



MARCH 25, 2025

S3 SUBMISSION

DOCUMENTATION

CASEY SHEA (BSC COM SC W DTP FT DA (PWC))
UNIVERSITY OF BIRMINGHAM

Table of Contents

Entity Development.....	2
Emissions	2
Insights Dashboard View.....	3
Git Commits and Kanban Evidence	4
Demo Preparation	5
App Demo Script	5
A11y App Demo Script	6
Slides.....	7
App Demo	7
A11y App Demo	8
Time Sheets	9

Entity Development

Emissions

Overview of the Entity and Its Purpose

The **Emissions** entity tracks CO₂ emissions for each trip users log in the system. It automatically calculates environmental impact based on distance and transport mode and sends this data to the **Insights Dashboard**. This allows users to monitor and reflect on their carbon footprint, reinforcing sustainable choices.

(Please note: Only the top metrics bar is currently working with the backend for Emissions. The rest is still static data as I haven't had time to code out the rest of the logic)

User-Facing Features and Frontend Components

The *Emissions* functionality directly feeds into the **Insights Dashboard**, which includes:

- **CO₂ Total** – shows total emissions over a time period
- **Green Journey Count** – counts low-emission trips (e.g. walking, cycling)
- **Rating Breakdown** – categorizes emissions as Excellent, Good, or Poor
- **Time Period Filter** – switch between weekly, monthly, and yearly views
- **Dynamic Charts:**
 - Pie chart: Trip ratings distribution
 - Line chart: Emissions trend over time
- **Smart Recommendations:** Based on high-emission trips, suggest greener alternatives

Frontend Code Components:

- insights-dashboard.component.ts – API integration
- insights-dashboard.component.html – metric cards, filter UI

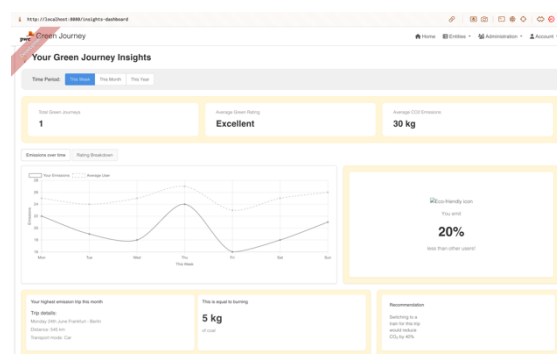


Figure 1 - Weekly Insights Dashboard with Metrics Display

Backend Constraints and Business Logic

- **Auto Create/Update/Delete:** Uses Spring event listeners via TripStorageEvent
- **CO₂ Calculations:** Based on transport mode (e.g., Car = 192g/km, Train = 41g/km)
- **Rating System:** Based on emissions per km

- **Hourly Sync Job:** Checks for missing emissions
- **Daily Cleanup Job:** Removes orphaned emissions
- **Data Validation:** Emissions only saved if trip + user are valid

Backend Code Components:

- EmissionsService.java (jHipster)– emissions logic and schedulers
- TripStorageService.java (jHipster)– includes event scheduling logic based on trip record changes
- InsightsDashboardService.java (self-created) – aggregates emissions for the dashboard
- EmissionsRepository.java (jHipster)– custom queries (e.g. orphan detection)

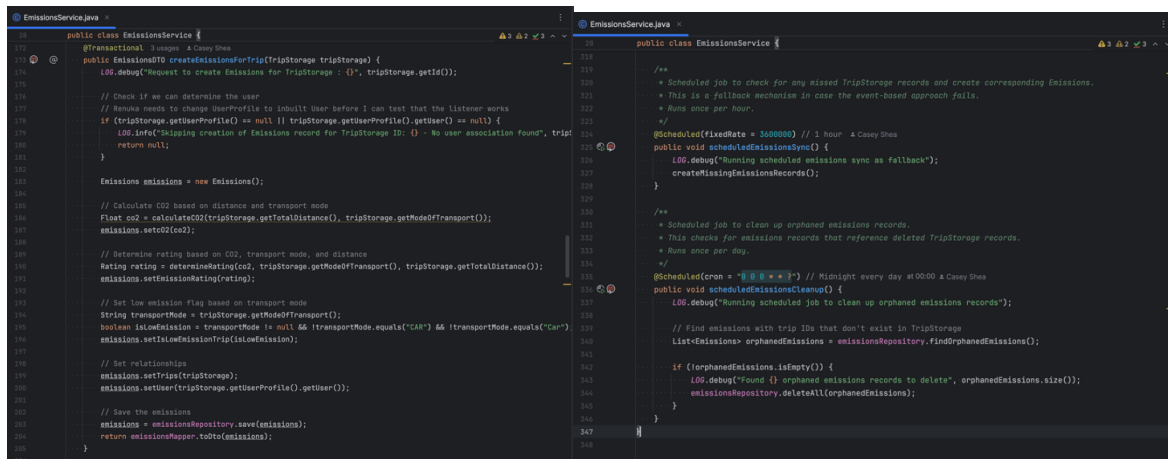


Figure 2 - createEmissionsForTrip() Logic in EmissionsService. Figure 3 - Logic for scheduled cleanup tasks in EmissionsService

CRUD Operations and End-User Interactions

Operation	Description	Trigger
Create	Emissions are automatically generated when a user logs a new trip	User creates a trip via Trip form
Read	Emissions are visualised in the Dashboard	User views their insights
Update	When trip details (e.g., distance or mode) are edited, emissions are recalculated	User edits a trip
Delete	If a trip is deleted, its associated emissions are also removed	User deletes a trip

Fallback logic also handles:

- Generating emissions for missed trips (hourly job)
- Removing emissions from deleted trips (daily job)

Insights Dashboard View

Because the Emissions entity itself is not directly user-editable in a typical CRUD sense, I implemented a supporting entity called InsightsDashboardView to enable future user-

driven configuration of the dashboard. This entity will allow users to **save personalized view preferences**, such as their default time period (week/month/year) or preferred chart type (e.g., pie or line). In the future, users will be able to **create, update, and delete multiple view presets**, enabling more tailored and accessible insights.

For now, the entity has been scaffolded and integrated with backend services and is ready to support future frontend CRUD operations when the feature is extended.

```
@dataclass
@service(serviceClass)
@paginate(pagination)
@filter
entity InsightsDashboardView {
  name String required,
  defaultTimeView TimeView required,
  defaultChartView ChartView required
}

enum TimeView {
  WEEK ("This Week"),
  MONTH ("This Month"),
  YEAR ("This Year")
}

enum ChartView {
  EMISSIONS_OVER_TIME ("Emissions over time"),
  RATING_BREAKDOWN ("Rating Breakdown")
}

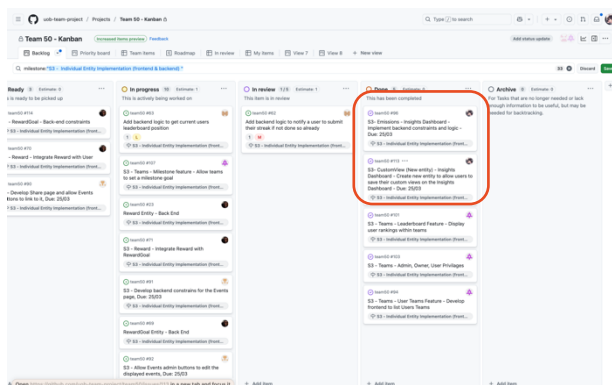
relationship ManyToOne {
  InsightsDashboardView to User with builtInEntity
```

Entity Fields:

- defaultTimeView: Stores the user's preferred time period
- defaultChartView: Stores the preferred chart style
- name: Allows naming of different view configurations
- Associated User: Each saved view is tied to a specific user

Git Commits and Kanban Evidence

Kanban



 [Team50 Kanban](#)

Key Commits



 [Emissions Backend](#)

 [Insights Dashboard View Entity Generation](#)

Demo Preparation

App Demo Script

Hello, my name is Casey Shea, and my primary contribution to the project has been the development of the *Emissions* entity and the implementation of the Insights Dashboard.

The Emissions feature is essential to the app's sustainability mission. It automatically calculates CO₂ output for each user journey based on transport mode and distance. This data is stored, rated, and visualised in the dashboard to help users reflect on their environmental impact.

The dashboard itself includes several dynamic features. First, users can filter their insights by time period - weekly, monthly, or yearly - using accessible toggle buttons. They can then view real-time metrics such as total green journeys taken, average emissions rating, and total CO₂ emitted.

Users can also switch between two chart views: a **line graph** comparing their own emissions to the average user over time, and a **pie chart** showing the breakdown of trip ratings.

To further support behaviour change, the dashboard highlights each user's **highest-emission trip** for the current month. It includes a carbon equivalency - for example, "5kg of coal" - and follows up with a personalised recommendation, such as switching to a lower-emission transport mode like train.

I also implemented a comparative view showing how much less the user emits compared to others, such as "You emit 20% less than other users," to reinforce positive behaviour.

On the backend, I used Spring Boot's event system to automate emissions lifecycle updates when trips are created, updated, or deleted. The frontend dashboard is powered by dynamic data calls, and I've also scaffolded a new entity, *InsightsDashboardView*, to allow users to save personalised layouts and viewing preferences in future versions.

A11y App Demo Script

Following on the Insights Dashboard functionality I just described, I'd like to briefly highlight the accessibility features and inclusive design principles integrated into the interface.

From the start, I've aimed to make this dashboard usable and informative for all users, regardless of device or ability. The layout follows a clear and predictable structure with semantic HTML and screen reader-friendly labelling on key data points—like CO₂ totals, trip ratings, and comparison insights.

Interactive elements, such as the time filter buttons, are fully keyboard-accessible, and the dashboard maintains a logical tab order to ensure smooth navigation. The graph and pie chart areas also include descriptive legends and ARIA attributes to help users understand the data visually and non-visually.

Visually, the dashboard uses large, readable fonts and high-contrast elements, particularly for key stats and recommendation cards. This ensures readability for users with low vision or contrast sensitivity.

The layout is responsive and adapts to mobile and tablet screens, so users can comfortably interact with the dashboard across devices without needing to zoom or scroll horizontally. I've also enabled support for **dark mode**, which can help reduce eye strain, especially for evening or mobile use.

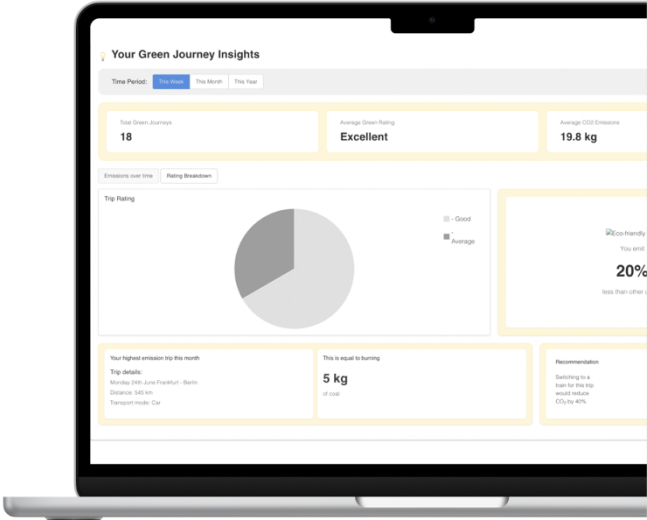
Looking ahead, I've introduced the *InsightsDashboardView* entity to support personalised configurations. This will allow users to set default layouts—such as simplified views without charts, or high-contrast themes—making the experience more tailored and inclusive.

Slides

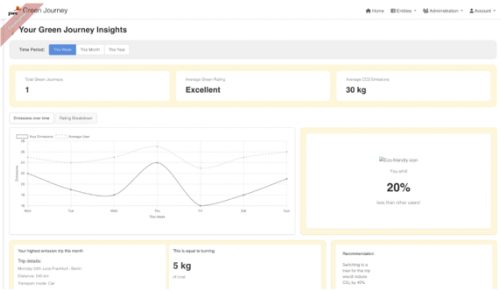
App Demo

Insights Dashboard

This dashboard transforms travel activity into meaningful, real-time environmental insights.

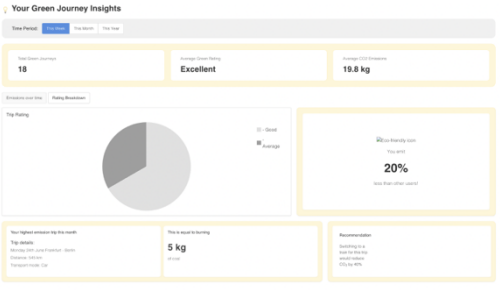


Let's take a closer look at the functionality



Emissions over time view

View carbon emissions over time and filter by time period.



Rating breakdown view

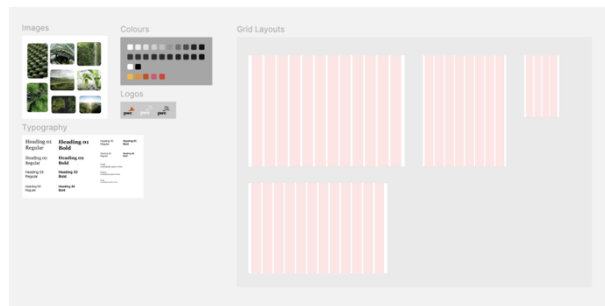
View rating break down based on the journeys you've made.

A11y App Demo

Accessibility

From screen readers, to keyboard accessibility, to dark mode, the dashboard supports all these features

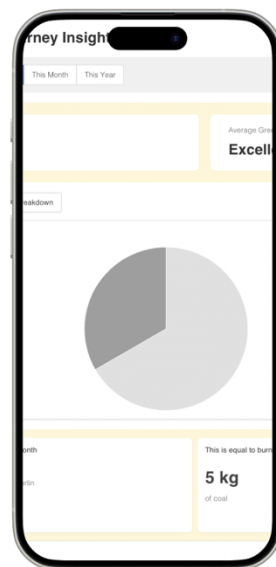
(This is not done yet so please enjoy my design file for the UI for now 🤔)



Mobile View

View the dashboard comfortably on mobile

(I have not implemented this yet as you can tell 🤔)



Time Sheets

Team sheet Number/ID: Team 50

~up to 18 hours expected per week

Team member name: Casey Shea

Team meeting sign off date: _____

Date until: 25/03/2025

Task	Date	Start time	End time	Total Hours
App design research for new UI	20/03/2025	3:00 PM	5:00 PM	2:00
Create design system in Figma	23/03/2025	2:00 PM	6:00 PM	4:00
Edit emissions entity and create new entity in JDL	24/03/2025	9:00 AM	10:00 AM	1:00
Regenerate app with JDL	24/03/2025	10:00 AM	12:00 PM	2:00
Work on backend logic for emissions	24/03/2025	1:00 PM	9:00 PM	8:00
S3 Submisison drafting, editing and finalisation	25/03/2025	9:00 AM	2:30 PM	5:30
				0:00

Total Hours 22:30