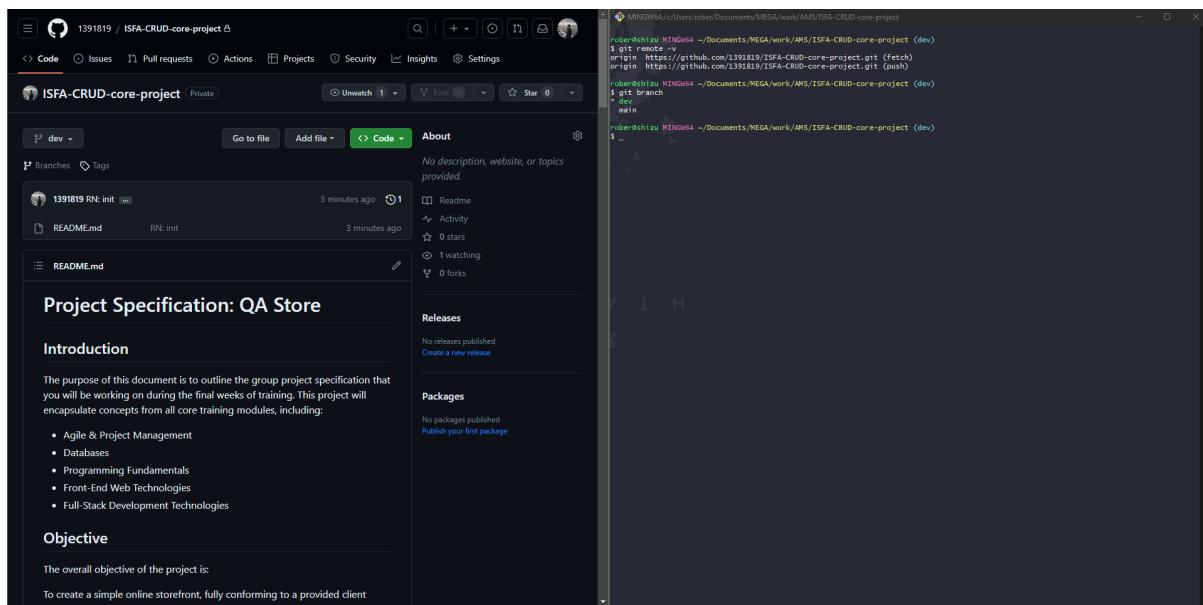


Workflow: screenshots

Git/GitHub setup



Creation of main Git/GitHub repo, with dev and main branches for now. dev contains the project brief, main will contain official documentation

Project Management with Jira

A screenshot of the Jira Software interface. On the left, there's a sidebar with project navigation: 'QA Store (Software project)', 'Planning (Timeline, Board)', 'Development (Code)', 'Project pages', 'Add shortcut', and 'Project settings'. The main area shows a 'QS board' for the 'QA Store' project. The board has three columns: 'TO DO', 'IN PROGRESS', and 'DONE'. A button '+ Create issue' is located at the bottom of the 'TO DO' column. At the top of the board, there are filters, a search bar, and a 'GROUP BY' dropdown set to 'None'. A 'Quickstart' button is at the bottom right. A note at the bottom left says 'You're in a team-managed project'.

Creation of a Kanban project using Jira

The screenshot shows the Jira settings page with several features listed:

- Backlog**: Plan and prioritize your team's work in a dedicated space. [Learn more about Backlog](#). Toggle switch is checked.
- Board**: View, track and manage work as it makes its way through your team's process. [Learn more about Board](#). Toggle switch is unchecked.
- Reports**: Analyse and track your team's work by reporting on the project's activity. [Learn more about Reports](#). Toggle switch is checked.
- Issue navigator**: Browse a list of issues in your project. Use the built-in filters and text search to find issues. [Learn more about Issue navigator](#). Toggle switch is unchecked.
- Sprints**: Complete work in fixed units of time. Requires a backlog. [Learn more about Sprints](#). Toggle switch is checked.
- Estimation**: Capture expected efforts to plan, track and analyse work. Impacts reports and insights. [Learn more about Estimation](#). Toggle switch is checked.
 - Story points:** Use units to estimate complexity, size and effort. (Selected)
 - Time:** Use minutes, hours, and days to estimate effort. The original estimate field is automatically added to issues when first enabled. Configure your fields manually after that.

Enabling useful features for our SDLC (estimation - story points, sprints - we only have two 2-day sprints, reports, and backlog items for prioritisation)

QS board

The screenshot shows a Jira Kanban board titled "QS board". It has four columns: "TO DO", "IN PROGRESS MAX 2", "REVIEW", and "DONE". The "TO DO" column contains a large blue circular arrow icon and the text "Get started in the backlog". Below it is a sub-section with the text "Plan and start a sprint to see issues here." and a "Go to Backlog" button. The "IN PROGRESS" column has a limit of 2 tasks. The "REVIEW" and "DONE" columns are currently empty. A "+" button is located at the top right of the board area.

Refining Kanban board, by limiting the amount of tasks that can be in progress and using a review column as well

The screenshot shows a Jira Backlog screen. At the top, there are filters for "Epic" and "Label". Below that, there are sections for "QS Sprint 1" (10 Aug – 11 Aug, 1 issue) and "QS Sprint 2" (14 Aug – 15 Aug, 2 issues). Each sprint section includes a "Start sprint" button and a "Create issue" link. Under "QS Sprint 2", there are three items: "QS-1 Daily stand-up meeting 1", "QS-2 Daily stand-up meeting 2", and "QS-3 Daily stand-up meeting 3". At the bottom, there is a section for "Backlog (0 issues)" with a "Create issue" link.

Creation of a blank backlog which includes only the two 2-day sprints, and the daily stand-up meetings for the incoming days (what did I do yesterday? What will I do today? Are there any blockers?) - also enabled useful issues features - i.e., prioritisation, could have used MoSCoW prioritisation but that requires the creation of a custom field

Risk[Edit workflow](#) ...

Bugs track problems or errors.

Description fields

Summary	REQUIRED
Severity	REQUIRED >
Likelihood	REQUIRED >
Description	>

Context fields

Status
Assignee
Labels
Sprint
HIDE WHEN EMPTY
Reporter

Give feedback

Discard

Save changes

Creation of a risk issue type, to be used to perform risk assessment and maybe in the future complete a risk assessment matrix

The screenshot shows a Jira board with three sprints:

- Set up**: 10 Aug – 10 Aug (3 issues)
 - ✓ QS-5 Requirements gathering
 - ✓ QS-50 Risk assessment
 - ✓ QS-6 Analysis and Design
- QS Sprint 1**: 11 Aug – 11 Aug (1 issue)
 - ☐ QS-1 Daily stand-up meeting 1
- QS Sprint 2**: 14 Aug – 15 Aug (2 issues)
 - ☐ ✓ QS-2 Daily stand-up meeting 2
 - ☐ ✓ QS-3 Daily stand-up meeting 3

Each sprint has a summary bar at the top with status indicators: IN PROGRESS (green), TO DO (yellow), and REVIEW (blue).

Refining sprint 1 day 1 to Set up “sprint”, which includes requirements gathering, risk assessment, and analysis and design

SDLC-1: Requirements gathering

Creation of all the backlog items part of the MVP (user stories + epics), including story points (Fibonacci sequence) and priority assignments. This was part of the requirement gathering issue. Moreover, the analysis issue has been started and each user story has been broken down into smaller tasks that need to be completed

Risk

Bugs track problems or errors.

[Edit workflow](#)

Description fields ①

Aa Summary	REQUIRED
≡ Mitigation	REQUIRED
Description	
How to mitigate the risk	
Default text	
Normal text v B I ... ≡ ≡ 🔗 @ ⌚ ↔ ➕ ✖ —	
<input checked="" type="checkbox"/> Required Remove	
Severity	REQUIRED >
Likelihood	REQUIRED >
≡ Description	>

Context fields ①

→ Status
👤 Assignee
🏷 Labels
📅 Sprint
👤 Reporter

[HIDE WHEN EMPTY](#)

Added mitigation field to risk issues

SDLC-1: Risk assessment

Projects / QA Store

Backlog

Q

1 selected

Epic

Issues without epic

+ Create issue

Set up 10 Aug – 10 Aug (0 of 3 issues visible)

Setting up the project: defining user stories and tasks using appropriate industry standards, setting up any risk assessment, logging, completing analysis and design section of the SDLC (mainly identifying the database schema).

There are no issues that match your current filter.

+ Create issue

QS Sprint 1 11 Aug – 11 Aug (0 of 3 issues visible)

There are no issues that match your current filter.

+ Create issue

QS Sprint 2 14 Aug – 15 Aug (0 of 2 issues visible)

There are no issues that match your current filter.

+ Create issue

Backlog (5 of 26 issues visible)

QS-00 Inadequate UX design might lead to customer frustration

QS-01 Slow loading times or unresponsive design could lead to decreased conversion rates

QS-02 Processing payment information can expose the website to data theft

QS-03 Confusing contact information might hinder customer feedback

QS-04 Inaccurate products information can lead to customer dissatisfaction and returns

+ Create issue

+ Create epic

Performed relatively simple risk assessment during such an early stage

Set up 10 Aug – 10 Aug (3 issues)

Setting up the project: defining user stories and tasks using appropriate industry standards, setting up any risk assessment logging, completing analysis and design section of the SDLC (mainly identifying the database schema)

TO DO IN PROGRESS 1 MAX 2 REVIEW DONE 2 ✓

Category	Task	Status	Progress	Labels
Analysis and Design	SDLC	IN PROGRESS	MAX 2	
	QS-6	IN PROGRESS	MAX 2	RN
		IN PROGRESS	MAX 2	
Requirements gathering	SDLC	DONE	MAX 2	
	QS-5	DONE	MAX 2	RN
Risk assessment	SDLC	DONE	MAX 2	
	QS-50	DONE	MAX 2	RN

12:47 - Set up sprint update

SDLC-2: Analysis

QS-39 / QS-6

Analysis

Description

1. Work on requirements and define:
 - a. database schema
 - b. store topic
 - i. products being sold
 - ii. categories

Child issues

Issue	Assignee	Status
QS-85 Design database schema	RN Roberto Nacu	TO DO
QS-86 Decide store topic	RN Roberto Nacu	TO DO

Activity

Show: All Comments History Newest first ↴

Add a comment...

Pro tip: press M to comment

Details

In Progress Actions

Assignee: Roberto Nacu (RN)

Priority: Highest

Labels: None

Sprint: Set up

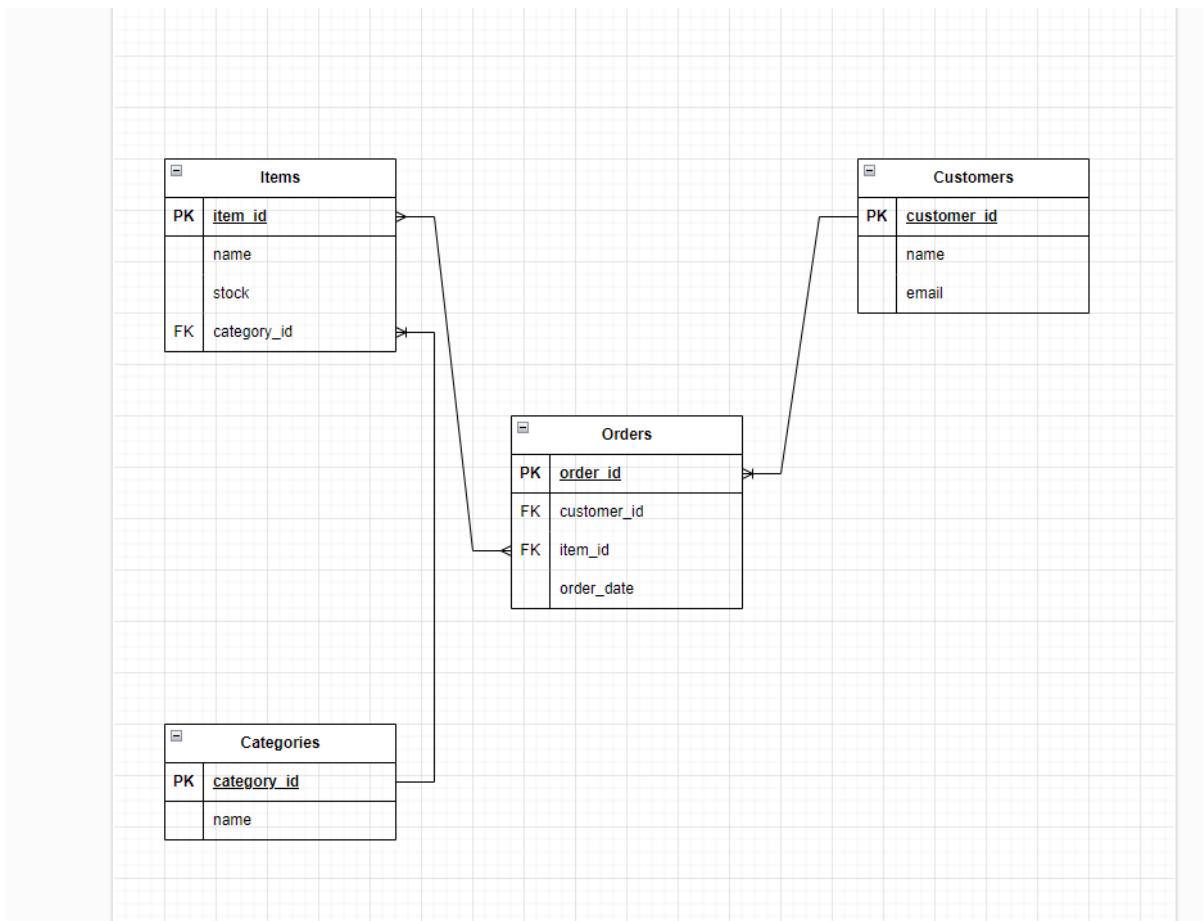
Story point estimate: None

Reporter: Roberto Nacu (RN)

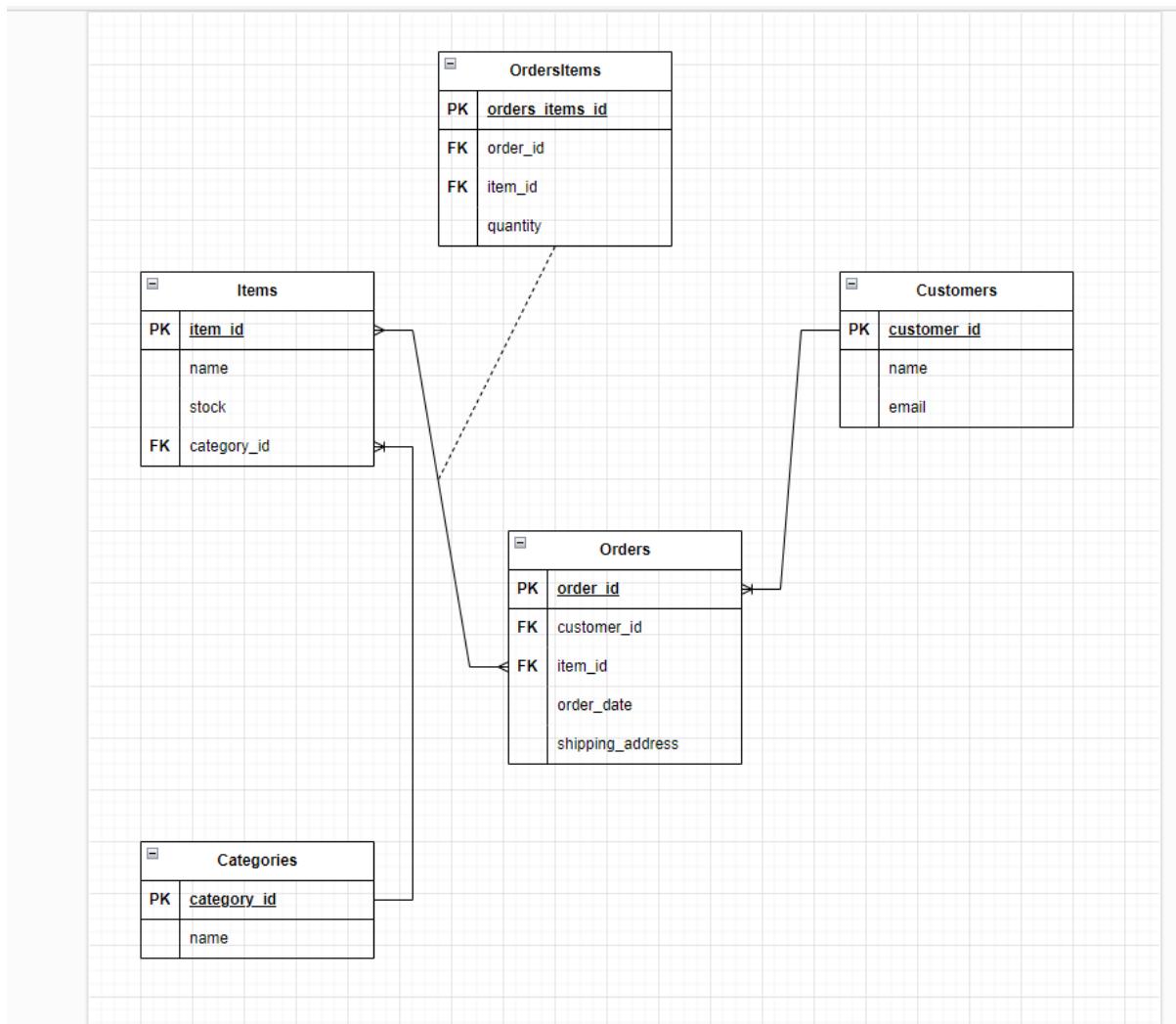
Created 2 hours ago | Updated 13 minutes ago

Configure

Changed Analysis and Design issue to Analysis and added sub-tasks



First design of the database schema



Second design of the database schema, added associative table (many to many relationship) and customer shipping address (although the latter, in the future, for maintainability purposes could be moved in to its own table and add some modularity - street, postcode, city, country)

The screenshot shows a Jira issue page for 'QS-39'. The top navigation bar includes 'QS-39 / QS-6' and various icons for locking, status, and sharing. The main content area has a title 'Analysis' and a 'Description' section containing a numbered list:

1. Work on requirements and define:
 - a. database schema
 - b. store topic
 - i. products being sold
 - ii. categories

Below the description is a 'Attachments (1)' section showing a single file named 'database_schema.drawio' from 10 Aug 2023, 02:10 pm. A progress bar at the bottom indicates '50% Done'.

The right side of the screen displays the 'In Progress' tab for the issue, with detailed information including:

 - Assignee: Roberto Nacu (RN)
 - Priority: Highest
 - Labels: None
 - Sprint: Set up
 - Story point estimate: None
 - Reporter: Roberto Nacu (RN)

At the bottom right, there are 'Created 4 hours ago' and 'Updated 1 hour ago' timestamps, along with a 'Configure' button.

Attached completed database schema to Jira issue

The screenshot shows a Jira issue page for 'QS-39'. The top navigation bar includes 'QS-39 / QS-6' and a 'DRAFT' button. The main area has tabs for 'Activity' (selected), 'Comments' (highlighted in blue), and 'History'. A comment from 'RN Roberto Nacu' is shown, with a note that it was posted 'in 1 second'. The comment text is 'Topic → Fashion/Clothing'. Below this, under 'Main categories:', a numbered list of items is provided: 1. T-shirts, 2. Dresses, 3. Accessories, 4. Formal Wear, 5. Activewear, 6. Skirts, 7. Jackets, 8. Sweaters, 9. Jeans, and 10. Underwear. At the bottom of the comments section are 'Edit · Delete' buttons and a user icon. To the right, a sidebar displays 'In Progress' status with a count of 1, and an 'Actions' dropdown. The 'Details' panel shows the assignee as 'RN Roberto Nacu', priority as 'Highest', labels as 'None', sprint as 'Set up', story point estimate as 'None', and reporter as 'RN Roberto Nacu'. It also shows creation and update times: 'Created 4 hours ago' and 'Updated 4 minutes ago'. A 'Configure' button is at the bottom of the sidebar.

Attached main categories to Jira issue

QS-39 / QS-6

database_schema.drawio
10 Aug 2023, 02:10 pm

Child issues Order by ... + 100% Done

qs-85 Design database schema RN DONE

qs-86 Decide store topic RN DONE

Activity Show: All Comments History Newest first ↓

RN Add a comment...

Pro tip: press M to comment

RN Roberto Nacu in 2 seconds
Store name → Fashionable
Edit · Delete · ⏲

RN Roberto Nacu 3 minutes ago
Topic → Fashion/Clothing
Main categories:
1. T-shirts
Picked store name (Fashionable)

This screenshot shows a Jira issue page for 'QS-39'. At the top, there are two tabs: 'QS-39' (with a plus icon) and 'QS-6' (with a checkmark icon). Below the tabs is a preview of a 'database_schema.drawio' file from 10 Aug 2023, 02:10 pm. The main content area has a green header bar with 'Child issues' on the left and 'Order by' dropdown on the right. A progress bar indicates '100% Done'. Two child issues are listed: 'qs-85 Design database schema' and 'qs-86 Decide store topic', both marked as 'DONE' with status indicators 'RN' and green 'DONE' buttons. Below this is a section titled 'Activity' with tabs for 'All', 'Comments' (which is selected), and 'History'. It shows a comment from 'Roberto Nacu' (RN) about picking a store name ('Fashionable') and setting the topic ('Fashion/Clothing'). The comment was made 3 minutes ago. A note says 'Main categories: 1. T-shirts' and 'Picked store name (Fashionable)'. On the right side of the screen, there are vertical icons for lock, trash, and other actions.

SDLC-3: Design

Projects / QA Store

Backlog

Set up 10 Aug – 10 Aug (4 issues)

- QG-5 Requirements gathering
- QG-50 Risk assessment
- QG-6 Analysis
- QS-87 Design

0 0 0 Complete sprint ...

SDLC DONE ✓ RN
SDLC DONE ✓ RN
SDLC IN PROGRESS ✓ RN
SDLC IN PROGRESS ✓ RN

+ Create issue

Projects / QA Store

Set up

Setting up the project: defining user stories and tasks using appropriate industry standards, setting up any risk assessment logging, completing analysis and design section of the SDLC (mainly identifying the database schema)

0 days remaining Complete sprint ...

GROUP BY None Insights

TO DO	IN PROGRESS	REVIEW
	MAX 2 Design SDLC <input checked="" type="checkbox"/> QS-87	DONE 3 ✓ Requirements gathering SDLC SDLC <input checked="" type="checkbox"/> QG-5 ✓ RN Risk assessment SDLC <input checked="" type="checkbox"/> QG-50 ✓ RN Analysis SDLC <input checked="" type="checkbox"/> QG-6 ✓ RN

Projects / QA Store

Backlog

Set up 10 Aug – 10 Aug (4 issues)

- QG-5 Requirements gathering
- QG-50 Risk assessment
- QG-6 Analysis
- QS-87 Design

0 0 0 Complete sprint ...

QS Sprint 1 11 Aug – 11 Aug (3 issues)

- QS-1 Daily stand-up meeting 1
- QS-78 Development: code
- QS-79 Development: testing

0 0 0 Start sprint ...

+ Create issue

QS-87 Design

Attach Add a child issue Link issue ...

Description Add a description... 0% Done

Child issues Order by 0% Done

- QS-88 Decide modules to be used
- QS-89 Decide architecture (classes, and how the...

Activity Show: All Comments History Newest first

Add a comment... Pro tip: press M to comment

Details

Assignee Roberto Nacu

Priority Highest

Labels None

Sprint Set up

Story point estimate None

Reporter Roberto Nacu

Created design backlog item, and added to current sprint - objective is to pick the right technologies, and design architecture and modules

The screenshot shows the 'Issue types' configuration screen in Jira. At the top, it says 'Projects / QA Store / Project settings / Issue types'. A section titled 'Problem' is selected. Below it, a note says 'Bugs track problems or errors.' There are two main sections: 'Description fields' and 'Context fields'. Under 'Description fields', there are four required fields: 'Summary' (Short text), 'Timestamp' (Time stamp), 'Date' (Date), and 'Description' (Text). Under 'Context fields', there are three optional fields: 'Status' (Status), 'Assignee' (Assignee), and 'Sprint' (Sprint). On the right side, there is a sidebar titled 'Fields' with a grid of field types: Short text, Paragraph, Date, Number, Time stamp, Labels, Dropdown, Checkbox, People, and Dependent dropdown. Below the grid is a search bar for 'Search all fields' and a list of 'Suggested fields' including 'Atlas project key', 'Atlas project status', 'Due date', 'Likelihood', and 'Can't find a field? Go to custom fields'.

Created Problem issue type just to track bugs/problems. Included date and timestamp fields.

Architecture

1. Presentation layer
 - a. The user interface
2. Application layer
 - a. Business logic, including processing orders, managing customers and items, and calculating costs
3. Data layer
 - a. Data storage and retrieval, interacting with the database

Modules

1. Front-end
 - a. User interface and user experience
 - b. Online store-front with all the needed pages according to the set requirements
 - c. HTML, CSS, JS
2. Back-end
 - a. Business logic and processes
 - b. RESTful APIs for communication with the front-end

c. Flask, Python

3. Database

a. Persistent data

b. MySQL

4. Unit and integration testing

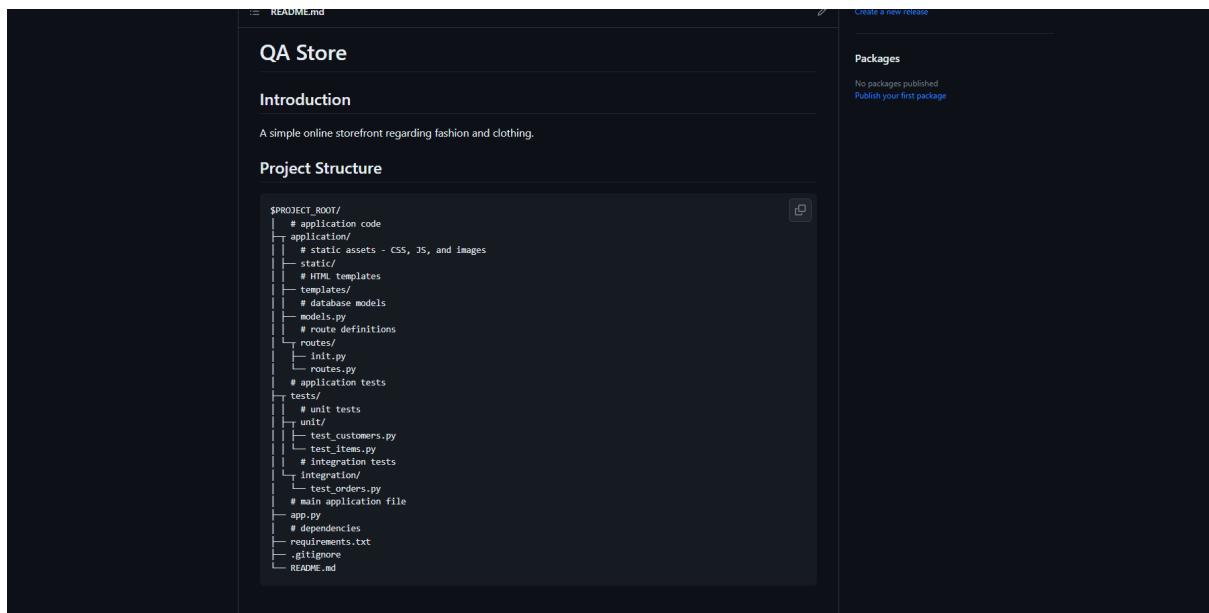
a. Validate the functionality of the application

b. `pytest` and `flask-testing`

5. Version Control + CI

a. Git and GitHub

b. Jenkins



The screenshot shows a GitHub repository interface for a project named "QA Store". The repository has a README.md file. The "Project Structure" section displays the following directory tree:

```
$PROJECT_ROOT/
    # application code
    application/
        # static assets - CSS, JS, and images
        static/
        templates/
            # HTML templates
            templates/
                # database models
                models.py
                # route definitions
                routes/
                    init.py
                    routes.py
                # application tests
                tests/
                    # unit tests
                    unit/
                        test_customers.py
                        test_items.py
                    # integration tests
                    integration/
                        test_orders.py
                    # main application file
                    app.py
    # dependencies
    requirements.txt
    .gitignore
    README.md
```

The right sidebar shows "Packages" with a message: "No packages published" and "Publish your first package".

I planned the entire project structure while taking in consideration all the modules and the overall architecture previously defined

QS-39 / QS-6

Analysis

[Attach](#) [Add a child issue](#) [Link issue](#) [...](#)

Description

1. Work on requirements and define:
 - a. database schema
 - b. store topic
 - i. products being sold
 - ii. categories

Attachments (2)

database_schema... kup.png
10 Aug 2023, 03:27 pm
Preview unavailable
database_schema.drawio
10 Aug 2023, 02:10 pm

Child issues Order by [...](#) + 100% Done

Done [Done](#) [Actions](#)

Details

Assignee Roberto Nacu

Priority Highest

Labels None

Sprint Set up

Story point estimate None

Reporter Roberto Nacu

Created 5 hours ago [Configure](#)

Added database schema screenshot to Jira issue

QS-39 / QS-87

Design

[Attach](#) [Add a child issue](#) [Link issue](#) [...](#)

Description

Add a description...

Attachments (2)

architecture_modules.png
10 Aug 2023, 03:22 pm
project_structure.png
10 Aug 2023, 03:21 pm

Child issues Order by [...](#) + 50% Done

In Progress [Actions](#)

Details

Assignee Roberto Nacu

Priority Highest

Labels None

Sprint Set up

Story point estimate None

Reporter Roberto Nacu

Created 1 hour ago [Configure](#)

Pro tip: press **M** to comment

Added architecture and project structure screenshots to Jira issue

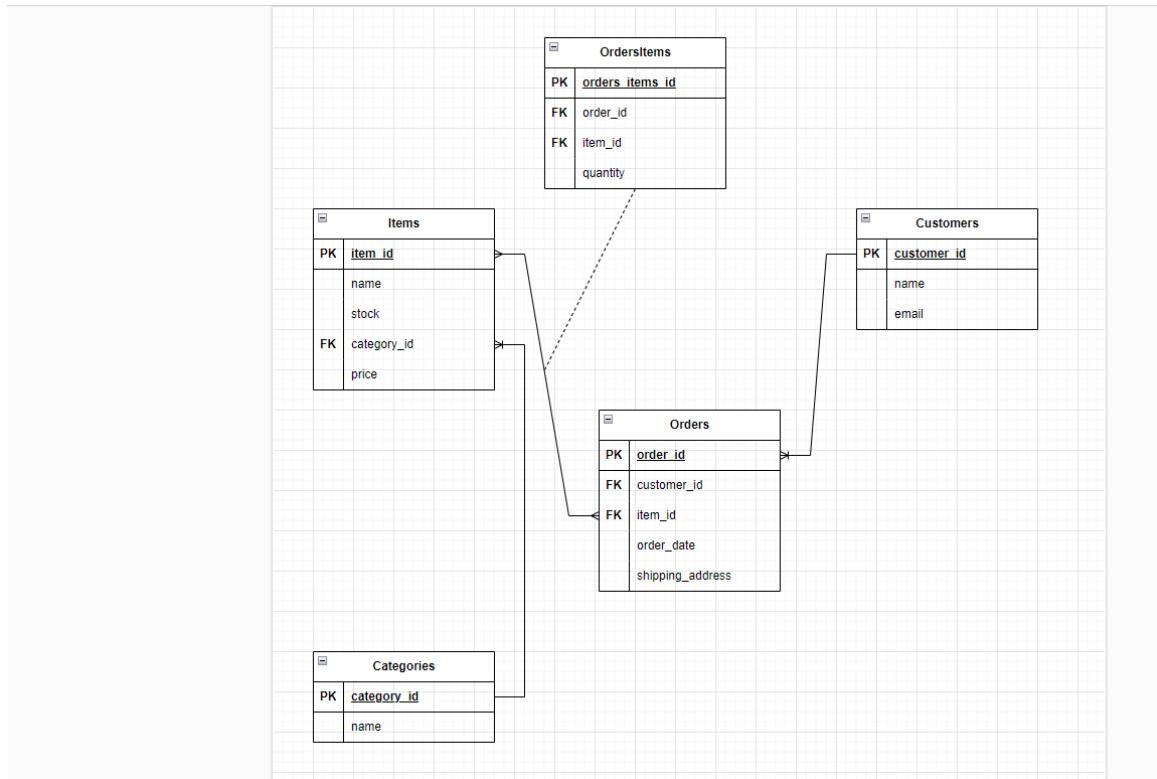
Classes definition

Given the initial brief, as well as the work done during the database schema/ER UML creation, we can mock up five initial classes:

Customers

Field	Data type	Constraints
customer_id	INT	PK, AUTO_INCREMENT
name	VARCHAR(255)	NOT NULL
email	VARCHAR(255)	NOT NULL, UNIQUE

Problem: During the creation of the Item class, I noticed that we actually do not track in any way the price of the item.



Updated UML ER

The screenshot shows two Jira issue pages side-by-side. The left page displays issue QS-39, titled 'Analysis'. It includes sections for 'Description' (a list of requirements), 'Attachments' (two files: 'database_schema.drawio' and 'database_schem... kup.png'), and 'Child issues' (empty). The right page shows issue QS-6, which is marked as 'Done'. The 'Details' panel lists fields such as 'Assignee' (Roberto Nacu), 'Priority' (Highest), 'Labels' (None), 'Sprint' (Set up), 'Story point estimate' (None), and 'Reporter' (Roberto Nacu). The status bar at the bottom indicates 'Created 5 hours ago' and 'Updated 27 minutes ago'.

Also updated attached files to related Jira issue

Items

Field	Data type	Constraints
item_id	INT	PK, AUTO_INCREMENT
name	VARCHAR(255)	NOT NULL
stock	INT	NOT NULL, DEFAULT 0
category_id	INT	FK, NOT NULL
price	FLOAT	NOT NULL
filename	VARCHAR(500)	NOT NULL

Orders

Field	Data type	Constraints
order_id	INT	PK, AUTO_INCREMENT
customer_id	INT	FK, NOT NULL
order_date	DATETIME	NOT NULL
shipping_address	VARCHAR(255)	NOT NULL

Categories

Field	Data type	Constraints
category_id	INT	PK, AUTO_INCREMENT
name	VARCHAR(255)	NOT NULL

OrdersItems

Field	Data type	Constraints
orders_items_id	INT	PK, AUTO_INCREMENT
order_id	INT	FK, NOT NULL
item_id	INT	FK, NOT NULL
quantity	INT	NOT NULL

The screenshot shows the Jira interface with the following sections:

- Backlog:** A list of issues under "Set up" sprint (10 Aug – 10 Aug, 5 issues). One issue, "QS-87 Design", is selected and highlighted in blue.
- Design:** A detailed view of the selected issue QS-87. It includes a description placeholder, attachments (5 files: classes.definition.1.png, classes.definition.2.png, classes.definition.3.png, architecture_model.pdf), and a child issues section with two items: "QS-88 Decide modules to be used" and "QS-89 Decide architecture (classes, and how they will ...)".
- In Progress:** A sidebar showing the issue details: Assignee (Roberto Nacu), Priority (Highest), Labels (None), Sprint (Set up), Story point estimate (None), Reporter (Roberto Nacu), and a comment input field.

Added classes definition to Design Jira issue

The screenshot shows a Jira issue titled "Design".

Top Bar: QS-39 / QS-87, Lock icon, 1 comment, Like icon, Share icon, More options, Close icon.

Issue Title: Design

Buttons: Attach, Add a child issue, Link issue, More options.

Description: Add a description...

Attachments (5):

- classes_definition_3.png (10 Aug 2023, 04:02 pm)
- classes_definition_2.png (10 Aug 2023, 04:02 pm)
- classes_definition_1.png (10 Aug 2023, 04:02 pm)
- architecture_modu... (10 Aug 2023, 03:22)

Child issues:

Issue Key	Description	Assignee	Status
QS-88	Decide modules to be used	RN	DONE
QS-89	Decide architecture (classes, and how they will interact)	RN	DONE
QS-93	Create project structure within the local repo	RN	IN PROGRESS

Details Panel:

- Assignee:** Roberto Nacu (RN)
- Priority:** Highest
- Labels:** None
- Sprint:** Set up
- Story point estimate:** None
- Reporter:** Roberto Nacu (RN)
- Created:** 2 hours ago
- Updated:** 4 minutes ago
- Resolved:** 4 minutes ago
- Configure:** gear icon

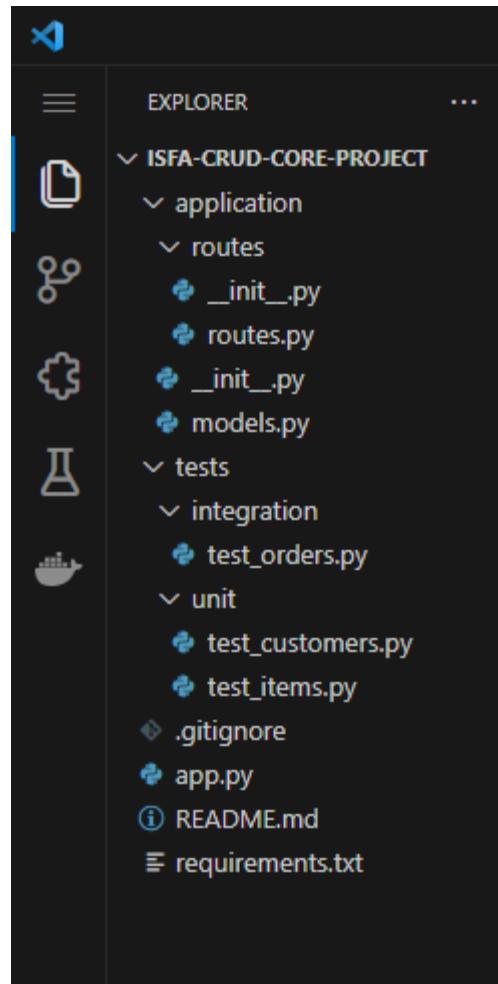
Added additional issue to Design issue

The screenshot shows a dark-themed interface for a code editor or file browser. On the left, there is a tree view of a project structure. The root directory is labeled '\$PROJECT_ROOT/'. Inside it, there is a folder named 'application/' which contains several files and sub-folders: 'init.py', 'static/' (containing 'css', 'js', and 'images'), 'templates/' (containing 'html'), 'models.py', 'routes/' (containing 'init.py' and 'routes.py'), and 'tests/' (containing 'unit/' which has 'test_customers.py' and 'test_items.py'). There is also an 'integration/' folder with 'test_orders.py'. In addition to these, there are 'app.py', 'requirements.txt', '.gitignore', and 'README.md'. A small clipboard icon is located in the top right corner of the interface.

```
$PROJECT_ROOT/
|   # application code
|   application/
|   |   init.py
|   |   # static assets - CSS, JS, and images
|   |   static/
|   |   |   css
|   |   |   js
|   |   |   images
|   |   templates/
|   |   |   html
|   |   models.py
|   |   routes/
|   |   |   init.py
|   |   |   routes.py
|   |   tests/
|   |   |   unit/
|   |   |   |   test_customers.py
|   |   |   |   test_items.py
|   |   |   integration/
|   |   |   |   test_orders.py
|   |   app.py
|   |   requirements.txt
|   .gitignore
|   README.md
```

Modified mock-up project structure (simply added `__init__.py` in application folder (also updated in Design Jira issue))

SDLC-4: Code



Created project structure within the dev branch

```

.gitignore | C:\Users\rober\Documents\MEGA\work\AMS\ISFA-CRUD-core-project
roberfahru MINGW64 ~/Documents/MEGA/work/AMS/ISFA-CRUD-core-project (dev)
$ git status
On branch dev
Your branch is up to date with 'origin/dev'.

Changes not staged for commit:
  (use "git add <file>..." to update what will be committed)
  (use "git restore <file>..." to discard changes in working directory)
    modified:   .gitignore

no changes added to commit (use "git add" and/or "git commit -a")
roberfahru MINGW64 ~/Documents/MEGA/work/AMS/ISFA-CRUD-core-project (dev)
$ git add .gitignore

roberfahru MINGW64 ~/Documents/MEGA/work/AMS/ISFA-CRUD-core-project (dev)
$ git commit -m "RN: uploaded gitignore template"
[dev 9d0e10b] RN: uploaded gitignore template
  1 file changed, 160 insertions(+)
roberfahru MINGW64 ~/Documents/MEGA/work/AMS/ISFA-CRUD-core-project (dev)
$ [REDACTED]

```

Uploaded .gitignore template

The screenshot shows the VS Code interface with a dark theme. The Explorer sidebar on the left lists a project named 'ISFA-CRUD-CORE-PROJECT' containing an 'application' folder with files like __init__.py, routes.py, and models.py, and a 'tests' folder with test_orders.py, test_customers.py, and test_items.py. The terminal window at the bottom shows a user's session on a MINGW64 system, running commands to activate a virtual environment (venv), update pip, and list packages. The terminal output indicates a pip upgrade from 23.1.2 to 23.2.1.

Create virtual environment and set it up as default interpreter within VSC

The screenshot shows a Jira backlog board. At the top, there is a search bar and filters for 'RN' (Ready to Now). Below the header, a section titled 'Set up' (10 Aug – 10 Aug (5 issues)) is shown, with a note about defining user stories and tasks. Below this, several issues are listed with their status: Requirements gathering (SDLC, DONE), Risk assessment (SDLC, DONE), Analysis (SDLC, DONE), Design (SDLC, DONE), and Price field missing in Item (PROBLEMS, DONE). A 'Create issue' button is visible. Below the backlog, two sprints are listed: 'QS Sprint 1' (11 Aug – 11 Aug (3 issues)) and 'QS Sprint 2' (14 Aug – 15 Aug (2 issues)). Each sprint has a 'Start sprint' button and a 'Review' section with 'TO DO' items.

16:33: Set up sprint is completed, however, these issues are continuously reiterated over due to the nature of an Agile workflow. I did put them in a sprint within Jira but that was mainly to keep track of things during day 1

Projects / QA Store

Backlog

Continuous Iteration 10 Aug – 15 Aug (5 issues)

Previously named "Set up": Setting up the project: defining user stories and tasks using appropriate industry standards, setting up any risk assessment logging, completing analysis and design section of the SDLC (mainly identifying the database schema)

Issue	Labels	Status
QS-5 Requirements gathering	SDLC	IN PROGRESS
QS-50 Risk assessment	SDLC	IN PROGRESS
QS-6 Analysis	SDLC	IN PROGRESS
QS-87 Design	SDLC	IN PROGRESS
QS-94 Price field missing in Item	PROBLEMS	DONE

+ Create issue

Exactly for the above reason, I changed the sprint's date to the end of the available time, turned the issues into in progress, and changed the sprint's name from Set up to Continuous Iteration

QS Sprint 1 11 Aug – 11 Aug (2 issues)

QS-1 Daily stand-up meeting 1

QS-78 Development: code

+ Create issue

QS Sprint 2 14 Aug – 15 Aug (3 issues)

QS-2 Daily stand-up meeting 2

QS-79 Development: testing

QS-3 Daily stand-up meeting 3

+ Create issue

Moved Development: testing from QS Spring 1 to QS Spring 2

ISFA-CRUD-core-project

requirements.txt C:\Users\rober\Documents\MEGA\work\AMS\ISFA-CRUD-core-project

application
 > routes
 > __init__.py
 > models.py
tests
 > integration
 > test_orders.py
 > unit
 > test_customers.py
 > test_items.py
 > venv
 > .gitignore
 > app.py
README.md
requirements.txt

```
1 blinker==1.6.2
2 click==8.1.6
3 colorama==0.4.6
4 Flask==2.3.2
5 itsdangerous==2.1.2
6 Jinja2==3.1.2
7 MarkupSafe==2.1.3
8 Werkzeug==2.3.6
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL

```
[venv]
rober@shizu MINGW64 ~/Documents/MEGA/work/AMS/ISFA-CRUD-core-project (dev)
$ pip freeze > requirements.txt
[venv]
rober@shizu MINGW64 ~/Documents/MEGA/work/AMS/ISFA-CRUD-core-project (dev)
$
```

Using a requirements.txt file to keep track of dependencies. For now, only Flask has been installed

```

requirements.txt | C:\Users\rober\Documents\MEGA\work\VAMS\ISFA-CRUD-core-project
-----
1 blinker==1.6.2
2 click==8.1.6
3 colorama==0.4.6
4 Flask==2.3.2
5 Flask-SQLAlchemy==3.0.5
6 greenlet==2.0.2
7 itsdangerous==2.1.2
8 Jinja2==3.1.2
9 MarkupSafe==2.1.3
10 python-dotenv==1.0.0
11 SQLAlchemy==2.0.19
12 typing_extensions==4.7.1
13 Werkzeug==2.3.6
14

```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL

```

$ pip install python-dotenv
Collecting python-dotenv
  Using cached python_dotenv-1.0.0-py3-none-any.whl (19 kB)
Installing collected packages: python-dotenv
Successfully installed python-dotenv-1.0.0
[rober@ahzu MINGW64 ~/Documents/MEGA/work/VAMS/ISFA-CRUD-core-project (dev)]
$ pip freeze > requirements.txt
[rober@ahzu MINGW64 ~/Documents/MEGA/work/VAMS/ISFA-CRUD-core-project (dev)]
$ 

```

> OUTLINE > TIMELINE

Installed flask-sqlalchemy and python-dotenv, in order to manage environment variables and all the database related work

```

app.py | C:\Users\rober\Documents\MEGA\work\VAMS\ISFA-CRUD-core-project
-----
1 from application import app
2 :
3 :
4 if __name__ == "__main__":
5     app.run(debug=True, host="0.0.0.0", port=5000)
6

```

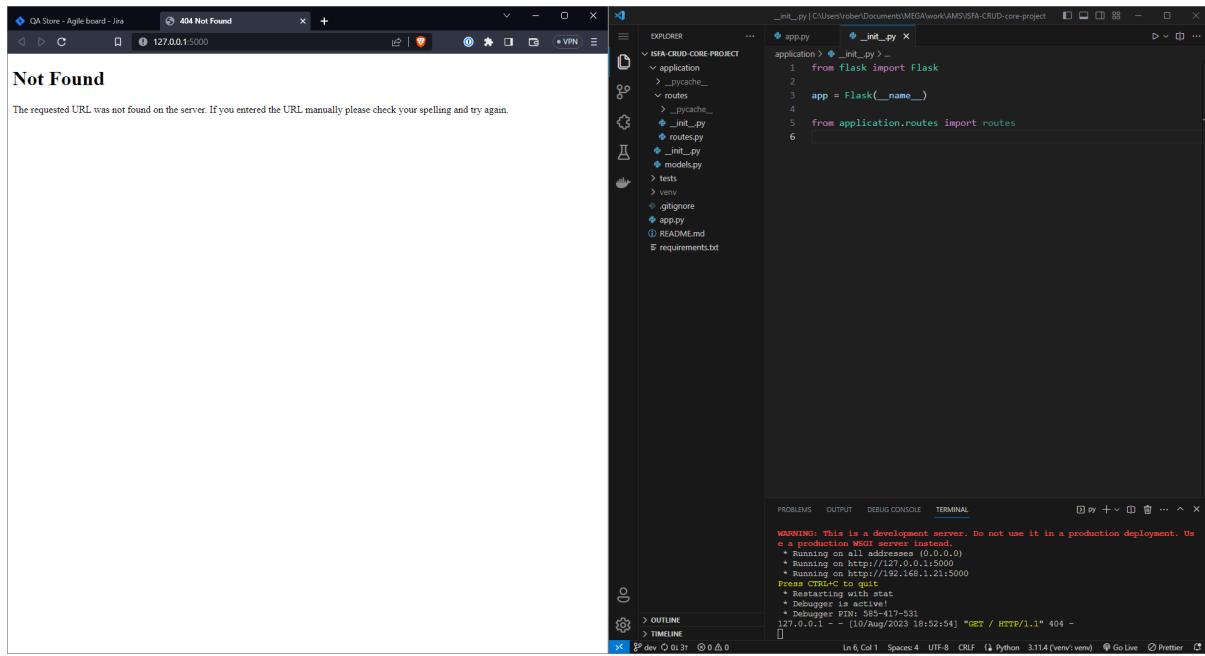
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL

```

e a production WSGI server instead.
+ Running on local development server (0.0.0.0:5000)
+ Running on http://127.0.0.1:5000
+ Running on http://192.168.1.21:5000
Press CTRL+C to quit
+ Reloads automatically
+ Debugger is active!
+ Debugger FIN: 585-417-531
127.0.0.1 - [10/Aug/2023 18:43:49] "GET / HTTP/1.1" 404 -
127.0.0.1 - [10/Aug/2023 18:43:49] "GET /favicon.ico HTTP/1.1" 404 -

```

> OUTLINE > TIMELINE



Created extremely simple Flask app. Will be used as the foundation for the entire work

Continuous Iteration 10 Aug – 15 Aug (7 issues)

- GS-5 Requirements gathering
- GS-50 Risk assessment
- GS-6 Analysis
- GS-87 Design
- GS-94 Price field missing in item
- GS-78 Development: code
- GS-79 Development: testing

QS Sprint 1 11 Aug – 11 Aug (1 issue)

- QS-1 Daily stand-up meeting 1

QS Sprint 2 14 Aug – 15 Aug (2 issues)

- QS-2 Daily stand-up meeting 2
- QS-3 Daily stand-up meeting 3

Moved code and testing parts of the SDLC to the Continuous Iteration sprint

The screenshot shows a Jira issue page for an epic titled "Development: code". The top navigation bar includes "QS-39 / QS-78", a lock icon, a progress indicator (1/1), a share icon, and a "..." button. A "Insights" button is also present.

The main content area has tabs for "Attach", "Add a child issue", "Link issue", and "...". Below these are sections for "Description" (with placeholder "Add a description...") and "Child issues".

The "Child issues" section lists three sub-tasks:

- QS-94 Create basic flask app: Status is DONE
- QS-95 Create basic MySQL database: Status is TO DO
- QS-96 Create basic page layout (for each EPIC page): Status is TO DO

Below the child issues is a "What needs to be done?" input field containing "Choose an existing issue" with "Create" and "Cancel" buttons. The "Activity" section shows "All" selected, "Comments" highlighted, and "History". It also includes a "Newest first" sorting option and a "Add a comment..." input field.

The right sidebar displays the "In Progress" status, a "Actions" dropdown, and detailed information about the epic:

- Assignee: Roberto Nacu (RN)
- Priority: Highest
- Labels: None
- Sprint: Continuous Iteration
- Story point estimate: None
- Reporter: Roberto Nacu (RN)

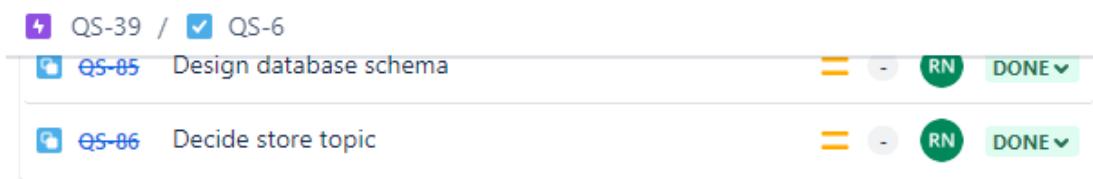
At the bottom of the sidebar, it shows "Created 8 hours ago" and "Updated 5 minutes ago", along with a "Configure" button.

Added three sub-tasks to code issue which will set up the foundation for the entire coding

The screenshot shows the Visual Studio Code interface with the following details:

- EXPLORER** view: Displays the project structure under "ISFA-CRUD-C...". The "application" folder is expanded, showing subfolders like __pycache__, routes, static, templates, and files like __init__.py, models.py, tests, venv, .gitignore, app.py, README.md, and requirements.txt.
- ROUTER** view: Shows the current file path: application > routes > routes.py. The file contains the number "1".
- TERMINAL** view: Shows a terminal session in a bash environment. The prompt is "(venv)" followed by "rober@shizu MINGW64 ~/Documents/MEGA/work/AMS/ISFA-CRUD-core-project (dev)".

Added static and templates folders (specified in project structure but forgot to create them)



Activity

Show: All Comments History

Newest first ↓



Add a comment...

Pro tip: press **M** to comment



Roberto Nacu 7 hours ago

Store name → Fashionable

Edit · Delete ·



Roberto Nacu 7 hours ago Edited

Topic → Fashion/Clothing

Main categories:

1. T-shirts
2. Dresses
3. Skirts
4. Sweaters
5. Jeans

Edit · Delete ·

Edited store categories. Simplified from 10 categories to only 5. There will be two products for each category. Main audience is women.

The screenshot shows the Jira Software interface with the 'Backlog' view. On the left, there's a sidebar with filters for 'RN', 'Epic', 'Label', and 'Type'. Below it, a section titled 'Continuous Iteration' lists several issues under 'IN PROGRESS' status, including 'QS-5 Requirements gathering', 'QS-50 Risk assessment', and 'QS-6 Analysis'. A note indicates this was previously named 'Set up'. Below this is a section for 'QS Sprint 1' (11 Aug – 11 Aug) containing one issue: 'QS-1 Daily stand-up meeting' in 'TO DO' status. At the bottom, there's a section for 'QS Sprint 2' (14 Aug – 15 Aug) with two issues: 'QS-7 Design' and 'QS-94 Price field missing in item' both in 'IN PROGRESS' status.

QS-6 Analysis

- QS-5 Requirements gathering (IN PROGRESS)
- QS-50 Risk assessment (IN PROGRESS)
- QS-6 Analysis (IN PROGRESS)
- QS-87 Design (IN PROGRESS)
- QS-94 Price field missing in item (PROBLEMS)
- QS-78 Development: code (IN PROGRESS)
- QS-79 Development: testing (IN PROGRESS)

QS Sprint 1 11 Aug – 11 Aug (1 issue)

- QS-1 Daily stand-up meeting (TO DO)

QS Sprint 2 14 Aug – 15 Aug (2 issues)

- QS-7 Design (IN PROGRESS)
- QS-94 Price field missing in item (IN PROGRESS)

Added new sub-task to Analysis issue: finding product images

The screenshot shows the Jira issue detail page for 'QS-6 Analysis'. At the top, there are two attachments: 'database_schema.drawio' (10 Aug 2023, 03:54 pm) and 'database_schema.kup.png' (10 Aug 2023, 03:54 pm).

Child issues

Order by: 100% Done

qs-85 Design database schema	DONE
qs-86 Decide store topic	DONE
qs-98 Find product images	DONE

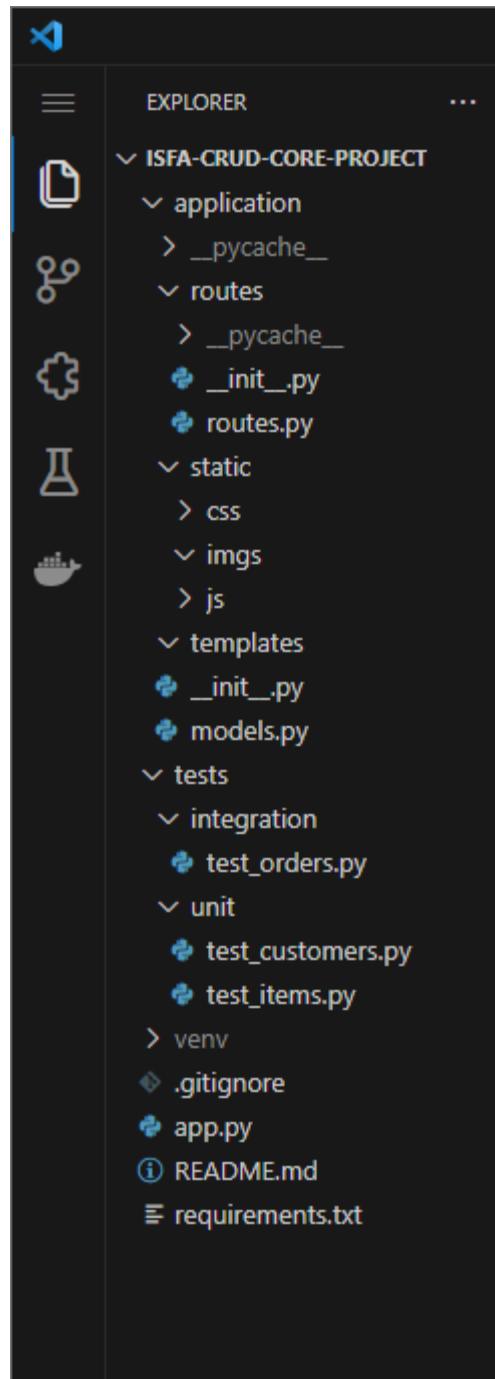
Activity

Show: All Comments History Newest first ↴

RN Add a comment...
Pro tip: press **M** to comment

RN Roberto Nacu in 3 seconds
[k E-commerce Product Images](#)
Edit · Delete · ⚙

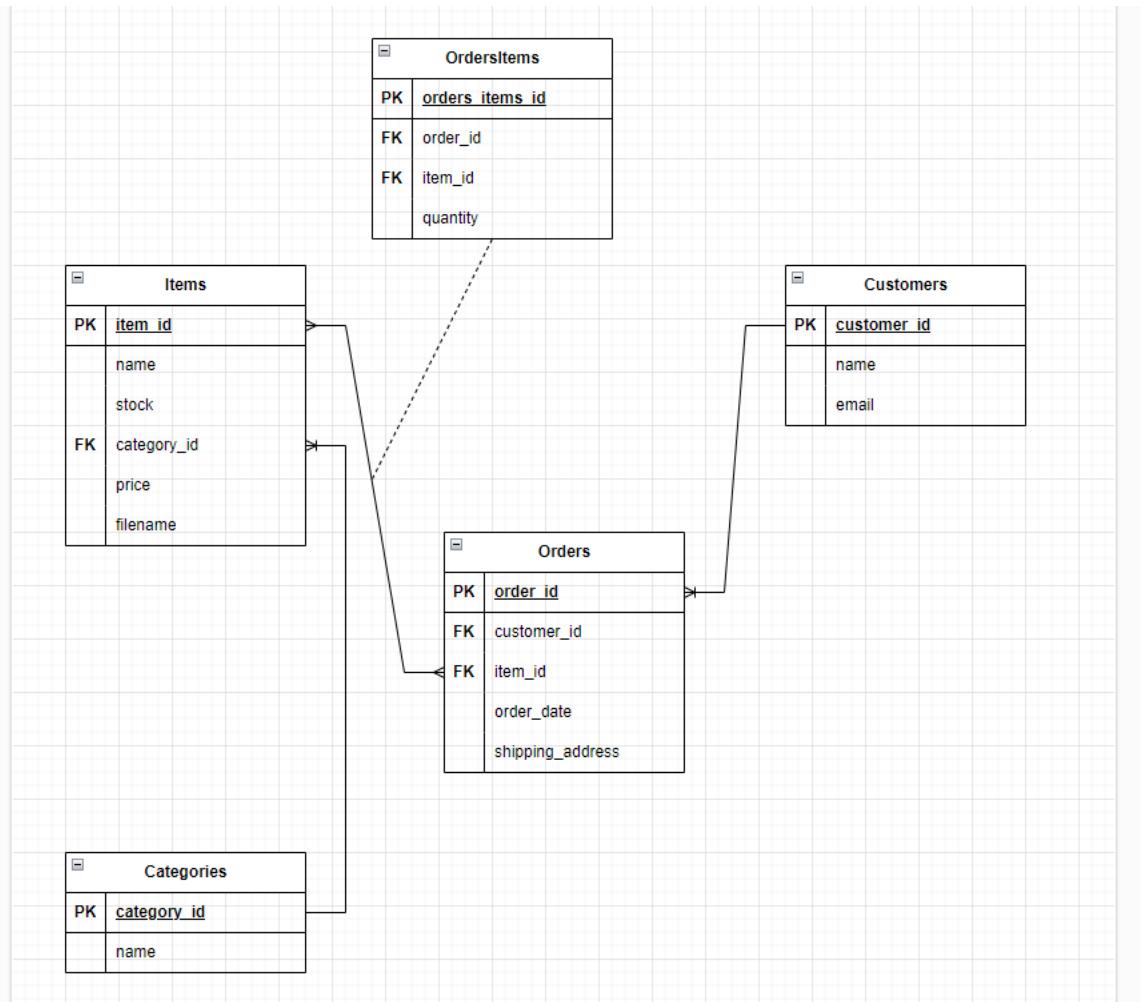
RN Roberto Nacu 7 hours ago
Updated sub-task



Split static folder into css, js, and imgs

Problem: How do we store images? We should not store them in the database, but how will items be related to images in the static/ folder?

- Each item should have an additional field (i.e., filename) specifying the file name of the related image saved in `static/imgs/`



Updated ER UML diagram, included a filename in Items

QS-39 / QS-6

Attachments (2)

database_schema... kup.png
10 Aug 2023, 09:27 pm

database_schema.drawio
10 Aug 2023, 09:27 pm

Child issues

Order by ... + 100% Done

QS-85 Design database schema (RN DONE)

QS-86 Decide store topic (RN DONE)

QS-90 Find product images (RN DONE)

Activity

Show: All Comments History Newest first ↴

RN Add a comment...

Pro tip: press M to comment

Roberto Nacu in 4 seconds
Updated ER UML diagram: added filename field to Items

Edit · Delete · ⚙

In Progress

Actions

Details

Assignee Roberto Nacu

Priority Highest

Labels None

Sprint Continuous Iteration

Story point estimate None

Reporter Roberto Nacu

Created 11 hours ago
Updated 26 minutes ago

⚙ Configure

The screenshot shows a Jira issue page for 'QS-39'. The left panel displays attachments ('database_schema... kup.png' and 'database_schema.drawio') and a list of child issues ('QS-85', 'QS-86', 'QS-90') all marked as 'DONE'. The right panel shows detailed information about the issue, including assignee (Roberto Nacu), priority (Highest), and sprint (Continuous Iteration). A comment from Roberto Nacu is visible in the activity feed, mentioning an update to an ER UML diagram. The sidebar also shows creation and update times.

Added comment and updated attachments in Analysis issue on Jira

QS-87 / QS-39

Child issues

- QS-88 Decide modules to be used (RN, DONE)
- QS-89 Decide architecture (classes, and how they will interact) (RN, DONE)
- QS-90 Create project structure within the local repo (RN, DONE)

Activity

Show: All Comments History Newest first

RN Add a comment...
Pro tip: press M to comment

RN Roberto Nacu in 4 seconds
Updated Items class specification: added filename field
Edit · Delete · ⚙️

In Progress

Actions

Details

Assignee: Roberto Nacu (RN)
Priority: Highest
Labels: None
Sprint: Continuous Iteration
Story point estimate: None
Reporter: Roberto Nacu (RN)
Created 7 hours ago
Updated 4 hours ago
⚙️ Configure

Items

Field	Data type	Constraints
<u>item_id</u>	INT	PK, AUTO_INCREMENT
name	VARCHAR(255)	NOT NULL
stock	INT	NOT NULL, DEFAULT 0
<u>category_id</u>	INT	FK, NOT NULL
price	FLOAT	NOT NULL
filename	VARCHAR(500)	NOT NULL

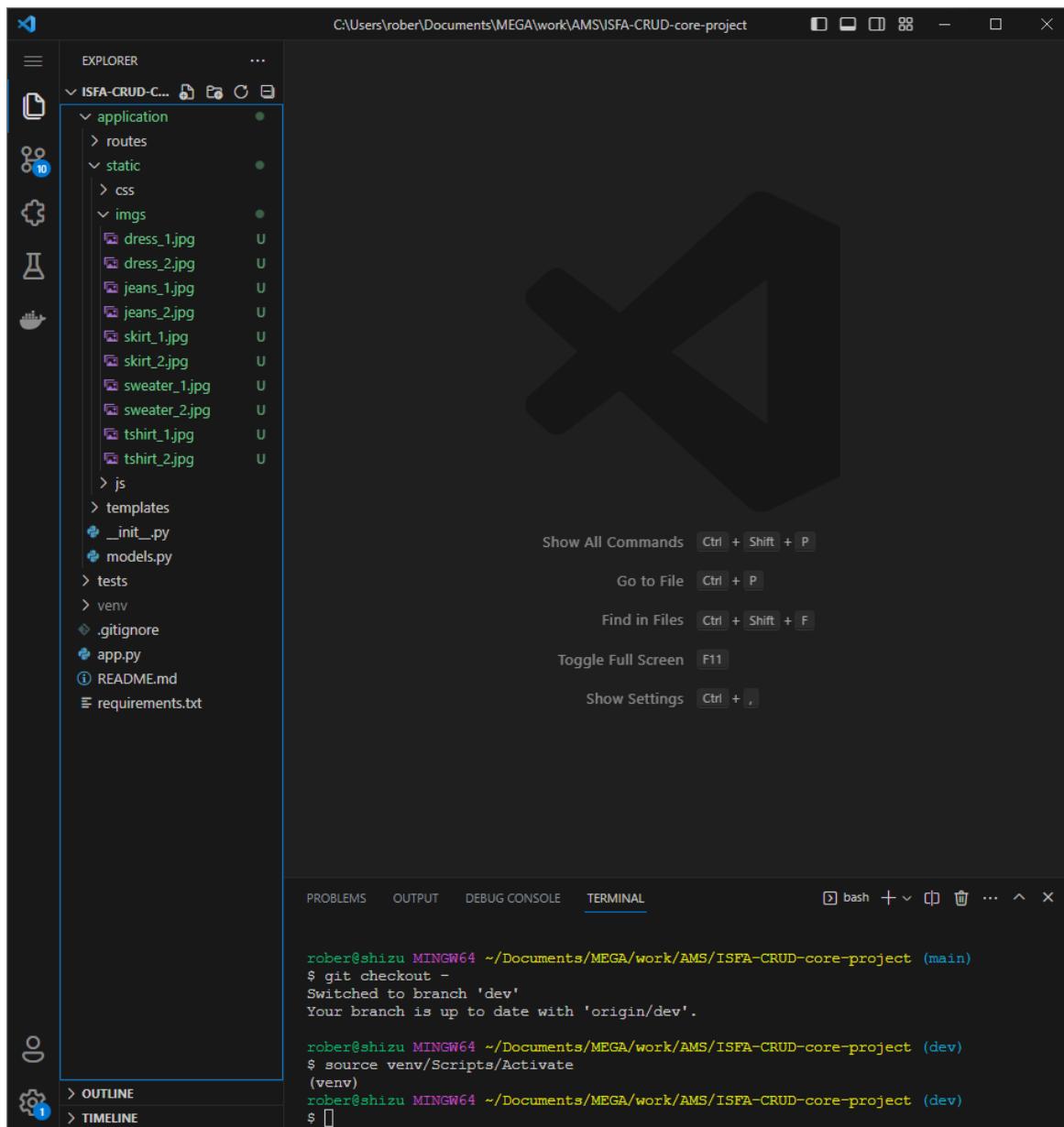
Orders

Field	Data type	Constraints
<u>order_id</u>	INT	PK, AUTO_INCREMENT
<u>customer_id</u>	INT	FK, NOT NULL
<u>item_id</u>	INT	FK, NOT NULL
<u>order_date</u>	DATETIME	NOT NULL
<u>shipping_address</u>	VARCHAR(255)	NOT NULL

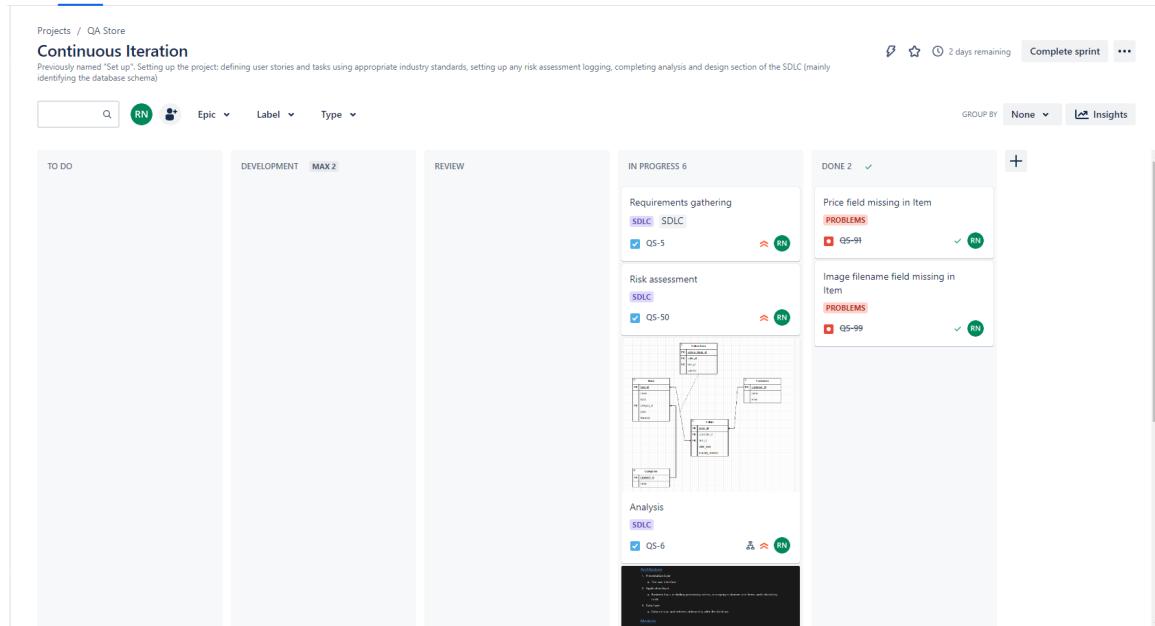
Categories

Field	Data type	Constraints
<u>category_id</u>	INT	PK, AUTO_INCREMENT
name	VARCHAR(255)	NOT NULL

Also updated class specification within Design Jira issue

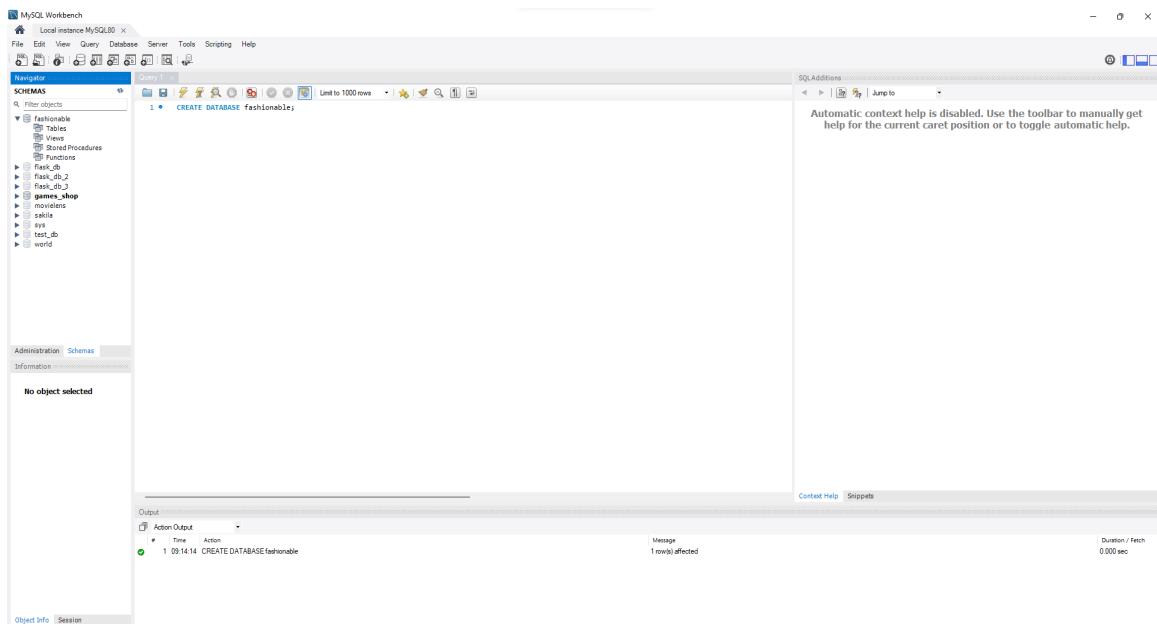


Added some product images

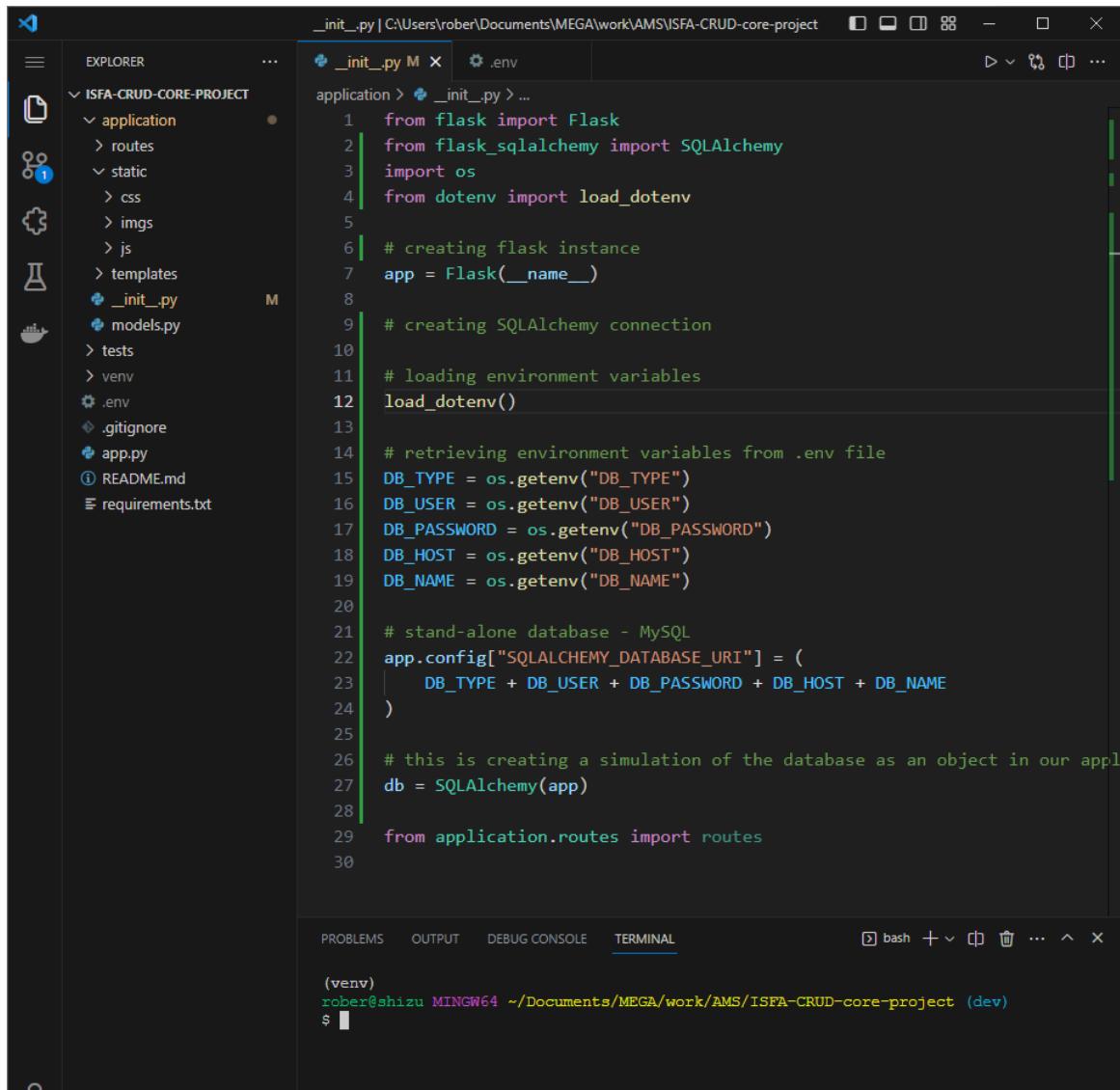


Reviewed Kanban board in order to keep track of development work on user stories/tasks. In progress column is pretty much for the main SDLC parts of the project which we continuously iterate over.

Completed daily stand-up meeting



Created database schema using MySQL Workbench



```
_init_.py | C:\Users\rober\Documents\MEGA\work\AMS\ISFA-CRUD-core-project
```

```
application > _init_.py > ...
1  from flask import Flask
2  from flask_sqlalchemy import SQLAlchemy
3  import os
4  from dotenv import load_dotenv
5
6  # creating flask instance
7  app = Flask(__name__)
8
9  # creating SQLAlchemy connection
10
11 # loading environment variables
12 load_dotenv()
13
14 # retrieving environment variables from .env file
15 DB_TYPE = os.getenv("DB_TYPE")
16 DB_USER = os.getenv("DB_USER")
17 DB_PASSWORD = os.getenv("DB_PASSWORD")
18 DB_HOST = os.getenv("DB_HOST")
19 DB_NAME = os.getenv("DB_NAME")
20
21 # stand-alone database - MySQL
22 app.config["SQLALCHEMY_DATABASE_URI"] = (
23     DB_TYPE + DB_USER + DB_PASSWORD + DB_HOST + DB_NAME
24 )
25
26 # this is creating a simulation of the database as an object in our app
27 db = SQLAlchemy(app)
28
29 from application.routes import routes
30
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL

(venv) rober@shizu MINGW64 ~/Documents/MEGA/work/AMS/ISFA-CRUD-core-project (dev)

Added MySQL database integration using sqlalchemy. Also used a .env file to store the database URI

```
requirements.txt
1  blinker==1.6.2
2  click==8.1.6
3  colorama==0.4.6
4  Flask==2.3.2
5  Flask-SQLAlchemy==3.0.5
6  greenlet==2.0.2
7  itsdangerous==2.1.2
8  Jinja2==3.1.2
9  MarkupSafe==2.1.3
10 PyMySQL==1.1.0
11 python-dotenv==1.0.0
12 SQLAlchemy==2.0.19
13 typing_extensions==4.7.1
14 Werkzeug==2.3.6
15
```

Updating dependencies, pymysql is required to work with MySQL via sqlalchemy

The screenshot shows the Visual Studio Code interface with a dark theme. The Explorer sidebar on the left displays the project structure for 'ISFA-CRUD-C...', which includes 'application', 'tests', 'venv', '.env', '.gitignore', 'app.py', 'create.py', 'README.md', and 'requirements.txt'. The 'create.py' file is open in the main editor area, showing the following code:

```
from application import app, db
# create the database schema or even just test db connection
with app.app_context():
    db.create_all()
```

The terminal at the bottom shows the command-line output of running the script:

```
(venv) rober@shizu MINGW64 ~/Documents/MEGA/work/AMS/ISFA-CRUD-core-project (dev)
$ py create.py
(venv)
rober@shizu MINGW64 ~/Documents/MEGA/work/AMS/ISFA-CRUD-core-project (dev)
$
```

Testing the database connection via a create.py file. No models have been created yet

The screenshot shows the VS Code interface with the following details:

- EXPLORER** sidebar: Shows the project structure under "ISFA-CRUD-C...". The "application" folder contains: __pycache__, routes, static, templates, __init__.py, models.py, tests, venv, .env, .gitignore, app.py, create.py, README.md, and requirements.txt.
- models.py** file content (tab bar):

```
1 from application import db
2
3
4 class Items:
5     pass
6
7
8 class Orders:
9     pass
10
11
12 class Categories:
13     pass
14
15
16 class OrdersItems:
17     pass
18
19
20 class Customers:
21     pass
22
```
- TERMINAL** tab:

```
(venv)
rober@shizu MINGW64 ~/Documents/MEGA/work/AMS/ISFA-CRUD-core-project (dev)
$ py create.py
(venv)
rober@shizu MINGW64 ~/Documents/MEGA/work/AMS/ISFA-CRUD-core-project (dev)
$ [ ]
```

Added models layout based on classes definition done during the Design step

QS-39 / QS-78

Development: code

In Progress

Description
Add a description...

Child issues

Issue	Summary	Status
QS-94	Create basic flask app	DONE
QS-95	Create basic MySQL database	DONE

Activity
Show: All Comments History Newest first ↴

Add a comment...
Pro tip: press M to comment

Details

- Assignee: Roberto Nacu (RN)
- Priority: Highest
- Labels: None
- Sprint: Continuous Iteration
- Story point estimate: None
- Reporter: Roberto Nacu (RN)

Deleted task to create a template for each page at this stage. Doing it would actually make everything really messy → this will be done based on work in progress based on the thing we are working on

```

1  from application import db
2
3
4  class Items:
5      item_id = db.Column(db.Integer, primary_key=True)
6      name = db.Column(db.String(255), nullable=False)
7      stock = db.Column(db.Integer, nullable=False, default=0)
8      category_id = db.Column(
9          db.Integer, db.ForeignKey("categories.category_id"), nullable=False
10     )
11     price = db.Column(db.Float, nullable=False)
12     filename = db.Column(db.String(255), nullable=False)
13
14
15 class Categories:
16     category_id = db.Column(db.Integer, primary_key=True)
17     name = db.Column(db.String(255), nullable=False)
18     items = db.relationship("Items", backref="category")
19

```

Created Items and Categories models, which are essential for the project

Problem: Customers emails must be unique.

→ Fixed definition

Customers		
Field	Data type	Constraints
customer_id	INT	PK, AUTO_INCREMENT
name	VARCHAR(255)	NOT NULL
email	VARCHAR(255)	NOT NULL, UNIQUE

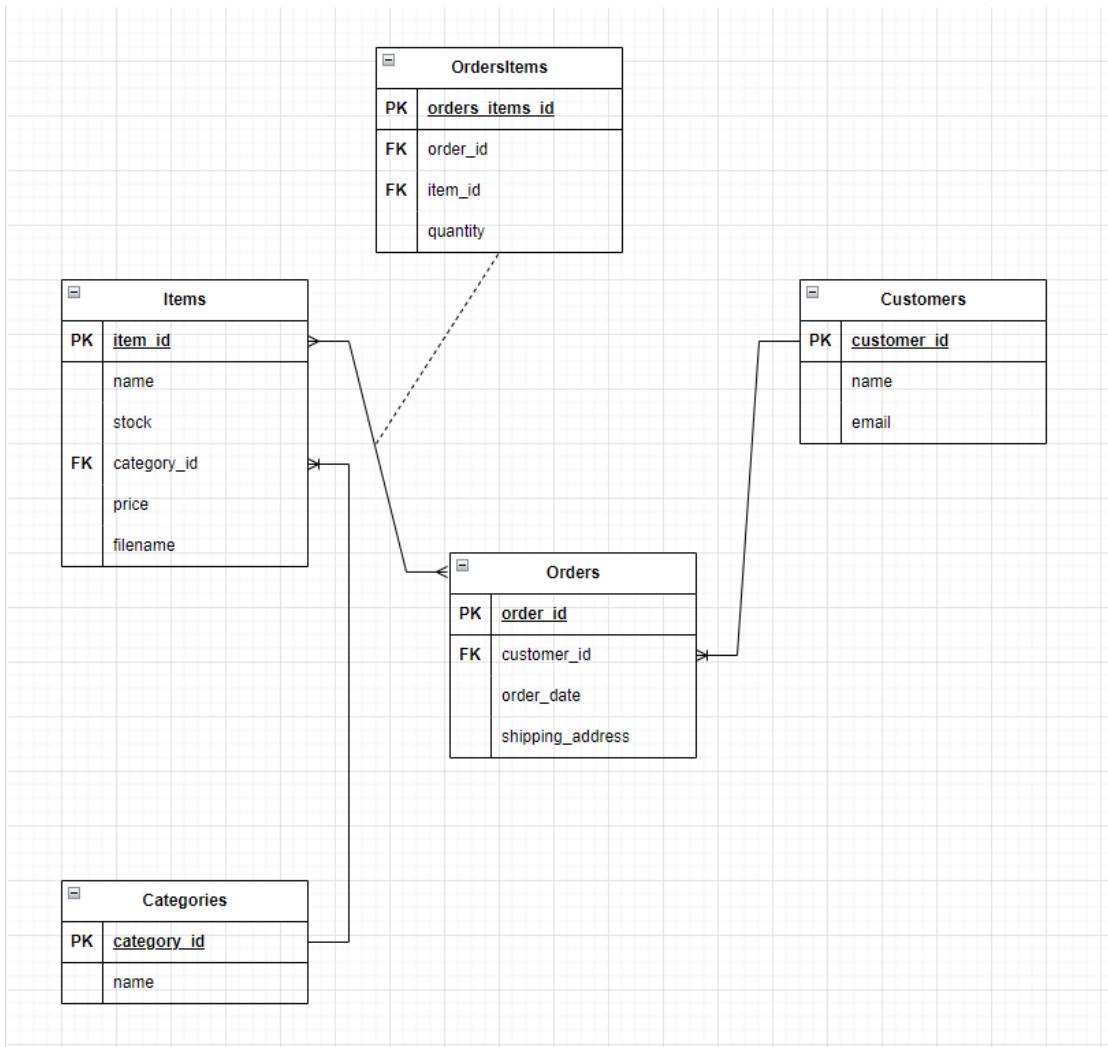
Problem: Orders table should not have item_id as a foreign key. This is because the relationship is many to many and we use an associative table that will store order_id and item_id.

Items		
Field	Data type	Constraints
item_id	INT	PK, AUTO_INCREMENT
name	VARCHAR(255)	NOT NULL
stock	INT	NOT NULL, DEFAULT 0
category_id	INT	FK, NOT NULL
price	FLOAT	NOT NULL
filename	VARCHAR(500)	NOT NULL

Orders		
Field	Data type	Constraints
order_id	INT	PK, AUTO_INCREMENT
customer_id	INT	FK, NOT NULL
order_date	DATETIME	NOT NULL
shipping_address	VARCHAR(255)	NOT NULL

Categories		
Field	Data type	Constraints
category_id	INT	PK, AUTO_INCREMENT
name	VARCHAR(255)	NOT NULL

Fixed class definition



Updated ER UML diagram

```
models.py | C:\Users\rober\Documents\MEGA\work\AMS\SFACRUD-core-project - X
routes.py  models.py M  create.py

application > models.py > OrdersItems
1  from application import db
2
3
4  class Items:
5      item_id = db.Column(db.Integer, primary_key=True)
6      name = db.Column(db.String(255), nullable=False)
7      stock = db.Column(db.Integer, nullable=False, default=0)
8      category_id = db.Column(
9          db.Integer, db.ForeignKey("categories.category_id"), nullable=False
10     )
11     price = db.Column(db.Float, nullable=False)
12     filename = db.Column(db.String(255), nullable=False)
13     orders_items = db.relationship("OrdersItems", backref="items")
14
15
16  class Categories:
17      category_id = db.Column(db.Integer, primary_key=True)
18      name = db.Column(db.String(255), nullable=False)
19      items = db.relationship("Items", backref="category")
20
21
22  class Customers:
23      customer_id = db.Column(db.Integer, primary_key=True)
24      name = db.Column(db.String(255), nullable=False)
25      email = db.Column(db.String(255), nullable=False, unique=True)
26      orders = db.relationship("Orders", backref="customer")
27
```

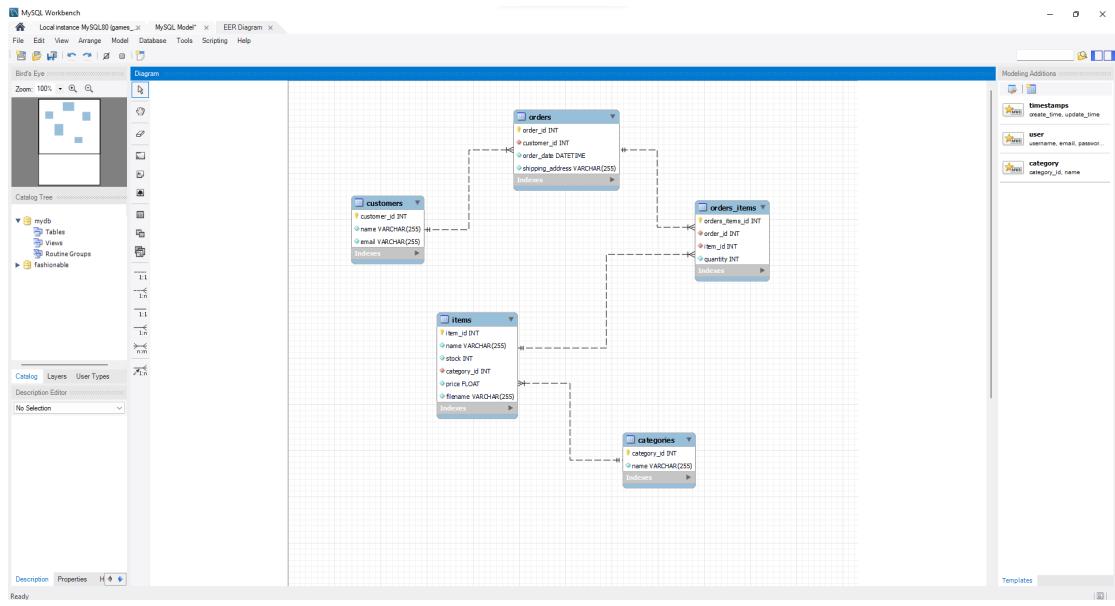
```
27
28
29  class Orders:
30      order_id = db.Column(db.Integer, primary_key=True)
31      customer_id = db.Column(
32          db.Integer, db.ForeignKey("customers.customer_id"), nullable=False
33     )
34      order_date = db.Column(db.DateTime, nullable=False)
35      shipping_address = db.Column(db.String(255), nullable=False)
36      orders_items = db.relationship("OrdersItems", backref="orders")
37
38
39  class OrdersItems:
40      orders_items_id = db.Column(db.Integer, primary_key=True)
41      order_id = db.Column(db.Integer, db.ForeignKey("orders.order_id"), nullable=False)
42      item_id = db.Column(db.Integer, db.ForeignKey("items.item_id"), nullable=False)
43      quantity = db.Column(db.Integer, nullable=False)
44
```

Created models based on class definition and defined relationships based on ER UML diagram

```
models.py | C:\Users\rober\Documents\MEGA\work\AMS\ISFA-CRUD-core-project | □ □ □ 88 - □ ×
routes.py  models.py M X  create.py  __init__.py  ▶ v ⌂ ⌂ ...
```

```
application > models.py > OrdersItems
1   from application import db
2
3
4   class Items(db.Model):
5       item_id = db.Column(db.Integer, primary_key=True)
6       name = db.Column(db.String(255), nullable=False)
7       stock = db.Column(db.Integer, nullable=False, default=0)
8       category_id = db.Column(
9           db.Integer, db.ForeignKey("categories.category_id"), nullable=False
10      )
11      price = db.Column(db.Float, nullable=False)
12      filename = db.Column(db.String(255), nullable=False)
13      orders_items = db.relationship("OrdersItems", backref="items")
14
15
16   class Categories(db.Model):
17       category_id = db.Column(db.Integer, primary_key=True)
18       name = db.Column(db.String(255), nullable=False)
19       items = db.relationship("Items", backref="category")
20
21
22   class Customers(db.Model):
23       customer_id = db.Column(db.Integer, primary_key=True)
24       name = db.Column(db.String(255), nullable=False)
25       email = db.Column(db.String(255), nullable=False, unique=True)
26       orders = db.relationship("Orders", backref="customer")
27
28
29   class Orders(db.Model):
30       order_id = db.Column(db.Integer, primary_key=True)
31       customer_id = db.Column(
32           db.Integer, db.ForeignKey("customers.customer_id"), nullable=False
33      )
34       order_date = db.Column(db.DateTime, nullable=False)
35       shipping_address = db.Column(db.String(255), nullable=False)
36       orders_items = db.relationship("OrdersItems", backref="orders")
37
38
39   class OrdersItems(db.Model):
40       orders_items_id = db.Column(db.Integer, primary_key=True)
41       order_id = db.Column(db.Integer, db.ForeignKey("orders.order_id"), nullable=False)
42       item_id = db.Column(db.Integer, db.ForeignKey("items.item_id"), nullable=False)
```

Fixed inheritance from db.Model class



Inspecting database schema using the create.py file

```

create.py | C:\Users\rober\Documents\MEGA\work\AMS\ISFA-CRUD-core-project
=====
create.py M x models.py

create.py > ...
1  from application import app, db
2  from application.models import *
3
4  # create the database schema or even just test db connection
5  with app.app_context():
6      db.drop_all()
7      db.create_all()
8
9      # adding sample categories and items
10     jeans_category = Categories(name="Jeans")
11     skirts_category = Categories(name="Skirts")
12     tshirts_category = Categories(name="T-shirts")
13     dresses_category = Categories(name="Dresses")
14     sweaters_category = Categories(name="Sweaters")
15
16     db.session.add(jeans_category)
17     db.session.add(skirts_category)
18     db.session.add(tshirts_category)
19     db.session.add(dresses_category)
20     db.session.add(sweaters_category)
21

```

The screenshot shows a code editor with two tabs open: 'create.py M' and 'models.py'. The 'create.py' tab contains the following code:

```
jeans_item = Items(
    name="Jeans Item 1",
    stock=10,
    category=jeans_category,
    price=9.99,
    filename="jeans_1.jpg",
)

skirt_item = Items(
    name="Skirt Item 1",
    stock=5,
    category=skirts_category,
    price=4.99,
    filename="skirt_1.jpg",
)

tshirt_item = Items(
    name="T-shirt Item 1",
    stock=10,
    category=tshirts_category,
    price=14.99,
    filename="tshirt_1.jpg",
)

dress_item = Items(
    name="Dress Item 1",
    stock=2,
    category=dresses_category,
    price=29.99,
    filename="dress_1.jpg",
)

sweater_item_1 = Items(
    name="Sweater Item 1",
    stock=5,
    category=sweaters_category,
    price=19.99,
    filename="sweater_1.jpg",
)

db.session.add(jeans_item)
db.session.add(skirt_item)
db.session.add(tshirt_item)
db.session.add(dress_item)
db.session.add(sweater_item_1)

db.session.commit()
```

The image shows two screenshots of MySQL Workbench. Both screenshots have a similar layout: a top navigation bar with tabs like File, Edit, View, Query, Database, Server, Tools, Scripting, and Help; a left sidebar titled 'Navigator' showing 'SCHEMAS' and 'fashionable' schema details; and a right sidebar with icons for Result Grid, Form Editor, Field Types, Query Stats, and Execution Plan.

Screenshot 1:

```

USE fashionable;
SELECT * FROM items;

```

Result Grid:

item_id	name	stock	category_id	price	filename
1	Jeans Item 1	10	1	9.99	jeans_1.jpg
2	Skirt Item 1	5	2	4.99	skirt_1.jpg
3	T-shirt Item 1	10	3	14.99	tshirt_1.jpg
4	Dress Item 1	2	4	29.99	dress_1.jpg
5	Sweater Item 1	5	5	19.99	sweater_1.jpg

Screenshot 2:

```

USE fashionable;
SELECT * FROM categories;

```

Result Grid:

category_id	name
1	Jeans
2	Skirts
3	T-shirts
4	Dresses
5	Sweaters

Modified create.py to add sample products and categories to the database. Verified that everything was working fine using MySQL workbench

QS-39 / QS-78

Development: code

[Attach](#) [Add a child issue](#) [Link issue](#) [...](#)

Description

Add a description...

Child issues

Order by [...](#) + 100% Done

Issue ID	Description	Status
qs-94	Create basic flask app	= - RN DONE
qs-95	Create basic MySQL database	= - RN DONE

Activity

Re-updated create basic MySQL database task

Projects / QA Store Backlog

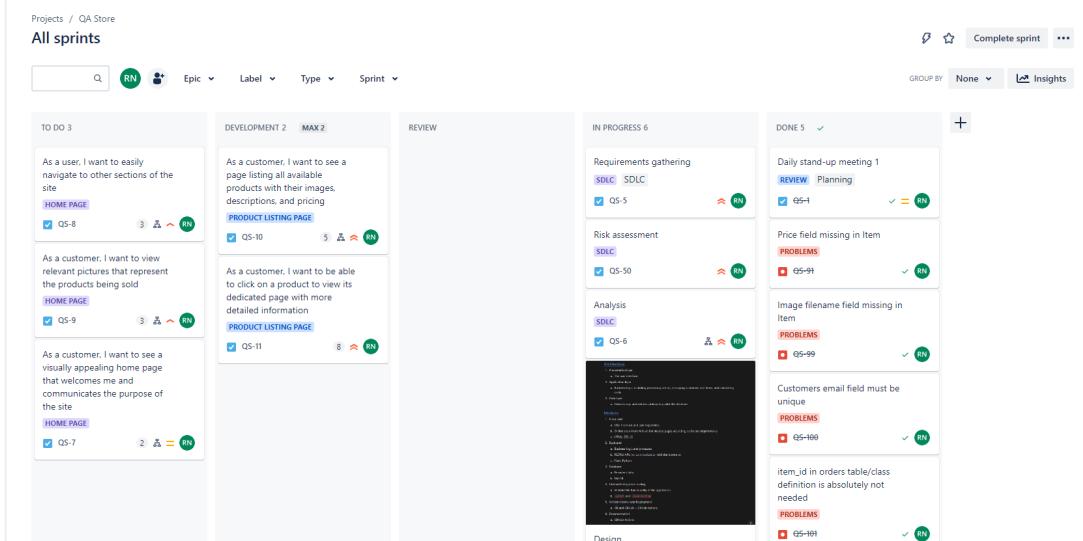
Issue ID	Description	Category	Status
qs-87	Design	SDLC	= - IN PROGRESS RN
qs-94	Price field missing in item	PROBLEMS	X - DONE RN
qs-78	Development: code	SDLC	= - IN PROGRESS RN
qs-99	Image filename field missing in item	PROBLEMS	X - DONE RN
qs-79	Development: testing	SDLC	= - IN PROGRESS RN

+ Create issue

▼ QS Sprint 1 11 Aug – 11 Aug (8 issues)

Issue ID	Description	Category	Status
qs-101	Daily stand-up meeting 1	REVIEW	= - DONE RN
qs-100	Customers email field must be unique	PROBLEMS	X - DONE RN
qs-101	item_id in orders table/class definition is absolutely not needed	PROBLEMS	X - DONE RN
qs-10	As a customer, I want to see a page listing all available products with their images, descriptions, and pricing	PRODUCT LISTING PAGE	= - DEVELOPMENT RN
qs-11	As a customer, I want to be able to click on a product to view its dedicated page with more detailed information	PRODUCT LISTING PAGE	= - DEVELOPMENT RN
qs-8	As a user, I want to easily navigate to other sections of the site	HOME PAGE	= - TO DO RN
qs-9	As a customer, I want to view relevant pictures that represent the products being sold	HOME PAGE	= - TO DO RN
qs-7	As a customer, I want to see a visually appealing home page that welcomes me and communicates the purpose of the site	HOME PAGE	= - TO DO RN

+ Create issue



Real development time: added first user stories/tasks to technically sprint 1

```
routes.py | C:\Users\rober\Documents\MEGA\work\AMS\ISFA-CRUD-core-project
```

```
application > routes > routes.py
```

```
1 from application import app
2 
```

```
(venv)
rober@shizu MINGW64 ~ /Documents/MEGA/work/AMS/ISFA-CRUD-core-project (feature-productpage)
$ 
```

Using feature- branch naming convention for working on new features. Screens might be omitted in the future but this will be done for each new feature

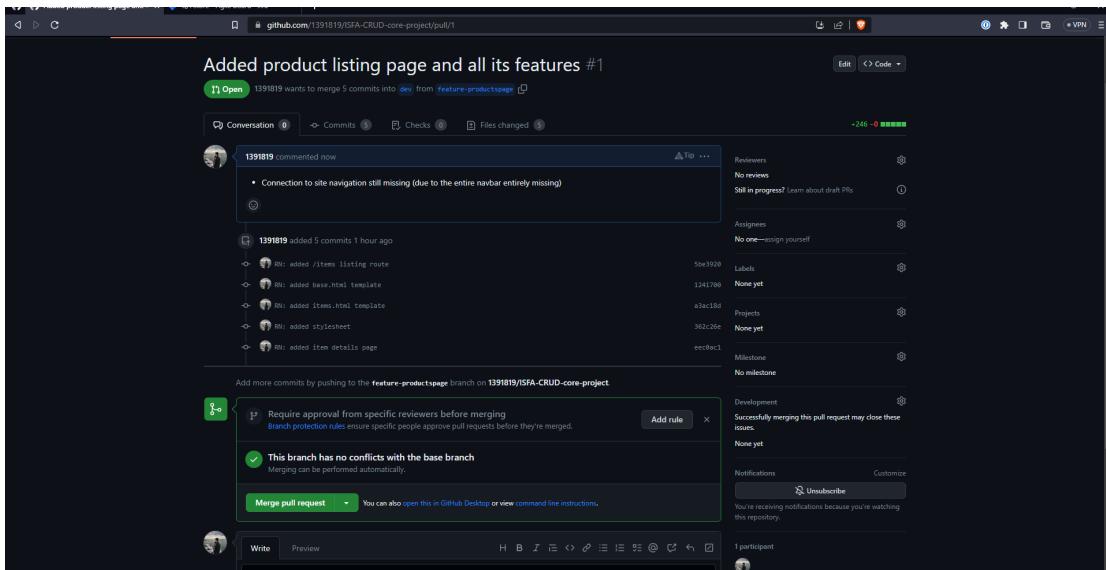
The screenshot shows a Python development environment with the following details:

- EXPLORER** pane on the left displays the project structure for "ISFA-CRUD-core-project".
- ROUTINES** pane at the top shows the current file open: routes.py.
- CODE** pane contains the Python code for the routes module:

```
application.routes > routes.py @ item_page
1  from application import app, db
2  from application.models import Items, Categories
3  from flask import render_template
4
5  @app.route("/items")
6  def all_items():
7      """Items endpoint: displaying all items
8
9      Returns:
10         _type_: _description_
11         """
12         # retrieving all items
13         items = Items.query.all()
14         # rendering appropriate template
15         return render_template("items.html", items=items)
16
17
18 @app.route("/items/<int:item_id>")
19 def item_page(item_id):
20     """Single item endpoint
21
22     Args:
23         item_id (int): ID of the item we want to display
24
25     Returns:
26         _type_: _description_
27         """
28         # retrieving item using passed item_id
29         item = Items.query.filter_by(item_id=item_id).first()
30         # rendering appropriate template
31         return render_template("item.html", item=item)
```

- PROBLEMS**, **OUTPUT**, **DEBUG CONSOLE**, and **TERMINAL** tabs are visible at the bottom.
- TERMINAL** tab shows the command: `robertshizhi@MINGW64 ~ /Documents/MINGW/work/ISFA-CRUD-core-project (feature-productpage) $`.

Product listing epic completed



Merging feature branch to dev branch

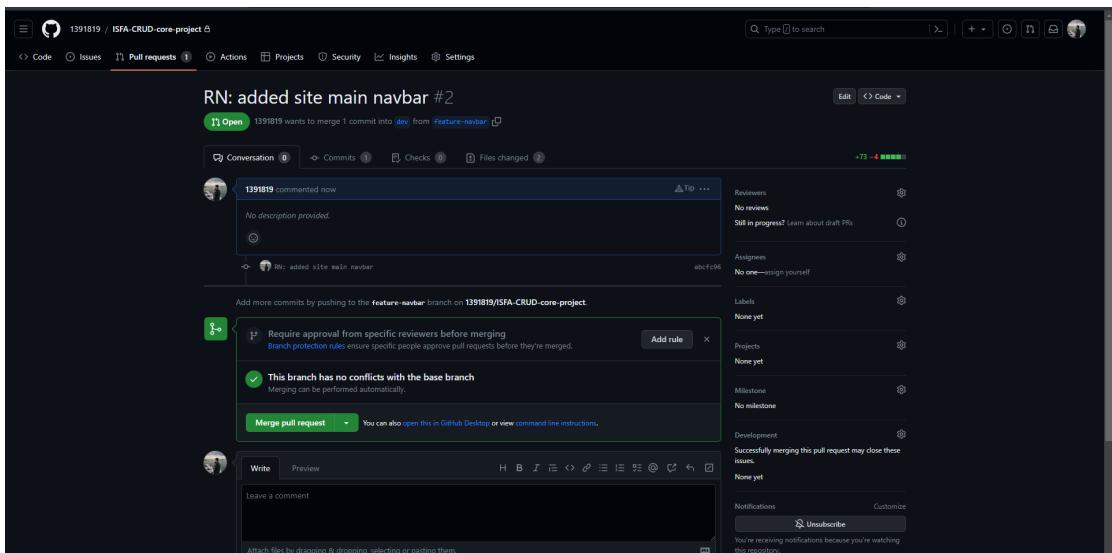
The screenshot shows the VS Code interface with the 'base.html' file open in the editor. The code is a template for a navigation bar, featuring a brand logo, a toggle button for mobile view, and a list of links for Home, Products, Categories, and About us. The file is part of the 'ISFA-CRUD-CORE-PROJECT' in the 'application' folder.

```

<div class="container">
  <a class="navbar-brand" href="#">
  <button class="navbar-toggler" type="button" data-bs-toggle="collapse" data-bs-target="#navbarSupportedContent" aria-controls="navbarSupportedContent" aria-expanded="false" aria-label="Toggle navigation">
    <span class="navbar-toggler-icon"></span>
  </button>
  <div class="collapse navbar-collapse justify-content-end" id="navbarSupportedContent">
    <ul class="navbar-nav">
      <li class="nav-item active">
        <a class="nav-link" href="#"> Home
      </li>
      <li class="nav-item">
        <a class="nav-link" href="#"> Products
      </li>
      <li class="nav-item">
        <a class="nav-link" href="#"> Categories
      </li>
      <li class="nav-item">
        <a class="nav-link" href="#"> About us
      </li>
    </ul>
  </div>

```

Created feature-navbar and completed site's main navigation



Merged feature-navbar branch to dev branch

QS-29 / QS-10

As a customer, I want to see a page listing all available products with their images, descriptions, and pricing

Attach Add a child issue Link issue ...

Description
Add a description...

Child issues Order by ... + 100% Done

	Issue	Priority	Status
Q5-40	Make it part of the overall site navigation	RN	DONE
Q5-41	Have at least 4 different products images on display	RN	DONE
Q5-59	Create overall layout	RN	DONE
Q5-60	Design product card	RN	DONE

Activity Show: All Comments History Newest first ↓

Add a comment...
Pro tip: press M to comment

Completed this user story

Development Actions

Details

Assignee RN Roberto Nacu

Priority Highest

Labels None

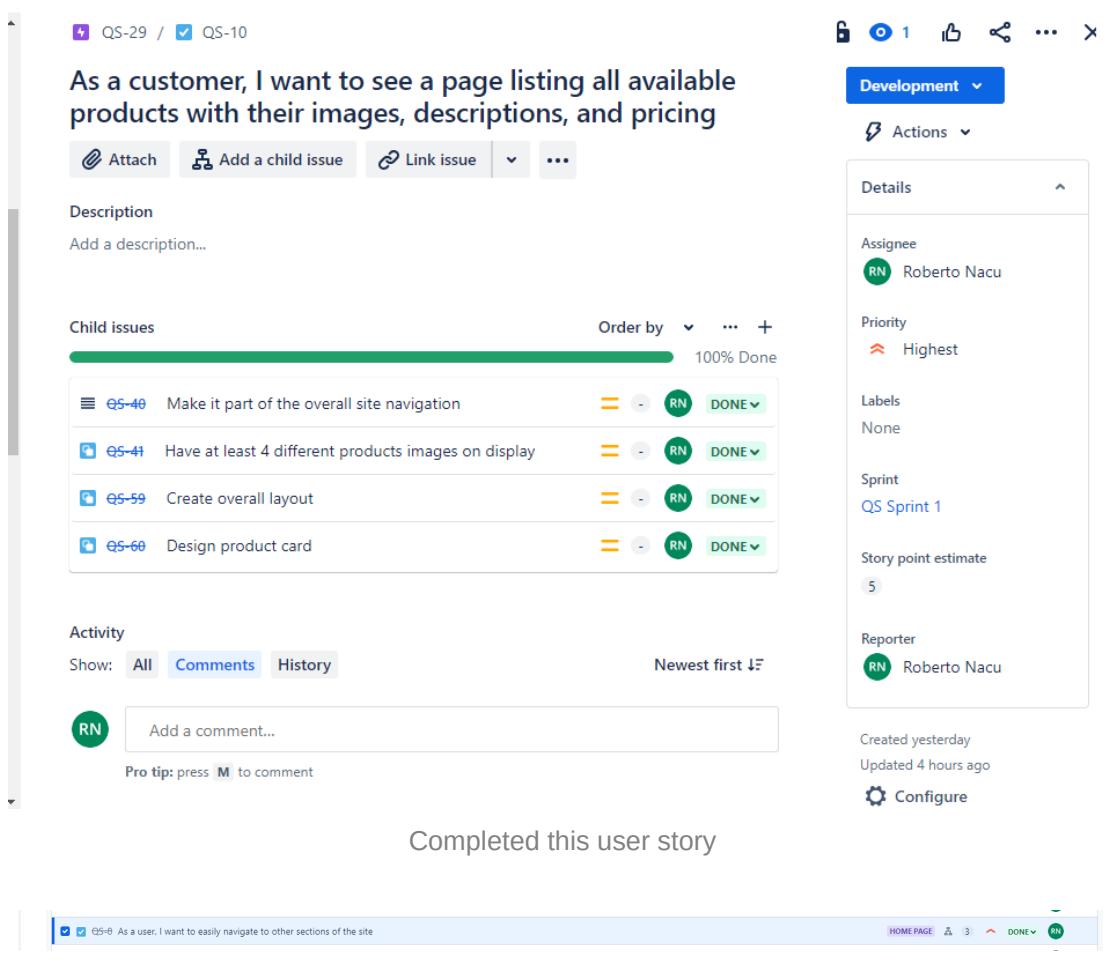
Sprint QS Sprint 1

Story point estimate 5

Reporter RN Roberto Nacu

Created yesterday Updated 4 hours ago

Configure



Q5-60 As a user, I want to easily navigate to other sections of the site

HOME PAGE 3 DONE

Completed this user story

QS-28 / QS-9

As a customer, I want to view relevant pictures that represent the products being sold

Done ✓ Done

Actions

Description
Add a description...

Child issues

Issue	Order by	Status
QS-56 Create products displaying layout	1	DONE
QS-57 Retrieve products information from database	2	DONE
QS-58 Display retrieved information	3	DONE

Activity
Show: All Comments History Newest first ↴

Add a comment...
RN

Pro tip: press M to comment

Details

- Assignee: Roberto Nacu (RN)
- Priority: High
- Labels: None
- Sprint: QS Sprint 1
- Story point estimate: 3
- Reporter: Roberto Nacu (RN)

Created yesterday
Updated 18 seconds ago

Completed this user story and its tasks

RN: Added home page endpoint and all its features #3

Merged 1391819 merged 4 commits into dev from feature-homepage now

Conversation 0 Commits 4 Checks 0 Files changed 0

1391819 commented now
No description provided.

1391819 added 4 commits 3 hours ago

- RN: added home/index endpoint
- RN: create index.html template + stylesheet and added to navbar
- RN: created item_card template (modularity)
- RN: added home page endpoint and its features

Pull request successfully merged and closed You're all set—the feature-homepage branch can be safely deleted. Delete branch

Review

Reviewers: No reviews

Assignees: No one—assign yourself

Labels: None yet

Projects: None yet

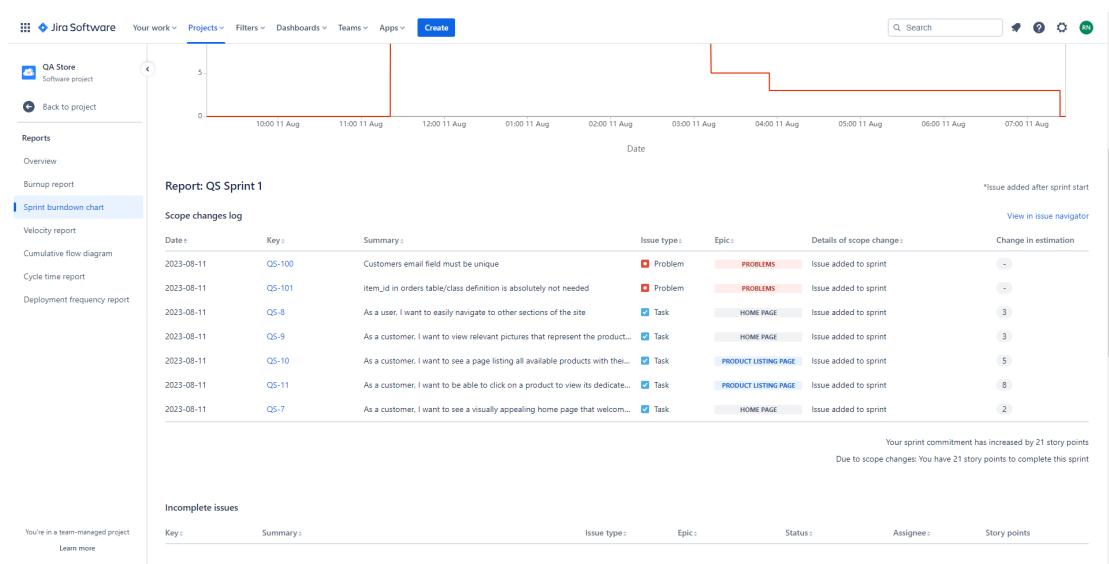
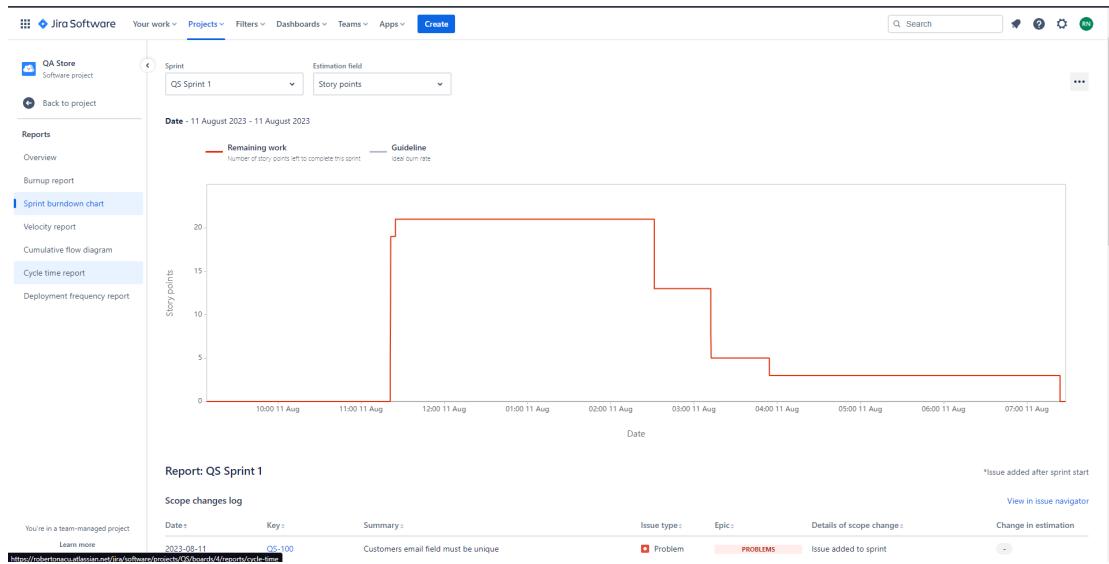
Milestone: No milestone

Development: Successfully merging this pull request may close these issues. None yet

Notifications: Unsubscribe Customize

You're receiving notifications because you're watching this repository.

Merged feature homepage branch to dev branch



Completed Sprint 1

✓ QS Sprint 2 12 Aug – 14 Aug (12 issues)

Development of the main parts of the project according to user stories. Prioritisation will be considered due to time constraints.

REVIEW	TO DO	IN PROGRESS	TESTED
<input type="checkbox"/> QS-2: Daily stand-up meeting 2	<input type="checkbox"/> QS-12: As a customer, I want to access a page dedicated to product categories so that I can easily find products based on my interests	<input checked="" type="checkbox"/> CATEGORY PAGE	<input checked="" type="checkbox"/> CATEGORY PAGE
<input type="checkbox"/> QS-13: As a user, I want the option to view products specific to a chosen category	<input checked="" type="checkbox"/> QS-13: As a user, I want the option to view products specific to a chosen category	<input checked="" type="checkbox"/> CART PAGE	<input checked="" type="checkbox"/> CART PAGE
<input type="checkbox"/> QS-14: As a customer, I want a page where I can see all the products I've added to my cart	<input checked="" type="checkbox"/> QS-14: As a customer, I want a page where I can see all the products I've added to my cart	<input checked="" type="checkbox"/> CHECKOUT PAGE	<input checked="" type="checkbox"/> CHECKOUT PAGE
<input type="checkbox"/> QS-15: As a customer, I want to be able to adjust the quantity of each product in my cart before proceeding to checkout	<input checked="" type="checkbox"/> QS-15: As a customer, I want to be able to adjust the quantity of each product in my cart before proceeding to checkout	<input checked="" type="checkbox"/> PAYMENT PAGE	<input checked="" type="checkbox"/> PAYMENT PAGE
<input type="checkbox"/> QS-38: As a customer, I want a page where I can see a summary of my cart and enter my shipping information	<input checked="" type="checkbox"/> QS-38: As a customer, I want a page where I can see a summary of my cart and enter my shipping information	<input checked="" type="checkbox"/> CONTACT US PAGE	<input checked="" type="checkbox"/> CONTACT US PAGE
<input type="checkbox"/> QS-16: As a customer, I want a page where I can securely enter my payment information including cardholder's name, card number, expiry date, and security code	<input checked="" type="checkbox"/> QS-16: As a customer, I want a page where I can securely enter my payment information including cardholder's name, card number, expiry date, and security code	<input checked="" type="checkbox"/> ABOUT US PAGE	<input checked="" type="checkbox"/> ABOUT US PAGE
<input type="checkbox"/> QS-17: As a customer, I want to see a confirmation that my payment details were successfully processed	<input checked="" type="checkbox"/> QS-17: As a customer, I want to see a confirmation that my payment details were successfully processed	<input checked="" type="checkbox"/> PAYMENT PAGE	<input checked="" type="checkbox"/> PAYMENT PAGE
<input type="checkbox"/> QS-18: As a user, I want to easily find contact information for the store, including address, phone number, and email	<input checked="" type="checkbox"/> QS-18: As a user, I want to easily find contact information for the store, including address, phone number, and email	<input checked="" type="checkbox"/> CONTACT US PAGE	<input checked="" type="checkbox"/> CONTACT US PAGE
<input type="checkbox"/> QS-19: As a user, I want to have a contact form to submit enquiries or provide feedback to the store	<input checked="" type="checkbox"/> QS-19: As a user, I want to have a contact form to submit enquiries or provide feedback to the store	<input checked="" type="checkbox"/> ABOUT US PAGE	<input checked="" type="checkbox"/> ABOUT US PAGE
<input type="checkbox"/> QS-20: As a user, I want to access a page that provides information about the store's mission, history, and the team behind it	<input checked="" type="checkbox"/> QS-20: As a user, I want to access a page that provides information about the store's mission, history, and the team behind it	<input checked="" type="checkbox"/> PAYMENT PAGE	<input checked="" type="checkbox"/> PAYMENT PAGE
<input type="checkbox"/> QS-21: As a user, I want to learn about the people responsible for running the store	<input checked="" type="checkbox"/> QS-21: As a user, I want to learn about the people responsible for running the store	<input checked="" type="checkbox"/> CONTACT US PAGE	<input checked="" type="checkbox"/> CONTACT US PAGE

+ Create issue

Added remaining user stories to Sprint 2. Will be completed based on assigned priority.
Intent is to leave last day for testing and reviewing project

This is named Sprint 3, however, it would be the last day of Sprint 2. Due to the nature of the project and for organisational purposes I just decided to create it.

Starting Sprint 2.

Q5-12 As a customer, I want to access a page dedicated to product categories so that I can easily find products based on my interests
 Q5-13 As a user, I want the option to view products specific to a chosen category

CATEGORY PAGE 5 IN DONE 6 IN

+ Create issue

Completed first two issues that were under development

1391819 / ISFA-CRUD-core-project

Pull requests

RN: Added categories page and its core features #4

Merged 1391819 merged 4 commits into dev from feature-categorypage now

Conversation 0 Commits 0 Checks 0 Files changed 0

1391819 commented now
No description provided.

1391819 added 4 commits 19 minutes ago
RN: added more sample products to database
RN: added scripts jstree2 block
RN: added categories page with displaying category items functionality
RN: added all categories option + items load on page load

1391819 merged commit 5d5eb2b into dev now

Pull request successfully merged and closed
You're all set—the feature-categorypage branch can be safely deleted.

Review

Reviewers
No reviews

Assignees
No one—assign yourself

Labels
None yet

Projects
None yet

Milestone
No milestone

Development
Successfully merging this pull request may close these issues.
None yet

Notifications
Unsubscribe

Merged feature-categorypage branch to dev branch

QA Store Software project

PLANNING Timeline Backlog Board Reports Project pages Add shortcut Project settings

Projects / QA Store All sprints

TO DO 7 DEVELOPMENT 2 MAX 2 REVIEW IN PROGRESS 6 DONE 5

As a customer, I want a page where I can see a summary of my cart and enter my shipping information
CHECKOUT PAGE
QS-38

As a customer, I want a page where I can securely enter my payment information including cardholder's name, card number, expiry date, and security code
PAYMENT PAGE
QS-16

As a customer, I want to see a confirmation that my payment details were successfully processed
PAYMENT PAGE
QS-17

As a customer, I want a page where I can see all the products I've added to my cart
CART PAGE
QS-14

As a customer, I want to be able to adjust the quantity of each product in my cart before proceeding to checkout
CART PAGE
QS-15

Requirements gathering SDLC QS-5

Risk assessment SDLC QS-50

Analysis SDLC QS-6

Daily stand-up meeting 2 REVIEW QS-2

Field price missing in Item PROBLEMS QS-9

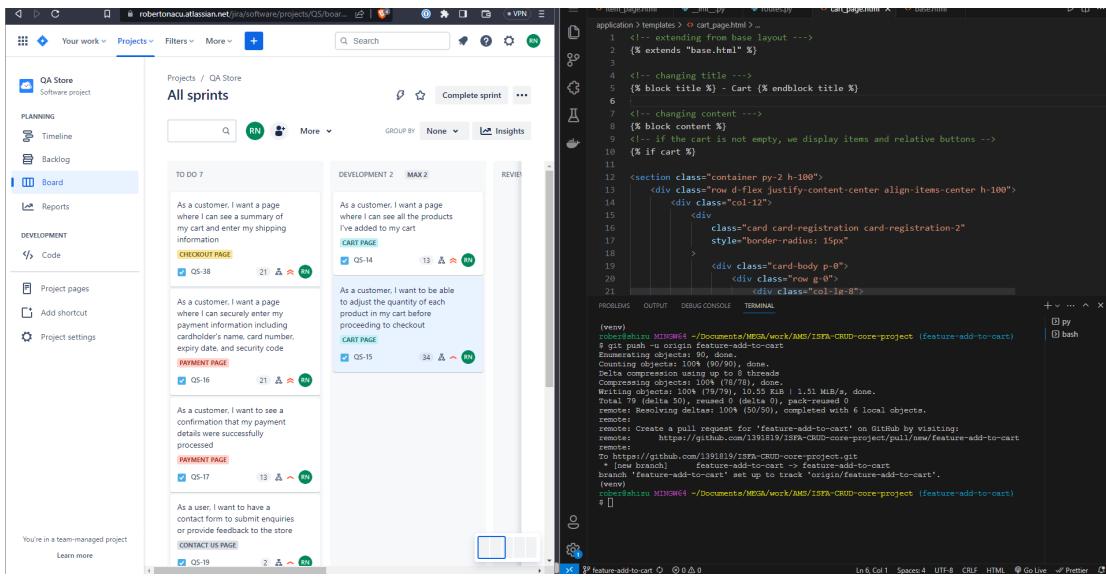
Image filename field missing in Item PROBLEMS QS-99

As a customer, I want to access a page dedicated to product categories so that I can easily find products based on my interests
CATEGORY PAGE QS-12

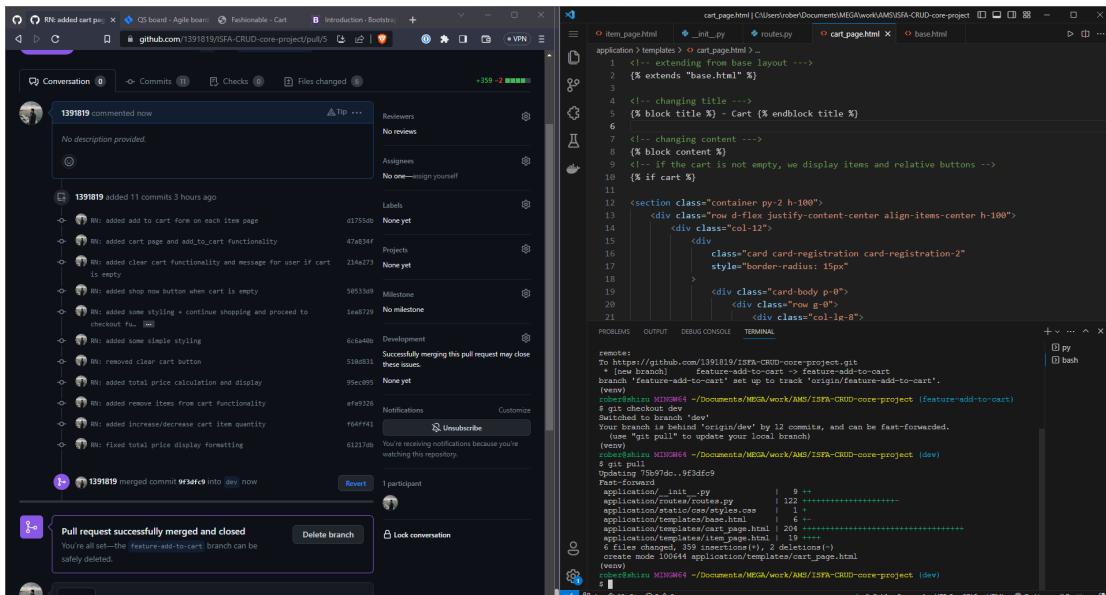
As a user, I want the option to view products specific to a chosen category
CATEGORY PAGE QS-11

You're in a team-managed project Learn more

Identifying next issues that must be worked on



Completed issues and child tasks under development



Merged feature-add-to-cart branch to dev branch

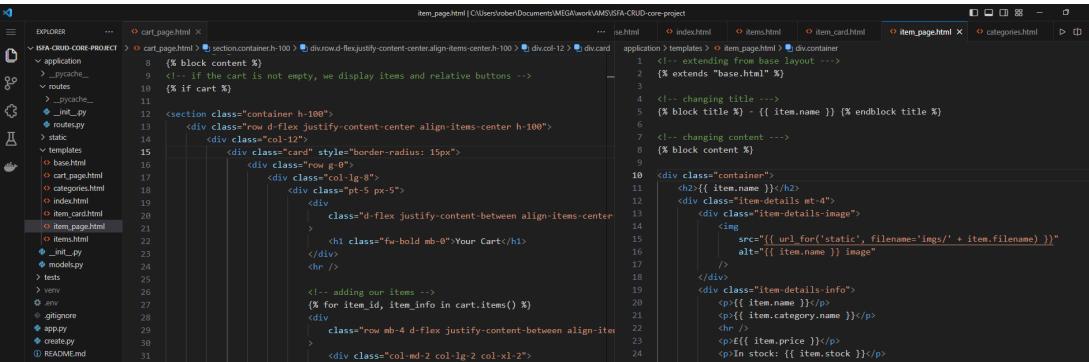
Projects / QA Store

All sprints

SEARCH GROUP BY Insights

EPIC	TYPE	STATUS	TESTS
As a customer, I want a page where I can securely enter my payment information including cardholder's name, card number, expiry date, and security code	PAYMENT PAGE	TO DO 6	<input checked="" type="checkbox"/> QS-16 21
As a customer, I want a page where I can see a summary of my cart and enter my shipping information	CHECKOUT PAGE	DEVELOPMENT 1 MAX 2	<input checked="" type="checkbox"/> QS-38 21
As a user, I want to see a confirmation that my payment details were successfully processed	PAYMENT PAGE	REVIEW	<input checked="" type="checkbox"/> QS-39 21
As a user, I want to have a contact form to submit enquiries or provide feedback to the store	CONTACT US PAGE	IN PROGRESS 6	<input checked="" type="checkbox"/> QS-5 21
As a user, I want to easily find contact information for the store, including address, phone number, and email	CONTACT US PAGE	DONE 7	<input checked="" type="checkbox"/> QS-12 5
As a customer, I want to access a page dedicated to product categories so that I can easily find products based on my interests	CATEGORY PAGE	IN PROGRESS 6	<input checked="" type="checkbox"/> QS-6 5
As a user, I want the option to view products specific to a chosen category	CATEGORY PAGE	DONE 7	<input checked="" type="checkbox"/> QS-13 8

Identified next development issue



The screenshot shows a code editor with two tabs open: `item_page.html` and `item_page.html.x`. The left pane displays the original `item_page.html` file, which contains HTML code for a shopping cart page. The right pane shows the modified version `item_page.html.x`, which includes additional CSS classes and a new section for displaying items. The code editor has a dark theme with syntax highlighting for HTML and CSS.

```
item_page.html | C:\Users\roben\Documents\MEGA\work\AMG\ISFA-CRUD-core-project | item_page.html.x | categories.html | ...
```

```
EXPLORER application _pycache_ routes _pycache_ routes.py static templates base.html cart_page.html categories.html index.html item_card.html item_page.html item_page.html.x item_page.html
```

```
application
  +-- __init__.py
  +-- __pycache__
  +-- routes
      +-- __pycache__
      +-- routes.py
  +-- static
  +-- templates
      +-- base.html
      +-- cart_page.html
      +-- categories.html
      +-- index.html
      +-- item_card.html
      +-- item_page.html
      +-- item_page.html.x
      +-- item_page.html~
  +-- requirements.txt
```

```
item_page.html
  8   <!-- block content -->
  9   <!-- if the cart is not empty, we display items and relative buttons -->
 10  <% if cart %>
 11
 12  <section class="container h-100">
 13    <div class="row d-flex justify-content-center align-items-center h-100">
 14      <div class="col-12">
 15        <div class="card" style="border-radius: 15px">
 16          <div class="row g-0">
 17            <div class="col-12">
 18              <div class="pt-5 px-5">
 19                <div class="d-flex justify-content-between align-items-center">
 20                  <h1 class="fw-bold mb-0">Your Cart</h1>
 21                  <hr />
 22
 23                  <!-- adding our items -->
 24                  <% for item_id, item_info in cart.items() %>
 25                    <div class="row mb-4 d-flex justify-content-between align-items-center">
 26                      <div class="col-md-2 col-lg-2 col-xl-2">
 27                        
 28                    </div>
 29
 30                    <div class="col-md-2 col-lg-2 col-xl-2">
 31                      <p>
```

```
item_page.html.x
  1 <!-- extending from base layout -->
  2 <% extends "base.html" %>
  3
  4   <% block title %> {{ item.name }} <% endblock title %>
  5
  6   <% block content %>
  7
  8   <% block content %>
  9
 10  <div class="container">
 11    <h2>{{ item.name }}</h2>
 12    <div class="item-details mt-4">
 13      <div class="item-details-image">
 14        
 16
 17      <div class="item-details-info">
 18        <p>{{ item.name }}</p>
 19        <p>{{ item.category.name }}</p>
 20        <hr />
 21        <p>{{ item.price }}</p>
 22        <p>In stock: {{ item.stock }}</p>
 23
 24      <!-- Add to cart functionality -->
 25      <form method="POST" action="{{ url_for('add_to_cart', item_id=item.id) }}>
 26        <button type="submit">>

```
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL
```



```
(venv) tshirtezhihu MINGW64 ~\Documents\MEGA\work\AMG\ISFA-CRUD-core-project (dev)
$
```



```
+ bash
+ py
```


```

Applying modifications to styling and other minor things. In a real world scenario, these changes would require a new branch just for safety reasons. However, due to the nature of the project (individual), doing this is fine for now.

```

29 |     orders = db.relationship('Orders', backref='customer')
30 |
31 |
32 class Orders(db.Model):
33     order_id = db.Column(db.Integer, primary_key=True)
34     customer_id = db.Column(
35         db.Integer, db.ForeignKey("customers.customer_id"), nullable=False
36     )
37     order_date = db.Column(db.DateTime, nullable=False)
38     shipping_address = db.Column(db.String(255), nullable=False)
39     orders_items = db.relationship("OrdersItems", backref="orders")
40 |
41 |
42 class OrdersItems(db.Model):
43     orders_items_id = db.Column(db.Integer, primary_key=True)
44     order_id = db.Column(db.Integer, db.ForeignKey("orders.order_id"), nullable=False)
45     item_id = db.Column(db.Integer, db.ForeignKey("items.item_id"), nullable=False)
46     quantity = db.Column(db.Integer, nullable=False)
47 |
48 |
49 # forms
50 class ShippingForm:
51     # customer table data

```

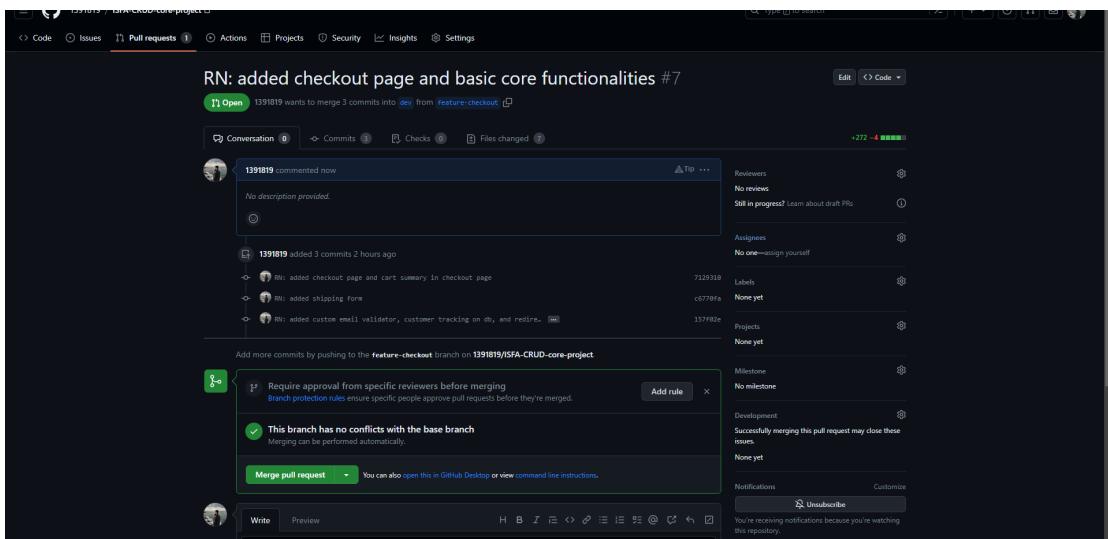
PROBLEMS 8 OUTPUT DEBUG CONSOLE TERMINAL

```

(venv)
rober@shizu MINGW64 ~/Documents/MEGA/work/AMS/ISFA-CRUD-core-project (feature-checkout)
$ pip install Flask-WTF WTForms
Collecting Flask-WTF
    Using cached Flask_WTF-1.1.1-py3-none-any.whl (12 kB)
Collecting WTForms
    Using cached WTForms-3.0.1-py3-none-any.whl (136 kB)
Requirement already satisfied: Flask in c:\users\rober\documents\mega\work\ams\isfa-crud-core-project\venv\lib\site-packages (from Flask-WTF) (2.3.2)
Requirement already satisfied: itsdangerous in c:\users\rober\documents\mega\work\ams\isfa-crud-core-project\venv\lib\site-packages (from Flask-WTF) (2.1.2)
Requirement already satisfied: MarkupSafe in c:\users\rober\documents\mega\work\ams\isfa-crud-core-project\venv\lib\site-packages (from WTForms) (2.1.3)
Requirement already satisfied: Werkzeug>=2.3.3 in c:\users\rober\documents\mega\work\ams\isfa-crud-core-project\venv\lib\site-packages (from Flask->Flask-WTF) (2.3.6)
Requirement already satisfied: Jinja2>=3.1.2 in c:\users\rober\documents\mega\work\ams\isfa-crud-core-project\venv\lib\site-packages (from Flask->Flask-WTF) (3.1.2)
Requirement already satisfied: click>=8.1.3 in c:\users\rober\documents\mega\work\ams\isfa-crud-core-project\venv\lib\site-packages (from Flask->Flask-WTF) (8.1.6)
Requirement already satisfied: blinker>=1.6.2 in c:\users\rober\documents\mega\work\ams\isfa-crud-core-project\venv\lib\site-packages (from Flask->Flask-WTF) (1.6.2)

```

Started working on shipping form for the checkout page. Installed additional dependencies



Merging feature-checkout branch to dev branch

This screenshot shows a Jira Kanban board for the 'QA Store' project. The board is divided into four columns: 'TO DO', 'DEVELOPMENT', 'REVIEW', and 'IN PROGRESS'. Each column contains several issues, each with a title, description, and status. The issues are color-coded by priority: RN (Red), MAX (Max Priority), and others. The 'DEVELOPMENT' column contains issues like 'CONTACT US PAGE' and 'PAYMENT PAGE'. The 'IN PROGRESS' column contains issues like 'Requirements gathering' and 'Risk assessment'. The 'DONE' section at the bottom right shows completed tasks.

Updating issues under development

This screenshot shows a GitHub pull request page for a repository named 'ISFA-CRUD-core-project'. The pull request is titled 'RN: added payment details form and redirect to order processed page #8'. It has been merged into the 'dev' branch. The page displays the commit history, which includes two commits from user '1391819'. The pull request has been successfully merged and closed. The GitHub interface also shows the issue it was addressing, the conversation, and various repository settings like milestones and notifications.

Merged new features to dev branch

This screenshot shows a Jira Kanban board for the 'QA Store' project, similar to the one above but with a different set of issues in the 'Development' column. The board has four columns: 'TO DO', 'DEVELOPMENT', 'REVIEW', and 'IN PROGRESS'. The 'DEVELOPMENT' column contains issues like 'CONTACT US PAGE' and 'PAYMENT PAGE'. The 'IN PROGRESS' column contains issues like 'Requirements gathering' and 'Risk assessment'. The 'DONE' section at the bottom right shows completed tasks.

Identifying next development issues

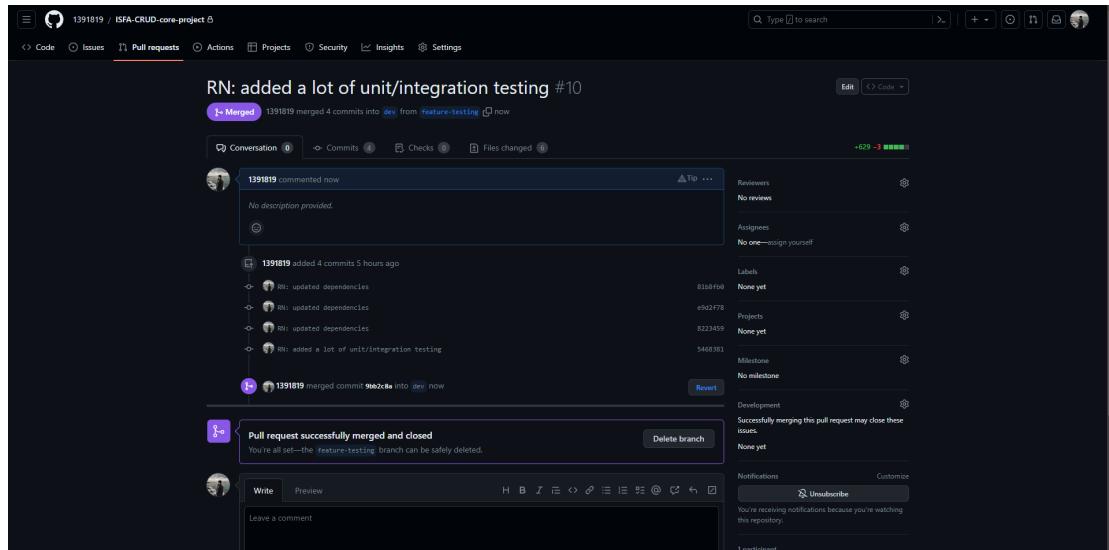
Started working on some refactoring before finishing the last 4 development issues which have a low priority/lower story points assigned

SDLC-5: Testing

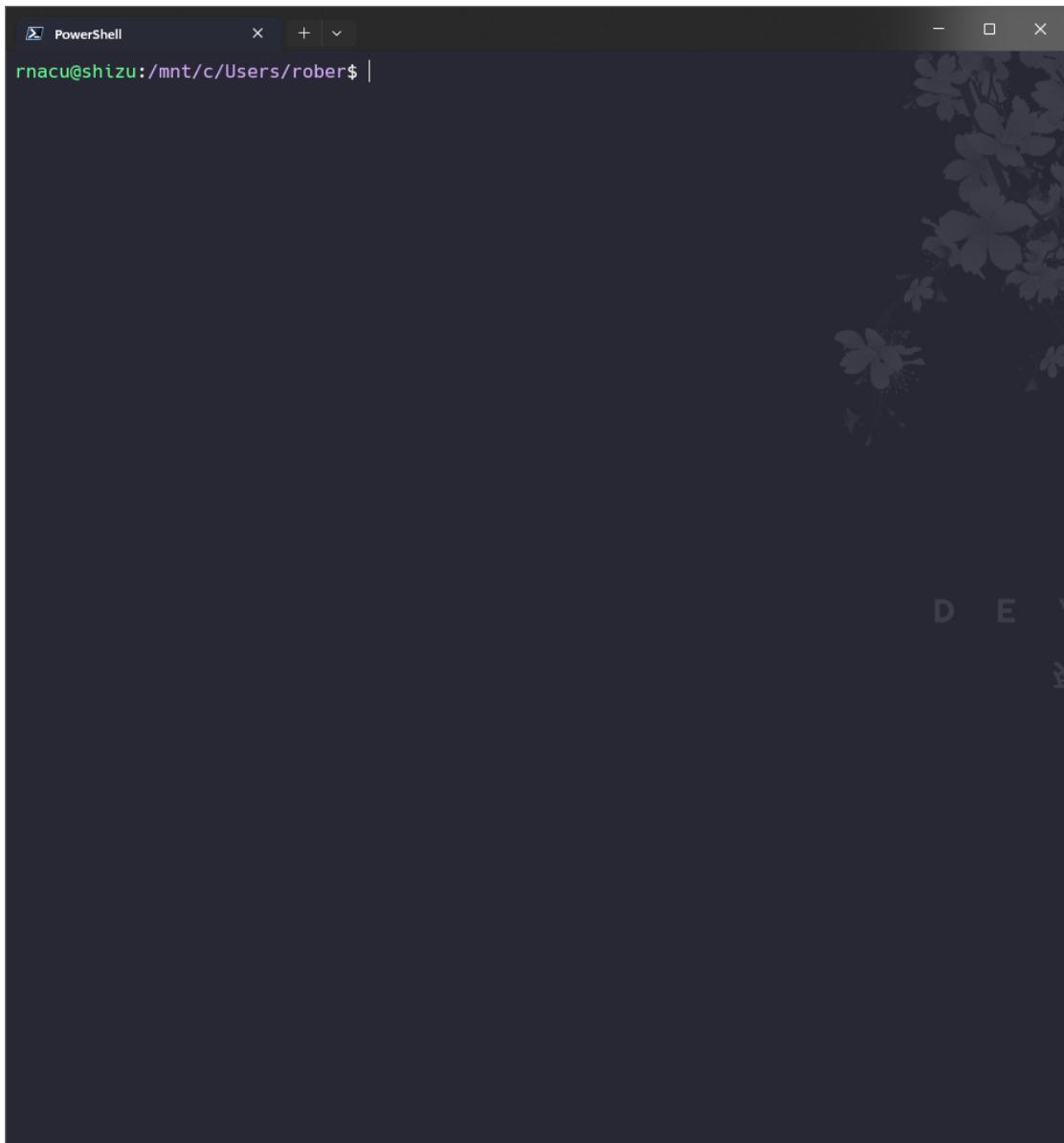
Decided on delaying the last 4 development issues since they would take relatively low amounts of time.

Main focus at this point is writing unit and integration testing for the entire application + setting up a build server using Jenkins.

Unfortunately, this pretty much means that we won't be able to achieve the MVP



Merged testing branch to dev branch



The testing won't be manually done. Instead, we will use Jenkins as a build server and run the tests in there. Installed WSL for windows

```

PowerShell          x + v - □ ×
Get:18 http://archive.ubuntu.com/ubuntu jammy/universe Translation-en [5652 kB]
Get:19 http://archive.ubuntu.com/ubuntu jammy/universe amd64 c-n-f Metadata [286 kB]
Get:20 http://archive.ubuntu.com/ubuntu jammy/multiverse amd64 Packages [217 kB]
Get:21 http://archive.ubuntu.com/ubuntu jammy/multiverse Translation-en [112 kB]
Get:22 http://archive.ubuntu.com/ubuntu jammy/multiverse amd64 c-n-f Metadata [8372 B]
Get:23 http://archive.ubuntu.com/ubuntu jammy-updates/main amd64 Packages [889 kB]
Get:24 http://archive.ubuntu.com/ubuntu jammy-updates/main Translation-en [214 kB]
Get:25 http://archive.ubuntu.com/ubuntu jammy-updates/main amd64 c-n-f Metadata [15.6 kB]
Get:26 http://archive.ubuntu.com/ubuntu jammy-updates/restricted amd64 Packages [710 kB]
Get:27 http://archive.ubuntu.com/ubuntu jammy-updates/restricted Translation-en [113 kB]
Get:28 http://archive.ubuntu.com/ubuntu jammy-updates/restricted amd64 c-n-f Metadata [536 B]
]
Get:29 http://archive.ubuntu.com/ubuntu jammy-updates/universe amd64 Packages [965 kB]
Get:30 http://archive.ubuntu.com/ubuntu jammy-updates/universe Translation-en [210 kB]
Get:31 http://archive.ubuntu.com/ubuntu jammy-updates/universe amd64 c-n-f Metadata [21.5 kB]
]
Get:32 http://archive.ubuntu.com/ubuntu jammy-updates/multiverse amd64 Packages [41.6 kB]
Get:33 http://archive.ubuntu.com/ubuntu jammy-updates/multiverse Translation-en [9768 B]
Get:34 http://archive.ubuntu.com/ubuntu jammy-updates/multiverse amd64 c-n-f Metadata [476 B]
]
Get:35 http://archive.ubuntu.com/ubuntu jammy-backports/main amd64 Packages [40.8 kB]
Get:36 http://archive.ubuntu.com/ubuntu jammy-backports/main Translation-en [10.2 kB]
Get:37 http://archive.ubuntu.com/ubuntu jammy-backports/main amd64 c-n-f Metadata [388 B]
Get:38 http://archive.ubuntu.com/ubuntu jammy-backports/restricted amd64 c-n-f Metadata [116 B]
]
Get:39 http://archive.ubuntu.com/ubuntu jammy-backports/universe amd64 Packages [22.3 kB]
Get:40 http://archive.ubuntu.com/ubuntu jammy-backports/universe Translation-en [15.4 kB]
Get:41 http://archive.ubuntu.com/ubuntu jammy-backports/universe amd64 c-n-f Metadata [576 B]
]
Get:42 http://archive.ubuntu.com/ubuntu jammy-backports/multiverse amd64 c-n-f Metadata [116 B]
Fetched 26.5 MB in 5s (5297 kB/s)
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
81 packages can be upgraded. Run 'apt list --upgradable' to see them.

WARNING: apt does not have a stable CLI interface. Use with caution in scripts.

Extracting templates from packages: 100%
configuring jenkins user
downloading latest jenkins WAR
      % Total    % Received % Xferd  Average Speed   Time     Time     Time  Current
                                     Dload  Upload Total   Spent   Left Speed
  100  245  100  245    0     0   147      0  0:00:01  0:00:01  --:--:--  147
  100  228  100  228    0     0   126      0  0:00:01  0:00:01  --:--:--  126
    0     0     0     0    0     0      0  --:--:--  0:00:02  --:--:--     0
  10 85.3M   10 9616k    0     0   471k      0  0:03:05  0:00:20  0:02:45  554k

```

Installing Jenkins

Complete QS Sprint 2

This sprint contains 8 completed issues.

That's all of them - well done!

Complete sprint

Category	Count	Status
REVIEW	5	DONE
CATEGORY PAGE	8	DONE
CART PAGE	13	DONE
CART PAGE	34	DONE
CHECKOUT PAGE	21	DONE
PAYMENT PAGE	21	DONE
PAYMENT PAGE	13	DONE

Completed sprint 2 day 1

Welcome to Jenkins!

This page is where your Jenkins jobs will be displayed. To get started, you can set up distributed builds or start building a software project.

Start building your software project

Create a job →

Set up a distributed build

Set up an agent →

Configure a cloud →

Learn more about distributed builds ↗

Finished installing Jenkins

The screenshot shows the Jenkins configuration interface for the 'ISFA-CRUD-core-project' job. The left sidebar lists configuration sections: General (selected), Source Code Management, Build Triggers, Build Environment, Build Steps, and Post-build Actions. The 'Source Code Management' section is expanded, showing 'Git' selected as the provider. Under 'Git', the 'Repository URL' is set to `https://github.com/1391819/ISFA-CRUD-core-project.git`. The 'Credentials' dropdown is set to '- none -'. An 'Advanced' button is visible. Below this, there is an 'Add Repository' button. The 'Branches to build' section shows a single branch specifier `*/dev`, with an 'Add Branch' button below it.

Dashboard > ISFA-CRUD-core-project > Configuration

Configure

Build Steps

Execute shell

Command

See the list of available environment variables

```
sudo apt update && sudo apt install python3 python3-pip python3-venv -y  
python3 -m venv venv  
bash -c "source venv/bin/activate"  
pip3 install -r requirements.txt  
python3 -m pytest --cov --cov-report html
```

Advanced ▾

Add build step ▾

Post-build Actions

Archive the artifacts

Files to archive

```
htmlcov/*
```

Advanced ▾

Delete workspace when build is done

Advanced ▾

Add post-build action ▾

Save **Apply**

REST API Jenkins 2.419

Now, however, we need to set up our environment variables within Jenkins

Credentials

T	P	Store	Domain	ID	Name
📱	👤	System	(global)	DB_TYPE	DB_TYPE
📱	👤	System	(global)	DB_USER	DB_USER
📱	👤	System	(global)	DB_PASSWORD	DB_PASSWORD
📱	👤	System	(global)	DB_HOST	DB_HOST
📱	👤	System	(global)	SECRET_KEY	SECRET_KEY
📱	👤	System	(global)	TEST_DB_NAME	TEST_DB_NAME
📱	👤	System	(global)	DB_NAME	DB_NAME

Stores scoped to Jenkins

Here, we need to be very careful about the DB_HOST, DB_USER and DB_PASSWORD environment variables. Based on how the MySQL Server instance is setup (i.e. root might be accessible only from localhost), these env variables might end up being different from the ones we had in our initial .env file.

```
../../../../local/lib/python3.10/site-packages/flask_wtf/recaptcha/widgets.py:2
../../../../local/lib/python3.10/site-packages/flask_wtf/recaptcha/widgets.py:2
/home/jenkins/.local/lib/python3.10/site-packages/flask_wtf/recaptcha/widgets.py:2:
DeprecationWarning: 'flask.Markup' is deprecated and will be removed in Flask 2.4.
Import 'markupsafe.Markup' instead.
    from flask import Markup

-- Docs: https://docs.pytest.org/en/stable/how-to/capture-warnings.html

----- coverage: platform linux, python 3.10.12-final-0 -----
Coverage HTML written to dir htmlcov

===== short test summary info =====
FAILED tests/test_app.py::TestViews::test_add_to_cart_existing_item -
sqlalchemy.exc.OperationalError: (pymysql.err.OperationalError) (2003, "Can't connect
to MySQL server on 'localhost' ([Errno 111] Connection refused)")
(Background on this error at: https://sqlalche.me/e/20/e3q8)
FAILED tests/test_app.py::TestViews::test_add_to_cart_new_item -
sqlalchemy.exc.OperationalError: (pymysql.err.OperationalError) (2003, "Can't connect
to MySQL server on 'localhost' ([Errno 111] Connection refused)")
(Background on this error at: https://sqlalche.me/e/20/e3q8)
FAILED tests/test_app.py::TestViews::test_cart_page - sqlalchemy.exc.OperationalError:
(pymysql.err.OperationalError) (2003, "Can't connect to MySQL server on 'localhost'
([Errno 111] Connection refused)")
(Background on this error at: https://sqlalche.me/e/20/e3q8)
FAILED tests/test_app.py::TestViews::test_categories -
sqlalchemy.exc.OperationalError: (pymysql.err.OperationalError) (2003, "Can't connect
```

```
(3.0.1)
Requirement already satisfied: toml>=1.0.0 in
/home/jenkins/.local/lib/python3.10/site-packages (from pytest==7.4.0->-r
requirements.txt (line 17)) (2.0.1)
Requirement already satisfied: exceptiongroup>=1.0.0rc8 in
/home/jenkins/.local/lib/python3.10/site-packages (from pytest==7.4.0->-r
requirements.txt (line 17)) (1.1.3)
+ python3 -m pytest --cov --cov-report html
===== test session starts =====
platform linux -- Python 3.10.12, pytest-7.4.0, pluggy-1.2.0
rootdir: /home/jenkins/.jenkins/workspace/ISFA-CRUD-core-project
plugins: cov-4.1.0
collected 20 items

tests/test_app.py ..... [100%]

=====
warnings summary =====
../../../../../local/lib/python3.10/site-packages/flask_wtf/recaptcha/widgets.py:2
../../../../../local/lib/python3.10/site-packages/flask_wtf/recaptcha/widgets.py:2
/home/jenkins/.local/lib/python3.10/site-packages/flask_wtf/recaptcha/widgets.py:2
DeprecationWarning: 'flask.Markup' is deprecated and will be removed in Flask 2.4.
Import 'markupsafe.Markup' instead.
    from flask import Markup

-- Docs: https://docs.pytest.org/en/stable/how-to/capture-warnings.html

----- coverage: platform linux, python 3.10.12-final-0 -----
Coverage HTML written to dir htmlcov

=====
20 passed, 2 warnings in 5.29s =====
Archiving artifacts
[WS-CLEANUP] Deleting project workspace...
[WS-CLEANUP] Deferred wipeout is used...
[WS-CLEANUP] done
Finished: SUCCESS
```

After a lot of troubleshooting, we successfully set up the job on Jenkins.

```

test_app.py | C:\Users\rober\Documents\MEGA\work\AMIS\ISFA-CRUD-core-project
-----+
  Explorer    File  README.md  test_app.py  env
  Tests      test_app.py  TestBase  create_app
  Application application
  Routes     routes.py
  Templates  templates
  Models    models.py
  HTML/CSS   htmvcov
  Instance  instance
  Tests     tests
  Coverage  coverage
  Env      env
  Ignore   ignore
  App     app.py
  Create  create.py
  Requirements requirements.txt
  README.md
  Requirements.txt

1 from flask import Flask
2 from flask_testing import TestCase
3
4 from flask import url_for
5
6
7 # create base class
8 class TestBase(TestCase):
9     def create_app(self):
10         # pass in testing configurations for the app
11         # we shouldn't use production database but testing database
12         # a new one should be created (we will just use an sqlite one)
13
14         # TODO: Check __init__.py under application to find out why this is
15         app.config.update(
16             SQLALCHEMY_DATABASE_URI="sqlite:///testdata.sqlite",
17             TESTING=True,
18             WTF_CSRF_ENABLED=False,
19         )
20
21         # print(app.config["SQLALCHEMY_DATABASE_URI"]) # debug
22
23         return app
24
25     # this will be called before every test
26     def setup(self):
27         # create tables
28         db.create_all()
29
30     # this will be called before every test
31     def teardown(self):
32         # drop tables
33         db.drop_all()
34
35     # creating testing categories and products
36
37     # this is creating a simulation of the database as an object in our application
38     db = SQLAlchemy(app)

PROBLEMS 1 OUTPUT DEBUG CONSOLE TERMINAL
(venv) rober@elizius MINGW64 ~/Documents/MEGA/work/AMIS/ISFA-CRUD-core-project (dev)
# [1]
(venv) rober@elizius MINGW64 ~/Documents/MEGA/work/AMIS/ISFA-CRUD-core-project (dev)
$ [1]

In 24, Col 63  Spaces: 4  UTF-8  CR/LF  Python 3.11.4 (venv) venv  Port: 5500  Prettier

```

A lot of problems happened regarding the creation of a testing database. Unfortunately, due to time constraints, I was not able to log this into the Kanban board.

PROBLEM: For some reason, pytest does not want to use the test database and keeps performing the tests using the production database.

- In order to fix this (and due to time constraints), 1) I could not log this issue into the Kanban board, and 2) I had to set a **FLASK** environment variable which specifies whether to use the production database or the testing database.
- This is something which must be further looked into since there could be problems regarding sessions or contexts, which are particularly useful within Flask.

Global credentials (unrestricted)			
ID	Name	Kind	Description
DB_TYPE	DB_TYPE	Secret text	
DB_USER	DB_USER	Secret text	
DB_PASSWORD	DB_PASSWORD	Secret text	
DB_HOST	DB_HOST	Secret text	
SECRET_KEY	SECRET_KEY	Secret text	
DB_NAME	DB_NAME	Secret text	
FLASK	FLASK	Secret text	

Updated credentials within Jenkins. Be careful, they are global

The screenshot shows the Jenkins build results for build #24. The top navigation bar includes 'Dashboard > ISFA-CRUD-core-project > #24'. The main header displays '#24 (15 Aug 2023, 21:00:18)' with a green checkmark icon. To the right are buttons for 'Keep this build forever' and a link to 'Add description'. Below the header, there's a sidebar with links like 'Status', 'Changes', 'Console Output', 'Edit Build Information', 'Delete build #24', 'Git Build Data', and 'Previous Build'. The main content area starts with a 'Build Artifacts' section containing a list of files with their sizes and 'view' links. Below this is a 'Changes' section listing a single commit: 'RN: added comments'. Further down are sections for 'Started by user' (with a person icon), 'Revision' (6d119ea9cf...), 'Repository' (https://github.com/1391819/ISFA-CRUD-core-project.git), and a note about refs/remotes/origin/dev.

Now Jenkins uses a test database instead of the original one, and the test results can be found within the post-build artifacts.

The screenshot shows the Jenkins project configuration page. On the left, a sidebar lists 'Configure' sections: General, Source Code Management, Build Triggers (which is selected and highlighted in blue), Build Environment, Build Steps, and Post-build Actions. The 'Build Triggers' section contains several checkboxes: 'Trigger builds remotely (e.g., from scripts)', 'Build after other projects are built', 'Build periodically', and 'GitHub hook trigger for GITScm polling' (which is checked). Below this is a 'Build Environment' section with a note 'Setting up GitHub webhook'.

Problem: Jenkins is running on local machine

→ it's not accessible over the public network. There are many workarounds for this (e.g., setting up port forwarding or additional DNS)

settings within our router, but it would take a long time)

→ We give up on configuring webhooks for now, but we keep the job setting on for future work

Unfortunately time is limited, and now we need to work on the presentation and documentation. The below is what is left from our initial issues. Most of them are low in priority so they wouldn't require much work. However, given that it's an Agile project, things should be continuously iterated over.

The screenshot shows a Jira backlog board with the following structure:

- Sprint 3 (15 Aug – 15 Aug):** Contains 2 issues.
 - QS-19: As a user, I want to have a contact form to submit enquiries or provide feedback to the store.
 - QS-18: As a user, I want to easily find contact information for the store, including address, phone number, and email.
- Backlog (10 issues):** Contains 10 issues categorized by priority:
 - 19 (High Priority):
 - QS-27: As a user, I want to have a smooth and seamless browsing experience without delays caused by slow loading times.
 - QS-24: As a user, I want the site's design to align with the brand's identity and appeal to my preferences.
 - QS-25: As a user, I expect the site to use appropriate colours, fonts, and imagery that create a cohesive and engaging experience.
 - QS-20: As a user, I want to access a page that provides information about the store's mission, history, and the team behind it.
 - QS-21: As a user, I want to learn about the people responsible for running the store.
 - 18 (Medium Priority):
 - CONTACT US PAGE
 - ABOUT US PAGE
 - 17 (Low Priority):
 - TO DO
 - CONTACT US PAGE
 - ABOUT US PAGE
 - 16 (Very Low Priority):
 - TO DO
 - CONTACT US PAGE
 - ABOUT US PAGE