# Implementing a story

1. Analyse the story
2. Create the controller
   1. Include calls to the service you wish to use
3. Create the service
4. Write the service methods needed by the controller
5. Write your view
6. Check your work

**STORY 01 – SHOW TASKS**

As a user

I can retrieve all tasks

So that I can see what I have left to do

**MOCKUP**

A screenshot of a cell phone

Description automatically generated

**ACCEPTANCE CRITERIA**

* You will see the list of tasks. For every task the title and due date are shown.
* If there are no tasks, you will see the message “No tasks left to do”.

**TECHNICAL CRITERIA**

* The path of this view is “/tasks”.

**STORY 02 – SHOW TASK DETAIL**

As a user

Given the id of a task

I can retrieve the detail of a task

So that I can see all its values

**MOCKUP**

A close up of text on a white background

Description automatically generated

**ACCEPTANCE CRITERIA**

* You will see the detail of the task. For every task the title, description and due date are shown.
* If the task for the id could not be found, you will be shown an error message “Task not found”.

**TECHNICAL CRITERIA**

* The path of this view is “/tasks/{id}”.

**STORY 03 – CREATE TASK**

As a user

I can create a new task

So that I can see it in the task overview

**MOCKUP**

A close up of text on a white background

Description automatically generated

**ACCEPTANCE CRITERIA**

* You will see a form to create a new task, with fields for title, description and due date, and a button to submit the form.
* When the task is successfully created, you will be redirected to the task list where you can see the new task.

**TECHNICAL CRITERIA**

* The path of this view is “/tasks/new”.

**STORY 04 – EDIT TASK**

As a user

I can edit a task

So that I can correct any mistakes made in the creation

**ACCEPTANCE CRITERIA**

* You will see a form to edit an existing task, with fields for title, description and due date, and a button to submit the form.
* When the task is successfully edited, you will be redirected to the task detail where you can see the task with its edited fields.

**TECHNICAL CRITERIA**

* The path of this view is “/tasks/edit/{id}”.

**STORY 05 – CREATE SUBTASK**

As a user

I can add subtasks to a task

So that I can divide a task up in smaller amounts of work

**ACCEPTANCE CRITERIA**

* You will see a form to add a subtask, with fields for title and description, and a button to submit the form.
* When the task is successfully edited, you will be redirected to the task detail where you can see the parent task with its subtasks.

**TECHNICAL CRITERIA**

* The path of this view is “/tasks/{id}/sub/create”.

**STORY 06 – NAVIGATION/INDEX**

As a user

I can navigate to all pages from the index page

So that the entire application is properly accessible

**ACCEPTANCE CRITERIA**

* You will see links to all views of the application (withholding those needing specific parameters)

**TECHNICAL CRITERIA**

* The path of this view is “/”.

**ASSIGNMENT : JPA**

Store all domain objects in a database of your choice using Spring Data JPA. Refactor your service classes to use repositories. Ensure that all existing functionality still works.

From now on, all new functionality will make use of the database for storage.

**STORY 07 – I18n**

As a user

I wish to view the application in my chosen language

**DESCRIPTION**

The goal of this story is to enable i18n on your application and provide users with two languages to choose from. Each page must have a button for each available language that sets that language for the user.

**ACCEPTANCE CRITERIA**

* Each page has buttons for all available languages
* When a language is chosen, the entire site is visible in that language

**TECHNICAL CRITERIA**

* Store the locale in the user session

**STORY 06 – SECURITY**

As a user

I am able to access only those pages I have authorization to view

I can sign in to authenticate myself to the application

I can create a new user account to get access the the application

So that the application is secure

**DESCRIPTION**

Implement a way for users to create an account, sign in, sign out and retain access to the application. Create two different levels of authorization: ADMIN and USER.

**ACCEPTANCE CRITERIA**

* ADMIN users can perform the following actions:
  + Create tasks
  + Edit tasks
  + Create subtasks
* All authenticated users can perform the following actions:
  + View all tasks
  + View task details
  + Sign in
  + Sign out
  + Create an account
* A sign out button/link is visible on each page that is limited to authenticated users

**TECHNICAL CRITERIA**

/