**Meal Planner CRUD Application**

**Introduction**

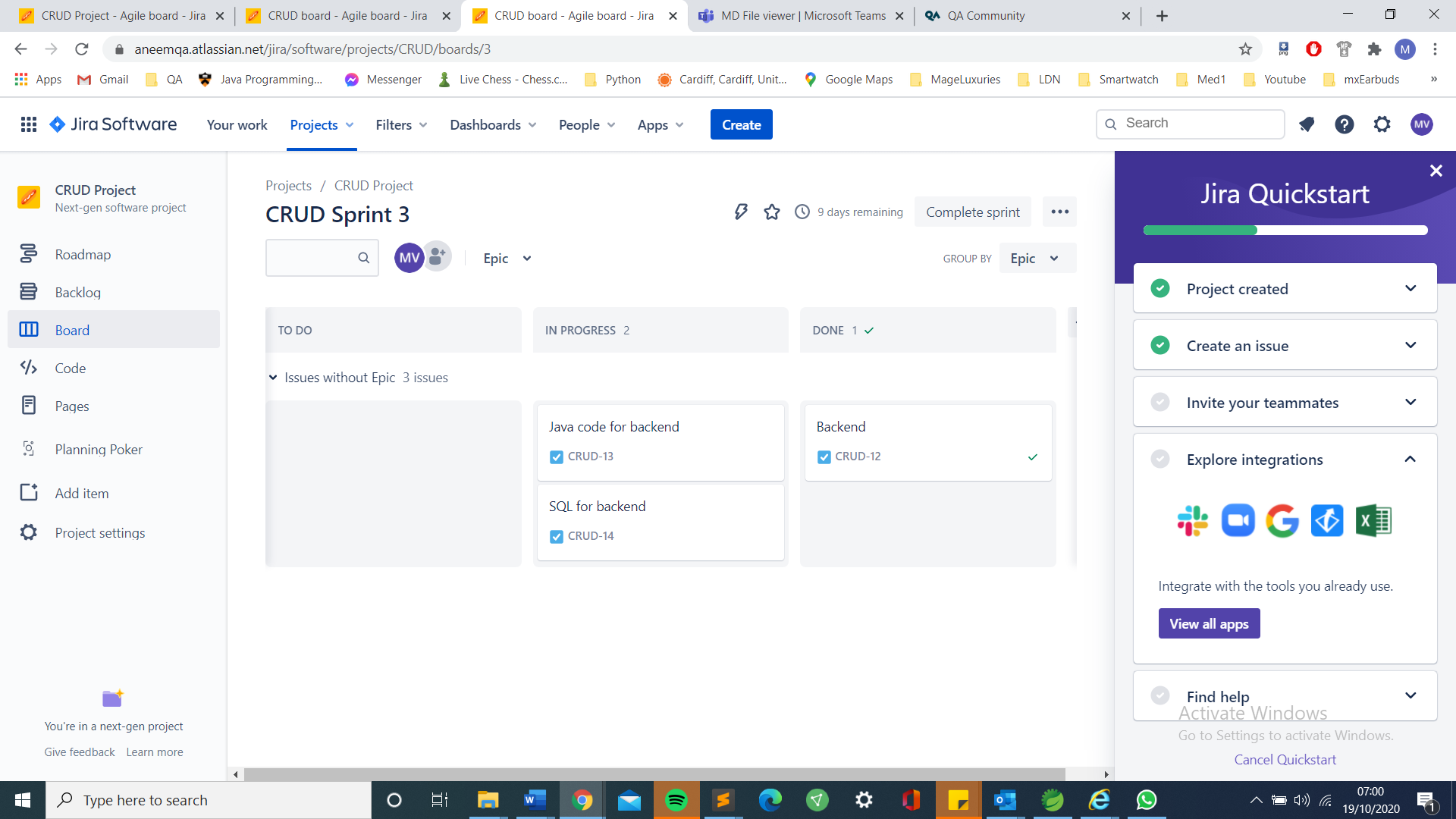
In this document, the planning, creation and testing of a Meal Planner is documented. To complete the project the following languages are used: Java (SpringBoot) for backend with a connection to a Database Management System, HTML for the mark up of the website and CSS for further design. The aims of the project are the encapsulate the four basic functions of storage: Create, Read, Update and Delete. The project will include the following concepts from the QA Academy Cloud Native course as specified:

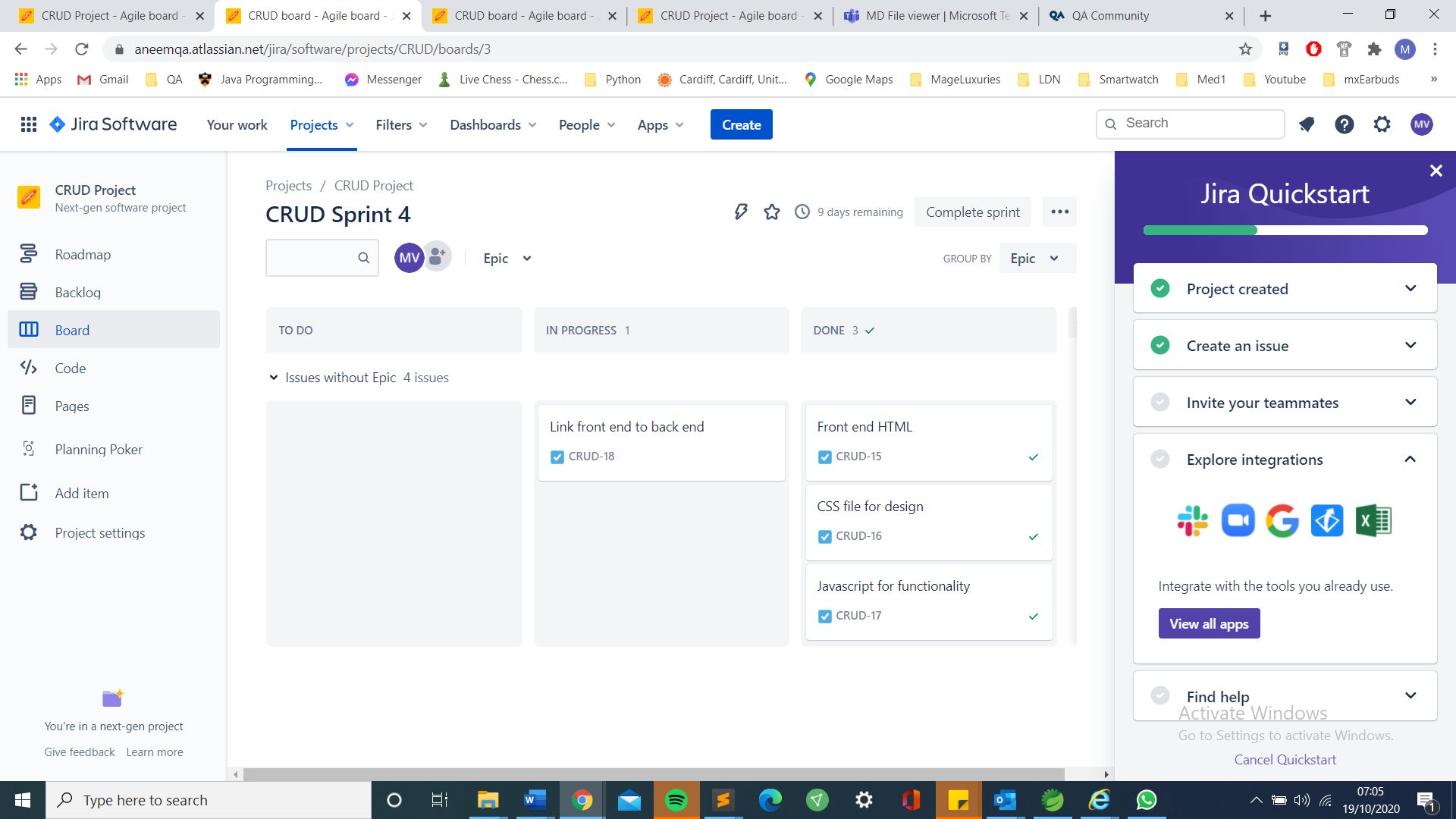
* Project Management
* Databases
* Java SE
* Spring Boot
* Front-End Development
* Automated Testing
* Continuous Integration
* Cloud Fundamentals

**Scope:**

* Jira board for user stories, use cases, project tasks
* Documentation of the design
* Risk Assessment
* Relational Database to store data
* Application created in Java, respecting best practices and design principles
* Front-end website and Integrated API
* Fully designed test suits for the application as well as automated test for validation on the app.
* Near full test coverage for the backend with reports and evidence
* Coding a fully integrated Version Control System

**Jira board for tracking progress**





**Database Table Structure**

|  |
| --- |
| CRUD Meal Planner App |
| ID (Primary Key) |
| Name (of food) |
| Type |
| Day |

**Risk Assessment**

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **Risk** | **Likelihood** | **Severity** | **Current** | **Proposed** | **Tolerance** | **Responsibility** | **Solution** |
| Eye Strain | Moderate | Low | Take regular breaks | Adjust screen and do eye exercises | Tolerable | Marcell Vesz | Regular breaks and exercises for the eye. |
| Loss of code/database | Low | Low | Auto generated DB | Multiple Backups | Tolerable | Marcell Vesz | Cloud/hard backups and using private PC. |
| Hacking | Low | Low | Keep information offline | Use private network | Tolerable | Marcell Vesz | Use private networks when connection to the cloud. |

**Design Plan**

Single table in Database

Empty lists: Monday through Sunday

Types of meals: Breakfast, Lunch, Dinner, Snack

Example

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
|  | Monday | Tuesday | Wednesday | Thursday | Friday | Saturday | Sunday |
| Breakfast |  |  |  |  |  |  |  |
| Lunch |  |  |  |  |  |  |  |
| Dinner |  |  |  |  |  |  |  |
| Snacks |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |

**Edit Update**

**Create Meal:**  
Meal: \_\_\_\_\_\_\_\_\_\_  
Drop down box to choose meal type

Drop down box to choose day

Create Meal Button

**Create meal type:** \_\_\_\_\_\_\_\_\_\_Adds meal type row

Create meal type button

CRUD:

Create: Create meal or meal type

Read: Text box: enter day to retrieve meals for that day

Update/Delete: Make table editable with an edit and an update button

Delete: Clear table or clear day, clear breakfast

**Testing**

Junit/SpringBoot tests have been carried out.

**Future Improvements**

Due to last minute changes, as of now the App is not functional. The project must be completed and a connection to the database must be re-established.