creates an organization object and POSTs to the route `/organizations/new`.

```
<a href="/organizations">All Organizations</a>

<main>
  <form method="POST" action="/organizations">
    <input type="text" name="name" />
    <input type="submit" />
  </form>
</main>
```

Create a View in Spring Boot

> *View*: The view displays the model data, and sends user actions (e.g. button clicks) to the controller. The view can be independent of both the model and the controller.
>
> Source https://www.tomdalling.com/blog/software-design/model-view-controller-explained

Note: skip creation of `PostMapping` it will be handled by 📖U1T5: Organization Creation Controller

https://spring.io/guides/gs/handling-form-submission

☑ Dependencies

<u>Hide completed items</u> <u>Delete.</u>

100%

- ✓ 📖 ~~U1T1: Organization Model~~
- ✓ 📖 ~~U1T3: Organization Controller~~

☑ Checklist

<u>Hide completed items</u> <u>Delete.</u>

100%

- ✓ ~~Create~~ ~~`organization-create.html`~~ ~~template~~
- ✓ ~~Create~~ ~~`/organizations/create`~~ ~~GET mapping in~~ ~~`OrganizationController`~~

U1T5: Organization Creation Controller

in list Done 👁

MEMBERS   STORY POINTS

2

📝 Description Edit

Create a `PostMapping` under the route `/organizations` to receive the form data from the view use the data to create a new `Organization` (created by 📖U1T1: Organization Model) using the `OrganizationRepository` (created by 📖U1T2: Organization Repository).

Create a `PostMapping` in Spring Boot

Note: skip the creation of a `GetMapping` it will be handled by 📖U1T4: Organization Creation View

https://spring.io/guides/gs/handling-form-submission

☑ Dependencies

Hide completed items   Delete.

100% �_____

✓   📖 *U1T1: Organization Model*

✓   📖 *U1T2: Organization Repository*   ⋯

✓   📖 *U1T3: Organization Controller*

☑ Checklist

Hide completed items   Delete.

100% �_____

✓   *Create* `/organizations` *POST mapping in* `OrganizationController`

✓   `Organization` *is created in the database based on the form data*

✓   *After the mapping, redirect to* `/organizations`

☰ Activity

Show Detai

U2T6: Organization Read List View

in list <u>Done</u>

MEMBERS   **STORY POINTS**

3

Description <u>Edit</u>

Show all organizations in `/organizations` in an unordered list

```
<ul>
  <li>
    <a href="/organizations/1">
      Ontario Council of Agencies Serving Immigrants
    </a>
  </li>
  <li>
    <a href="/organizations/2">
      Ottawa Community Immigrant Services Organization
    </a>
  </li>
</ul>
```

Loop in Thymeleaf

<u>https://www.thymeleaf.org/doc/tutorials/2.1/usingthymeleaf.html#using-theach</u>

Dependencies

<u>Hide completed items</u> <u>Delete.</u>

100% |‾‾‾|

- ✓ 📖 ~~U1T1: Organization Model~~
- ✓ 📖 ~~U1T2: Organization Repository~~
- ✓ 📖 ~~U1T3: Organization Controller~~

Checklist

<u>Hide completed items</u> <u>Delete.</u>

100% |‾‾‾|

- ✓ ~~Create~~ ~~organizations-read-list.html~~ ~~template~~
- ✓ ~~update~~ ~~/organizations~~ ~~GET mapping in~~ ~~OrganizationController~~

U2T7: Organization Read Single View

in list Done

MEMBERS   STORY POINTS

JM   3

Description Edit

Show a single organization in `/organizations/{id}`

```
<a href="/organizations">All Organizations</a>

<main>
  <div>
    <b>Name</b> Ontario Council of Agencies Serving Immigrants
  </div>

  <div>
    <b>Address</b> 200 - 110 Eglinton Avenue West
    <br>
    Toronto, Ontario M4R 1A3 Canada
  </div>

  <a href="/organizations/1/update">Edit</a>
</main>
```

Dependencies

Hide completed items Delete.

100% �_____

✓ 📖 *U1T1: Organization Model*

✓ 📖 *U1T2: Organization Repository*

✓ 📖 *U1T3: Organization Controller*   ⋯

Checklist

Hide completed items Delete.

100% �_____

✓ *Create* `organization-read-single.html`

✓ *Create* `/organizations/{id}` *GET mapping in* `OrganizationController`

✓ *If given ID does not exist, have static 404 page to show instead*

U3T8: Organization Update View

in list Done

MEMBERS   STORY POINTS

DR   2

Description Edit

Edit an organization in `/organizations/{id}/update` the form is identical to 📖U1T4: Organization Creation View however the fields have default values already set using the model passed by the controller.

```
<a href="/organizations">All Organizations</a> &bull; <a href="/organizations/1":

<main>
  <form method="POST" action="/organizations/1">
    <input type="text" name="name" value="Ontario Council of Agencies Serving Imr
    <input type="submit" />
  </form>
</main>
```

If the Organization with id `id` does not exist, show a friendly message saying 404 not found

Dependencies

Hide completed items   Delete.

100% [_____]

- ✓ 📖 *U1T1: Organization Model*
- ✓ 📖 *U1T2: Organization Repository*
- ✓ 📖 *U1T3: Organization Controller*
- ✓ 📖 *U1T4: Organization Creation View*

Checklist

Hide completed items   Delete.

100% [_____]

- ✓ *Create* `organization-update.html` *template*
- ✓ *Create* `/organizations/{id}/update` *GET mapping in* `OrganizationController`
- ✓ *If given ID does not exist, have static 404 page to show instead*

Activity

Show Detai

⊟ U3T9: Organization Update Controller

in list <u>Done</u>

MEMBERS   STORY POINTS

CZ   2

📄 Description <u>Edit</u>

Create a POST mapping under the route `/organizations/{id}` , when the update form is submitted (📋 <u>U3T8: Organization Update View</u>) you should update the `Organization` with the id `id` using the repository

☑ Dependencies

<u>Hide completed items</u> <u>Delete.</u>

100% |▬▬▬▬▬▬▬▬▬▬▬▬▬▬▬▬▬▬▬▬▬▬▬▬▬▬|

✓   📖 *U1T1: Organization Model*

✓   📖 *U1T2: Organization Repository*

✓   📖 *U1T3: Organization Controller*

☑ Checklist

<u>Hide completed items</u> <u>Delete.</u>

100% |▬▬▬▬▬▬▬▬▬▬▬▬▬▬▬▬▬▬▬▬▬▬▬▬▬▬|

✓   *Create* `/organizations/{id}` *POST mapping in* `OrganizationController`   **…**

✓   `Organization` *with organization id* `id` *is updated in the database to match the form data*

✓   *After the mapping, redirect to* `/organizations/{id}` *where id is the* `id` *that was posted to*

☷ Activity

<u>Show Detai</u>

U4T10: Organization Delete View

in list Done

MEMBERS   STORY POINTS

CZ   1

☰ Description Edit

Add a "Delete" button to the 📖U2T7: Organization Read Single View view which sends an HTTP DELETE request to `/organizations/{id}`.

```
<button type="button" onclick="deleteOrganization(1)">
  Delete
</button>

<script>
async function deleteOrganization(id) {
  const yes = confirm("Are you sure you want to delete the organization? You can
  if (yes) {
    // Delete the organization
    await fetch(`/organizations/${id}`, { method: "DELETE" });
    // Navigate the user to /organizations
    window.location = '/organizations';
  }
}
</script>
```

☑ Dependencies

Delete.

0% ▕

☐ 📖U2T7: Organization Read Single View

☑ Checklist

Hide completed items Delete.

100% ▕

☑ ~~Update the~~ ~~`/organizations/{id}`~~ ~~to have a delete button~~

☰ Activity

Show Detai

U4T11: Organization Delete Controller

in list Done   👁

MEMBERS   **STORY POINTS**

2

Description Edit

Delete the `Organization` whose id is `id` when a HTTP DELETE `/organizations/{id}` request is made.

Dependencies

Hide completed items   Delete.

100%

- ☑ 📖 *U1T1: Organization Model*
- ☑ 📖 *U1T2: Organization Repository*
- ☑ 📖 *U1T3: Organization Controller*

Checklist

Hide completed items   Delete.

100%

- ☑ *Create* `/organizations/{id}` *DELETE mapping in* `OrganizationController`
- ☑ `Organization` *with organization id* `id` *is removed from the database*

Activity

Show Detai

U5T12: Organization Template Creation Form

in list <u>Done</u>

MEMBERS   STORY POINTS

2

Description <u>Edit</u>

Create form on the `/organizations/{id}` page that allows file submission

Also requires a dropdown menu for the type of template

- NARS Template : `"nars_template"`

<u>https://spring.io/guides/gs/uploading-files/</u>

Dependencies

<u>Hide completed items</u> <u>Delete.</u>

100%

✓ 📖 ~~U2T7: Organization Read Single View~~

Checklist

<u>Hide completed items</u> <u>Delete.</u>

100%

✓ ~~Update Organization Read Single View to include a new form~~

✓ ~~Form includes a dropdown menu for choosing type of template~~

✓ ~~Form includes a field for file upload~~   ...

✓ ~~Posts to TemplateController "/templates"~~

Activity

<u>Show Detai</u>

⊟ U5T13: Template Model Interface

in list <u>Done</u>   👁

MEMBERS   STORY POINTS

👤   1

☰ Description <u>Edit</u>

Create an interface that represents all template models for iCare
(is empty for the time being)

☑ Checklist

<u>Hide completed items</u>   <u>Delete.</u>

100% ▕▔▔▔▔▔▔▔▔▔▔▔▔▔▔▔▔▔▔▔▔▔▔▔▔▔▔▔▔▔▔▔▔▏

☑   ~~Create~~ ~~`TemplateInterface`~~ ~~class~~

☷ Activity

<u>Show Detai</u>

Omar Chehab Oct 25 at 11:31 PM

> Done 3530be64acb233993ea11214b41cd6766a63f38b

- <u>Edit</u> - <u>Delete</u>

chedy sankar Oct 24 at 2:43 PM

> do it for polymorphism

- <u>Reply</u> - <u>Delete</u>

U5T14: Template NARs Model

in list Done 👁

MEMBERS **STORY POINTS**

1

Description Edit

NARsTemplate(processing_details, update_record_id, client_validation_type_id, client_validation_id, client_birth_dt, postal_cd, assessment_start_dt, assessment_language_id, preferred_official_language_id, institution_type_id, assessment_referral_id, knwl_life_needs_ind, knwl_life_referrals_ind, knwl_comm_gov_svcs_needs_ind, knwl_comm_gov_svcs_referrals_ind, knwl_working_needs_ind, knwl_working_referrals_ind, knwl_education_needs_ind, knwl_education_needs_referrals_ind, social_networks_needs_ind, social_networks_referrals_ind, professional_networks_needs_ind, professional_networks_referrals_ind, local_community_svcs_needs_ind, local_community_svcs_referrals_ind, community_involvement_needs_ind, community_involvement_referrals_ind, language_skills_needs_ind, language_skills_referrals_ind, language_skills_needs_reason_id, other_skills_needs_ind, other_skills_referrals_ind, other_skills_needs_reason_id, find_employment_needs_ind, find_employment_referrals_ind, find_employment_period_id, canadian_employment_experience_id, intended_work_noc_group_id, intention_credentials_licence_ind, intention_canadian_citizen_ind, support_required_services_required, childminding_required_ind, transportation_required_ind, support_disability_required_ind, translation_required_ind, interpretation_required_ind, counselling_required_ind non_cic_program_needs_services_required, food_cloth_material_needs_ind, food_cloth_material_refs_ind, housing_accommodation_needs_ind, housing_accommodation_referrals_ind, hlth_mental_wellness_needs_ind hlth_mental_wellness_refs_ind, financial_needs_ind, financial_referrals_ind, family_support_needs_ind, family_support_referrals_ind, language_non_CIC_needs_ind, language_non_CIC_referrals_ind, edu_skills_devt_needs_ind, edu_skills_devt_referrals_ind, employment_related_needs_ind, employment_related_referrals_ind, legal_info_services_needs_ind, legal_info_services_referrals_ind, community_services_needs_ind, community_services_referrals_ind, support_received_ind, childminding_ind, childminding_NewcomerChildren[1]childminding_age_id, childminding_NewcomerChildren[1]childminding_type_id, childminding_NewcomerChildren[2]childminding_age_id, childminding_NewcomerChildren[2]childminding_type_id, childminding_NewcomerChildren[3]childminding_age_id, childminding_NewcomerChildren[3]childminding_type_id, childminding_NewcomerChildren[4]childminding_age_id, childminding_NewcomerChildren[4]childminding_type_id, childminding_NewcomerChildren[5]childminding_age_id, childminding_NewcomerChildren[5]childminding_type_id, transportation_ind, support_disability_ind, translation_ind, translation_language_from_id, translation_language_to_id, interpretation_ind, interpretation_language_from_id, interpretation_language_to_id, counselling_ind, settlement_plan_ind, assessment_completed_dt, assessment_update_reason_id)

☑ Dependencies

Hide completed items Delete.

100%

✓ 📖 *U5T13: Template Model Interface*

☑ **Checklist**

100%

✓ *Create model with above attributes, use camel case instead of kebab case*

✓ *Add foreign key to organisation entity*

☰ **Activity**

Omar Chehab Oct 26 at 12:02 AM

> Done

  - Edit - Delete

Omar Chehab Oct 26 at 12:01 AM

> Added organization foreign key 7f9280eb2332b447633217f38293db65468395f2

  - Edit - Delete

Omar Chehab Oct 26 at 12:01 AM

> https://hellokoding.com/jpa-one-to-many-relationship-mapping-example-with-spring-boot-maven-and-mysql/

  - Edit - Add Link as Attachment - Delete

Omar Chehab Oct 26 at 12:01 AM (edited)

> Added model 6a4fee4267da33ea02d323ee3d9b3039b62f0199

  - Edit - Delete

Omar Chehab Oct 25 at 11:19 PM

> Add foreign key to organization entity

  - Edit - Delete

chedy sankar Oct 24 at 2:46 PM

getters and setters are important eh

- <u>Reply</u> - <u>Delete</u>

U5T15: Template Controller

in list Current Work

MEMBERS   STORY POINTS

3

Description Edit

Create a controller which takes in the file input from the website and passes it to the necessary wrappers to add the given template to the database.

Dependencies

Hide completed items Delete.

67% |�|

✓ 📖 *U5T16: Sheet Adapter Wrapper*

✓ 📖 *U5T19: Template Factory Wrapper*

☐ 📖 U5T22: Template Repository Wrapper

Checklist

Delete.

0% | |

☐ Controller passes file to adapter wrapper to receive hash map list representation

☐ Controller passes hash map list to factory wrapper to receive template model

☐ Controller passes model to repository wrapper to add the model to database

Activity

Show Detai

U5T16: Sheet Adapter Wrapper

in list <u>Done</u> 👁

MEMBERS   STORY POINTS

1

Description <u>Edit</u>

Create a wrapper which handles the choosing of which sheet adapter to use based on the file type.

Dependencies

<u>Hide completed items</u> <u>Delete.</u>

100%

✓ 📖 *U5T18: CSV Sheet Adapter*

Activity

<u>Show Detai</u>

Omar Chehab Oct 26 at 1:38 AM

> Done 77ed708fa6490321d2de0f811132a1fd04623304

- <u>Edit</u> - <u>Delete</u>

U5T17: Sheet Adapter Interface

in list <u>Done</u> 👁

MEMBERS   STORY POINTS

1

☰ Description <u>Edit</u>

Create an interface which represents all sheet adapters, with generic `parse` method

☑ Checklist

<u>Hide completed items</u> <u>Delete.</u>

100%

☑ ~~Create~~ `SheetAdapterInterface` ~~interface, returns~~ `List<HashMap<String, String>>`

☑ ~~Method called~~ `parse` ~~that takes parameter~~ `File` ~~and returns~~ `List<HashMap<String, String>>`

☰ Activity

<u>Show Detai</u>

Omar Chehab Oct 26 at 1:11 AM (edited)

Done 8b242a7161fc50da494fa95143b50070000517e5

I added two parameters `headerRow` and `dataStartRow` to know which row the header is and which row the data start on. I did this because the NARSTemplate's header is on row 2 and the data starts on row 4.

I made it an abstract class instead because I overloaded the function for default values (headerRow: 0, dataStartRow: 1)

- <u>Edit</u> - <u>Delete</u>

U5T18: CSV Sheet Adapter

in list <u>Done</u> 👁

MEMBERS   STORY POINTS

2

Description <u>Edit</u>

Create CSVSheetAdapter that implements the adapter interface

Useful: <u>https://www.geeksforgeeks.org/reading-csv-file-java-using-opencv/</u>

Dependencies

<u>Hide completed items</u>   <u>Delete.</u>

100% |

✓   📖 *<u>U5T17: Sheet Adapter Interface</u>*

Checklist

<u>Hide completed items</u>   <u>Delete.</u>

100% |

✓   *Create* `CSVSheetAdapter` *class that implements* `SheetAdapterInterface`

✓   *Implement the parse method using the OpenCSV library*

✓   *Convert 2d string array into list of hashmaps*

Activity

<u>Show Detai</u>

Omar Chehab Oct 26 at 1:12 AM

> Done 47da785d67b11948d5520f8b2bcd01b71ce1b481

  - <u>Edit</u> - <u>Delete</u>

U5T19: Template Factory Wrapper

in list <u>Done</u>

MEMBERS   STORY POINTS

JM   1

Description <u>Edit</u>

Create a wrapper which handles the choosing of which factory to use based on the template type.

Dependencies

<u>Hide completed items</u> <u>Delete.</u>

100% |_____|

✓ 📖 *<u>U5T21: Template NARs Factory</u>*

Activity

<u>Show Detai</u>

U5T20: Template Factory Interface

in list <u>Done</u>

MEMBERS   STORY POINTS

JM   1

Description <u>Edit</u>

Create factory interface for template entities

Activity   <u>Show Detai</u>

U5T21: Template NARs Factory

in list <u>Done</u>

MEMBERS   STORY POINTS

JM   2

Description <u>Edit</u>

Create NARs factory that implements the factory interface
Takes single hashmap, returns model

Dependencies

<u>Hide completed items</u>   <u>Delete.</u>

100% |_____|

☑️ 📘 *<u>U5T20: Template Factory Interface</u>*

Activity   <u>Show Detai</u>

chedy sankar Oct 24 at 3:01 PM (edited)

USE AN IDE FOR THIS

- <u>Reply</u> - <u>Delete</u>

U5T22: Template Repository Wrapper

in list <u>Done</u> 👁

MEMBERS   STORY POINTS

1

Description <u>Edit</u>

Create a wrapper which handles the choosing of which repository to use based on the template type.

Dependencies

<u>Hide completed items</u>   <u>Delete.</u>

100% |▢_____|

☑ 📑 *<u>U5T24: Template NARs Repository</u>*

Activity

<u>Show Detai</u>

U5T23: Template Repository Interface

in list <u>Done</u>

MEMBERS   STORY POINTS

CZ   1

Description <u>Edit</u>

Create an interface which represents all template repositories (Currently empty)

Activity   <u>Show Detai</u>

U5T24: Template NARs Repository

in list <u>Done</u>

MEMBERS   STORY POINTS

CZ   1

Description

Add a more detailed description…

Dependencies

<u>Hide completed items</u> <u>Delete.</u>

100%

- ☑ 📖 <u>*U5T23: Template Repository Interface*</u>
- ☑ 📖 <u>*U5T14: Template NARs Model*</u>

Activity   <u>Show Detai</u>

chedy sankar Oct 24 at 3:03 PM

> 2 lines of code ez

- <u>Reply</u> - <u>Delete</u>

▭ U6T25: Template View

in list <u>Done</u>

MEMBERS	STORY POINTS
DR	1

▤ Description <u>Edit</u>

Show all templates in /templates in an unordered list

☑ Dependencies

<u>Delete.</u>

0% ▐_____

☑ Checklist

<u>Hide completed items</u> <u>Delete.</u>

100% ▐_____

- ✓ ~~Create~~ ~~/templates~~ ~~get mapping~~
- ✓ ~~Show an unordered list of hyperlink references to all the different templates (NARS Template for now)~~
- ✓ ~~Create a~~ ~~templates/NARs~~ ~~get mapping~~

☰ Activity

<u>Show Detai</u>

U6T26: Entity Table Fragment

in list <u>Current Work</u>

MEMBERS   STORY POINTS

DR   3

Description <u>Edit</u>

<u>https://www.baeldung.com/spring-thymeleaf-fragments</u>

Checklist

<u>Delete.</u>

0%

☐ Given an entity and it's repository, show an HTML table for each attribute and a row for each row in the database.

Activity

<u>Show Detai</u>

U6T27: Entity Table Select Parameter

in list Sprint Backlog

STORY POINTS

2

Description

Add a more detailed description…

Dependencies

Delete.

0%

☐ 📖 U6T26: Entity Table Fragment

Checklist

Delete.

0%

☐ When the query string parameter `select` is set, only show requested columns. For example `?` `select=updateRecordId,clientBirthDt,postalCd` should show the columns in that order.

☐ By default, when `select` is not set, show all columns.

Activity

Show Detai

U6T28: Entity Table Where Parameter

in list Sprint Backlog

STORY POINTS

2

Description

Add a more detailed description…

Dependencies

Delete.

0%

- [ ] 📘 U6T26: Entity Table Fragment

Checklist

Delete.

0%

- [ ] When the query string parameter `where.<column>` is set, only show the rows that match the condition for that column. For example `?where.postalCd=M1C` should show rows whose `postalCd` is equal to `"M1C"`.
- [ ] By default, when `where` is not set, do not filter any rows
- [ ] Support wildcard operator `%`. For example, `?where.postalCd=M1C%` should only show rows that start with `"M1C"`.
- [ ] Wildcard operator can appear more than once. For example, `?where.postalCd=%M1C%` should only show rows that contain `"M1C"`.

Activity

Show Detai

⊟ U6T29: Entity Table Sort Parameter

in list Sprint Backlog

STORY POINTS

2

▤ Description

Add a more detailed description…

☑ Dependencies

Delete.

0% |

☐ 📖 U6T26: Entity Table Fragment

☑ Checklist

Delete.

0% |

☐ When the query string parameter `sort` is set, sort the table by the given column. For example `?sort=updateRecordId` would sort by `updateRecordId`.

☐ When the query string parameter `sortDirection` can be set to `asc` or `desc`. It should sort the rows ascending or descending. Does nothing if `sort` is not set.

☐ If `sort` is not set, don't sort.

☐ If `sortDirection` is not set and `sort` is set, sort ascending by default.

☰ Activity

Show Detai

U6T30: NARS Template View

in list Sprint Backlog

STORY POINTS

1

Description

Add a more detailed description…

Dependencies

Hide completed items   Delete.

50%

- [x] 📘 *U5T14: Template NARs Model*
- [ ] 📘 U6T26: Entity Table Fragment

Checklist

Hide completed items   Delete.

50%

- [x] *Create a* `/templates/nars` *get mapping*
- [ ] Show a `TableFragment` for the `NARsTemplate` entity.

Activity

Show Detai

U22T##: Account Create

in list <u>Backlog</u>

STORY POINTS

Unestimated

Description

Add a more detailed description…

Dependencies

<u>Delete.</u>

0% |

☐ 📖<u>U22T##: Account Model and Repository</u>

Checklist

<u>Delete.</u>

0% |

☐ Create `account-create` template with HTML `form` with an input for each attribute in `Account`

☐ Create or use existing `AccountController` class

☐ Create `GetMapping` on `/accounts/new` in `AccountController` that returns `account-create`

☐ Create `PostMapping` on `/accounts` in `AccountController` that redirects to `/accounts`

☐ After submitting the form, a new account must be created in the database

Activity

<u>Show Detai</u>

▣ U22T##: Account Model and Repository

in list <u>Backlog</u>

STORY POINTS

Unestimated

☰ Description <u>Edit</u>

Account(**id**, firstName, lastName, email, password, phone)

☑ Checklist

<u>Delete.</u>

0%

☐ Create `Account` model with attributes described above

☐ Create `AccountRepository`

☰ Activity

<u>Show Detai</u>

⊟ U23T##: Account Read

in list <u>Backlog</u>

STORY POINTS

Unestimated

☰ Description

Add a more detailed description…

☑ Dependencies

<u>Delete.</u>

0% |▮

☐ 📘 <u>U22T##: Account Model and Repository</u>

☑ Checklist

<u>Delete.</u>

0% |▮

☐ Create `account-read-list.html` template that renders a list of accounts

☐ Create `GetMapping` on `/accounts` that returns `account-read-single` and shows all accounts in the database

☐ Create `account-read-single.html` template that renders a single account

☐ Create `GetMapping` on `/accounts/{id}` that returns `account-read-list` and shows the account with a given id that exists in the database

☐ If the account does not exist, throw `AccountNotFoundException`

☷ Activity

<u>Show Detai</u>

⊟ U24T##: Account Update

in list <u>Backlog</u>

STORY POINTS

Unestimated

☰ Description

Add a more detailed description…

☑ Dependencies

<u>Delete.</u>

0% |▎

⬚ 📖 <u>U22T##: Account Model and Repository</u>

☑ Checklist

<u>Delete.</u>

0% |▎

⬚ Create `account-update.html` template that shows a form with default values given by the controller

⬚ Create `GetMapping` on `/accounts/{id}/update` that returns `account-update`

⬚ Create `PostMapping` on `/accounts/{id}` that updates an account by a given id in the database based on the form data and redirects to `/accounts` when successful

⬚ If the account does not exist, throw `AccountNotFoundException`

☷ Activity

<u>Show Detai</u>

⊟ U25T##: Account Delete (Controller)

in list <u>Backlog</u>

STORY POINTS

Unestimated

☰ Description

Add a more detailed description…

☑ Dependencies

<u>Delete.</u>

0% |

☐ 📘<u>U22T##: Account Model and Repository</u>

☑ Checklist

<u>Delete.</u>

0% |

☐ Create `DeleteMapping` on `/accounts/{id}` that deletes an account by a given id and redirects to `/accounts` when successful

☐ If the account does not exist, throw `AccountNotFoundException`

☰ Activity

<u>Show Detai</u>

U26T##: Account-Organization Controller

in list <u>Backlog</u>

LABELS

STORY POINTS

Missing Checklist

Unestimated

Description

Add a more detailed description…

☑ Dependencies

<u>Delete.</u>

0% |

☐ 📘 <u>U26T##: Account-Organization Foreign Key</u>

☑ Checklist

<u>Delete.</u>

0% |

☰ Activity

<u>Show Detai</u>

▭ U26T##: Account-Organization Foreign Key

in list <u>Backlog</u>

STORY POINTS

Unestimated

≣ Description <u>Edit</u>

One Account may belong to many Organizations
One Organization may employ many Accounts

Create Entity Relationships in Spring

<u>https://www.baeldung.com/spring-data-rest-relationships</u>

☑ Dependencies

<u>Hide completed items</u>   <u>Delete.</u>

50% �_____

✓   📖 *U1T1: Organization Model*

☐   📖 <u>U22T##: Account Model and Repository</u>

☑ Checklist

<u>Delete.</u>

0% |_____

☐   Create a relationship between the two entities as described above

☷ Activity

<u>Show Detai</u>

U26T##: Account-Organization View

in list Backlog

LABELS

Missing Checklist

STORY POINTS

Unestimated

Description

Add a more detailed description…

Dependencies

Delete.

0%

☐ 📖 U26T##: Account-Organization Foreign Key

Checklist

Delete.

0%

Activity

Show Detai