GRIFFITH COLLEGE DUBLIN

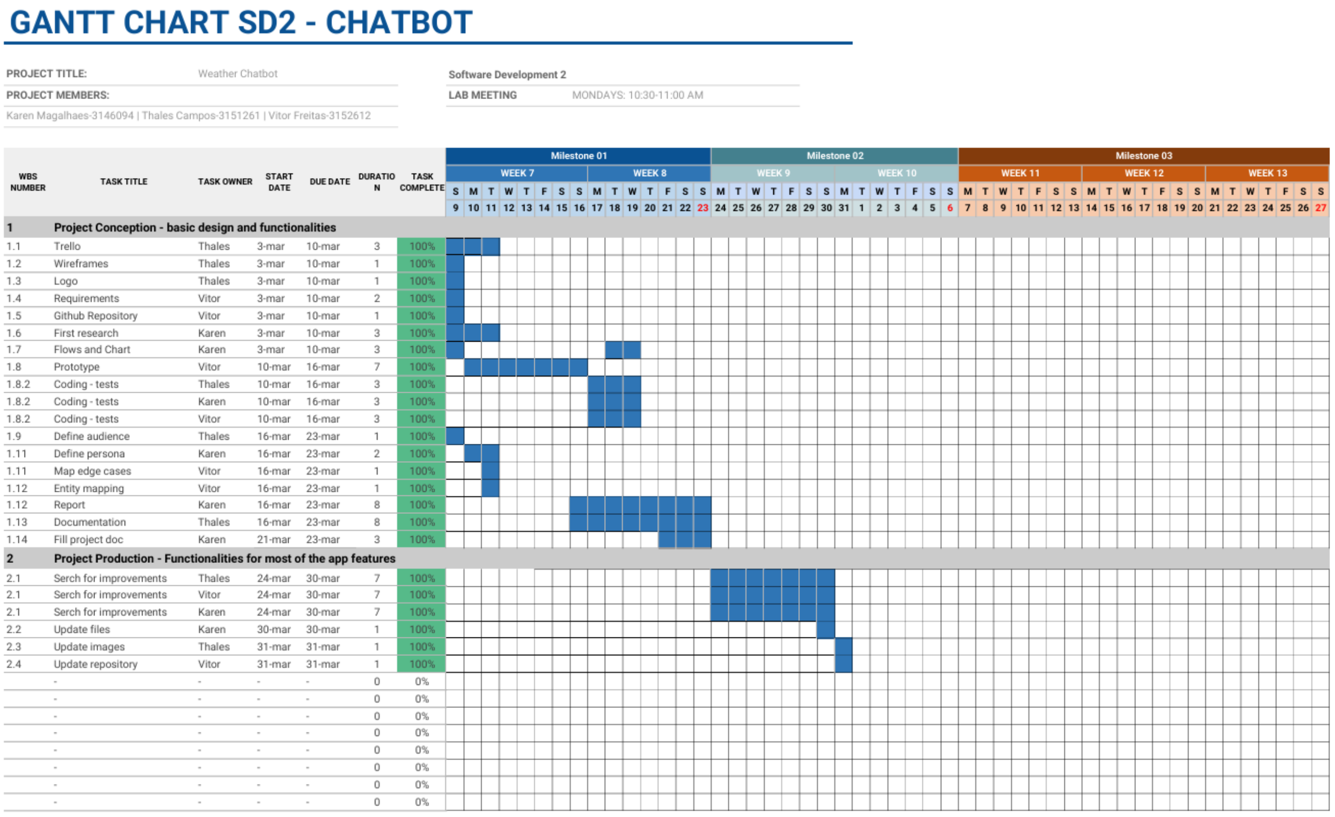
Student name:	Student 1: Karen Ferreira Magalhaes Student 2: Thales Campos Student 3: Vitor Freitas				
Student number:	Student 1: 3146094 Student 2: 3151261 Student 3: 3152612				
Faculty:	Computing Science				
Course:	BSCH/BSCO/EXCH		Stage/year:	2	
Subject:	Software Development 2				
Study Mode:	Full time	✓	Part-time		
Lecturer Name:	Haseeb Younis/ Muhammad Shoaib				
Assignment Title:	Project Final Documentation				
Date due:	27/04/2025				
Date submitted:	23/03/2025				

Software Development 2
BSCH-SD2
Chatbot Project

Table of Contents

1. Versioning Approach

2. Development Process



GANTT CHART SD2 - CHATBOT

PROJECT TITLE:	Weather Chatbot	Software Development 2	
PROJECT MEMBERS:		LAB MEETING	MONDAYS: 10:30-11:00 AM
Karen Magalhaes-3146094 | Thales Campos-3151261 | Vitor Freitas-3152612			

Milestone 01: WEEK 7, WEEK 8 · Milestone 02: WEEK 9, WEEK 10 · Milestone 03: WEEK 11, WEEK 12, WEEK 13

WBS NUMBER	TASK TITLE	TASK OWNER	START DATE	DUE DATE	DURATION	TASK COMPLETE
1	**Project Conception - basic design and functionalities**					
1.1	Trello	Thales	3-mar	10-mar	3	100%
1.2	Wireframes	Thales	3-mar	10-mar	1	100%
1.3	Logo	Thales	3-mar	10-mar	1	100%
1.4	Requirements	Vitor	3-mar	10-mar	2	100%
1.5	Github Repository	Vitor	3-mar	10-mar	1	100%
1.6	First research	Karen	3-mar	10-mar	3	100%
1.7	Flows and Chart	Karen	3-mar	10-mar	3	100%
1.8	Prototype	Vitor	10-mar	16-mar	7	100%
1.8.2	Coding - tests	Thales	10-mar	16-mar	3	100%
1.8.2	Coding - tests	Karen	10-mar	16-mar	3	100%
1.8.2	Coding - tests	Vitor	10-mar	16-mar	3	100%
1.9	Define audience	Thales	16-mar	23-mar	1	100%
1.11	Define persona	Karen	16-mar	23-mar	2	100%
1.11	Map edge cases	Vitor	16-mar	23-mar	1	100%
1.12	Entity mapping	Vitor	16-mar	23-mar	1	100%
1.12	Report	Karen	16-mar	23-mar	8	100%
1.13	Documentation	Thales	16-mar	23-mar	8	100%
1.14	Fill project doc	Karen	21-mar	23-mar	3	100%
2	**Project Production - Functionalities for most of the app features**					
2.1	Serch for improvements	Thales	24-mar	30-mar	7	100%
2.1	Serch for improvements	Vitor	24-mar	30-mar	7	100%
2.1	Serch for improvements	Karen	24-mar	30-mar	7	100%
2.2	Update files	Karen	30-mar	30-mar	1	100%
2.3	Update images	Thales	31-mar	31-mar	1	100%
2.4	Update repository	Vitor	31-mar	31-mar	1	100%
	-	-	-	-	0	0%
	-	-	-	-	0	0%
	-	-	-	-	0	0%
	-	-	-	-	0	0%
	-	-	-	-	0	0%
	-	-	-	-	0	0%
	-	-	-	-	0	0%
	-	-	-	-	0	0%

Figure 1 - Gantt Chart

Figure 2 - Chatbot logo designed using ChatGPT, representing the project's identity and visual branding.

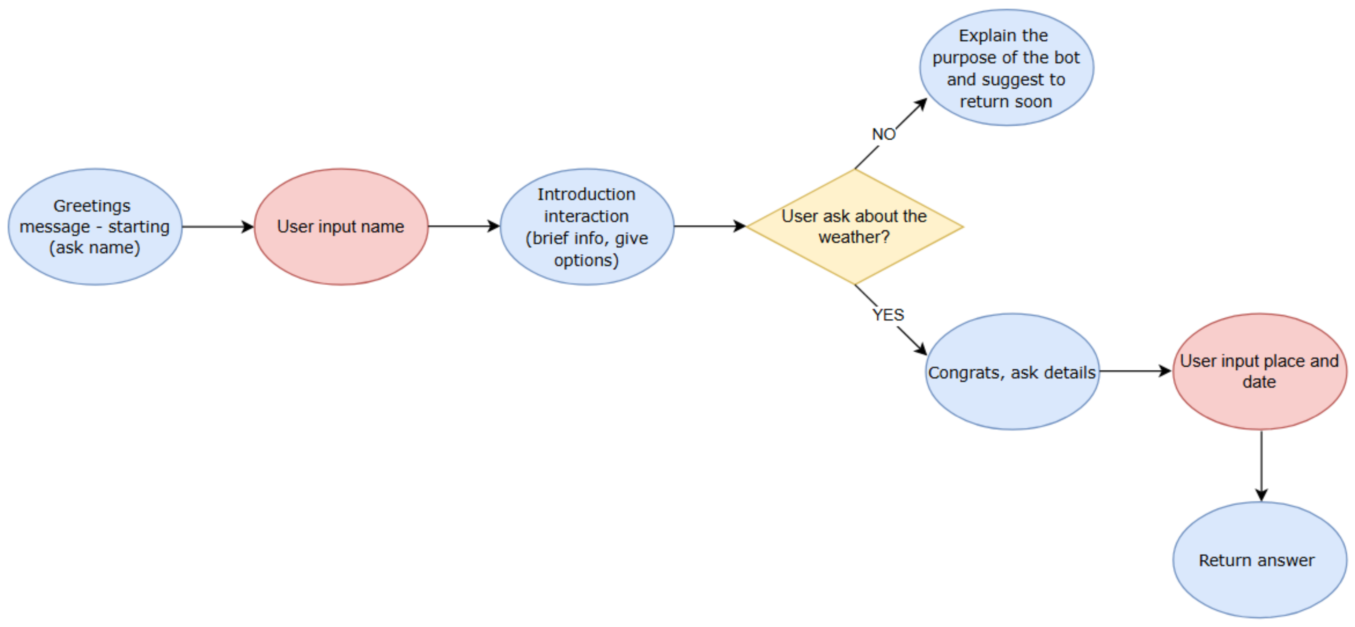
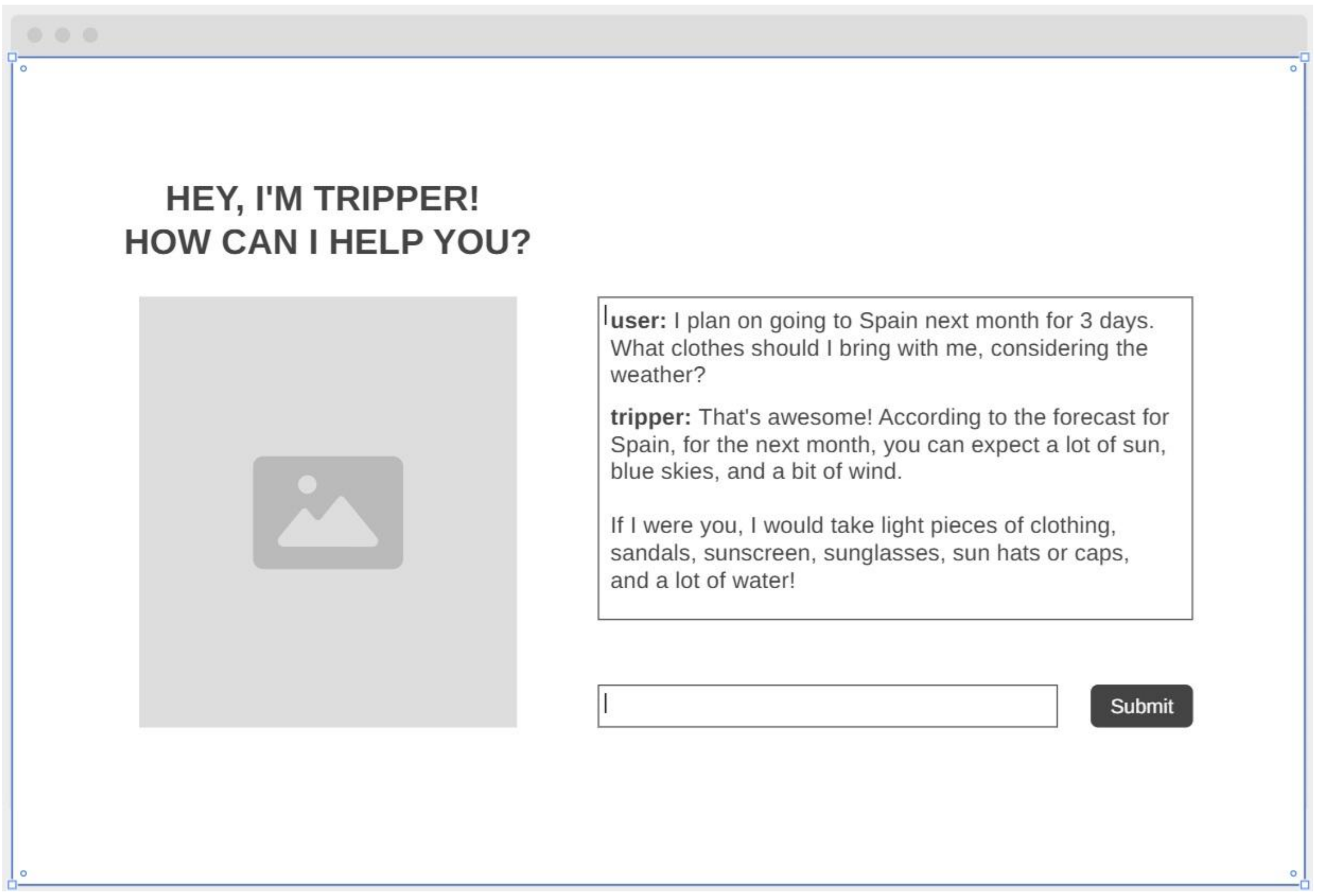

Greetings message - starting (ask name)
User input name
Introduction interaction (brief info, give options)
User ask about the weather?
NO
Explain the purpose of the bot and suggest to return soon
YES
Congrats, ask details
User input place and date
Return answer

Figure 3 - Simplified flowchart illustrating user interaction with the chatbot, from message input to response generation.



user: I plan on going to Spain next month for 3 days. What clothes should I bring with me, considering the weather?

tripper: That's awesome! According to the forecast for Spain, for the next month, you can expect a lot of sun, blue skies, and a bit of wind.

If I were you, I would take light pieces of clothing, sandals, sunscreen, sunglasses, sun hats or caps, and a lot of water!

Figure 4 - Wireframe design of the chatbot interface, showcasing the layout and user interaction flow.

ChatGPTClient Class Documentation

Overview: The ChatGPTClient class handles communication with the OpenAI ChatGPT API. It sends user messages, receives AI-generated responses, and processes the API response using the Gson library.

Package: main.java.com.tripper

Dependencies:

- java.io.* (For input/output operations)
- java.net.* (For handling HTTP connections)
- com.google.gson.* (For JSON processing)

Constants:

- String API_URL: The endpoint URL for OpenAI's ChatGPT API.
- String API_KEY: The API key used for authentication (should be kept secret and not hardcoded in production).

Methods:

1. String getChatResponse(String conversationContext):
    - Sends a user message to the OpenAI ChatGPT API and retrieves the AI-generated response.
    - Parameters:
        - conversationContext: The user's input message to be processed by the AI.

    - Returns:
        - A string containing the AI-generated response.
    - Process:
        - Establishes an HTTP connection with the OpenAI API.
        - Constructs a JSON payload containing the request data.
        - Sends the request and reads the API response.
        - Parses the JSON response to extract the AI's reply.
        - Returns the processed response or an error message if the API call fails.

Exception Handling:

- Catches general exceptions during the API request and response handling.
- Prints stack traces for debugging and returns an error message when an issue occurs.

Security Considerations:

- The API key should not be stored in the source code; it should be loaded from environment variables or secure storage in production.
- Ensure proper exception handling to avoid exposing sensitive information in error messages.

ConversationController Class Documentation

Overview: The ConversationController class manages the chatbot's interaction flow, handling user input, processing responses, and guiding the conversation through predefined states.

Package: main.java.com.tripper

Dependencies:

- java.util.Scanner (For user input handling)

Attributes:

- ConversationState state: Stores conversation progress and user data.
- ConversationManager conversationManager: Manages chatbot responses.
- ChatGPTClient chatGPTClient: Communicates with OpenAI's API.
- Scanner scanner: Handles user input from the console.

Methods:

1. void run():
    - Manages the chatbot's state-based conversation flow.
    - Guides the user through different states: GREETING, COLLECT_TRIP_DETAILS, CONFIRM_DETAILS, GENERATE_RECOMMENDATIONS, OFFER_PDF, END.
    - Integrates responses from ChatGPTClient.
    - Handles user inputs for trip details and PDF generation.
2. void typePrint(String message, int delayMs):
    - Simulates a typing effect when displaying chatbot messages.
    - Parameters:
        - message: The text to display.
        - delayMs: Delay per character in milliseconds.
3. String prompt(String message):
    - Displays a message and collects user input.
    - Parameters:

        - message: The prompt for the user.
    - Returns:
        - User input as a trimmed string.

Exception Handling:

- Ensures smooth conversation flow by handling user input errors.
- Uses try-catch to handle interruptions in the typing effect.

Functionality:

- Starts the chatbot with a greeting message.
- Collects and processes user trip details.
- Requests recommendations from ChatGPTClient.
- Offers an option to generate a PDF checklist.
- Ends the conversation with a farewell message.

ConversationManager Class Documentation

Overview: The ConversationManager class handles the chatbot's predefined responses to user input. It generates friendly and engaging messages to guide users through the interaction process.

Package: main.java.com.tripper

Methods:

1. String getGreeting(String userName):
    - Generates a personalized greeting for the user.
    - Parameters:
        - userName: The name of the user.
    - Returns:
        - A friendly welcome message including the user's name.
2. String askForTripDetails():
    - Prompts the user to provide details about their trip.
    - Returns:
        - A request message asking the user for trip information.
3. String friendlyResponse(String dynamicResponse):
    - Formats and returns a friendly response incorporating dynamically generated recommendations.
    - Parameters:
        - dynamicResponse: The AI-generated travel recommendations.
    - Returns:
        - A structured response including the recommendations.

Functionality:

- Ensures a smooth and engaging chatbot experience.
- Provides user-friendly prompts and structured responses to enhance interaction.
- Acts as an intermediary between user input and AI-generated recommendations.

ConversationState Class Documentation

Overview: The ConversationState class maintains the state of a chatbot conversation. It stores user-specific data such as name, trip details, and confirmation status to ensure a smooth interaction flow.

Package: main.java.com.tripper

Dependencies:

- java.util.ArrayList (For handling dynamic lists)
- java.util.List (For managing location storage)

Attributes:

- String userName: Stores the user's name.
- List locations: Holds the list of travel locations provided by the user.
- String tripDetails: Contains user-supplied details about the trip.
- boolean detailsConfirmed: Indicates whether the trip details have been confirmed by the user.

Methods:

1. ConversationState():
    - Constructor that initializes the locations list and sets detailsConfirmed to false.
2. String getUserName():
    - Retrieves the user's name.
    - Returns:
        - The name of the user.
3. void setUserName(String userName):
    - Sets the user's name.
    - Parameters:
        - userName: The name of the user.
4. List getLocations():
    - Retrieves the list of locations.
    - Returns:
        - A list of location names.
5. void addLocation(String location):
    - Adds a new location to the list.
    - Parameters:
        - location: The name of the location to add.
6. String getTripDetails():
    - Retrieves the trip details provided by the user.
    - Returns:
        - The trip details as a string.
7. void setTripDetails(String tripDetails):
    - Sets the trip details.
    - Parameters:
        - tripDetails: A string containing trip information.
8. boolean isDetailsConfirmed():
    - Checks if the trip details have been confirmed by the user.
    - Returns:
        - True if confirmed, false otherwise.
9. void setDetailsConfirmed(boolean detailsConfirmed):

    - Updates the confirmation status of the trip details.
    - Parameters:
        - detailsConfirmed: Boolean value indicating confirmation status.

Functionality:

- Tracks user input and progress throughout the chatbot conversation.
- Stores key trip-related information for personalized recommendations.
- Ensures state persistence for an improved chatbot experience.

InputParser Class Documentation

Overview:

The InputParser class processes and extracts travel-related information from user input. It uses regular expressions and string manipulation to identify the travel month and potential locations provided by the user.

Package:

main.java.com.tripper

Dependencies:

- java.util.* (For handling collections and arrays)
- java.util.regex.* (For regular expression matching)

Constants:

- Pattern monthPattern: A regular expression pattern to match any month name (case-insensitive).

Methods:

1. TripDetails parseTripDetails(String input):
    - Parameters:
        - input: The user-provided string containing trip details.
    - Returns:
        - A TripDetails object containing the extracted travel month and locations.
    - Process:
        - Uses regex to identify a month in the input and sets the travel month in the TripDetails object.
        - Splits the input into tokens and identifies capitalized words (excluding month names) as potential locations.
        - Removes punctuation from the tokens and adds valid locations to the list.

Functionality:

- Identifies and extracts the travel month and location information from a free-form user input string.
- Uses regular expressions for month matching and simple string checks for locations.
- Ensures proper location parsing even if months and locations are mixed in the input.

NLPInputParser Class Documentation

Overview:

The NLPInputParser class is responsible for processing input text using natural language processing (NLP) techniques. It uses OpenNLP tools to perform sentence detection, tokenization, part-of-speech (POS) tagging, and optional lemmatization. The primary goal is to parse trip details, such as locations and travel dates, from user input.

Dependencies:

- OpenNLP Library (tools for sentence detection, tokenization, POS tagging, and lemmatization).

Attributes:

- SentenceDetectorME sentenceDetector: Detects sentence boundaries in the input text.
- TokenizerME tokenizer: Tokenizes the text into individual words.
- POSTaggerME posTagger: Tags each token with its part-of-speech.
- DictionaryLemmatizer lemmatizer (optional): Lemmatizes tokens to their base forms.

Constructor:

- Initializes models for sentence detection, tokenization, POS tagging, and lemmatization (if available).
- Loads model files from the local file system for OpenNLP tools.

Methods:

1. parseTripDetails(String input):
   - Parameters: String input – The input text containing trip details.
   - Returns: TripDetails – A TripDetails object containing locations and the travel month parsed from the input.
   - Description:
     - Detects sentences in the input.
     - Tokenizes sentences into words.
     - Performs POS tagging on the tokens.
     - Identifies proper nouns (NNP/NNPS tags) as potential location names.
     - Identifies date-like tokens (e.g., "12/12/2025" or month names).
     - Collects and returns the locations and travel month as part of the TripDetails object.

Security Considerations:

- Ensure the model files are securely stored and not exposed to unauthorized access.
- Handle any exceptions that might occur during model loading or processing.

PDFGenerator Class Documentation

Overview:

The PDFGenerator class is responsible for creating a custom PDF document containing a checklist for travel preparation. It includes multiple sections such as trip details, essential items, recommendations, optional items, and accessories. The class uses the Apache PDFBox library to generate the PDF.

Dependencies:

- Apache PDFBox (used for PDF creation and manipulation)

Methods:

1. generateChecklist(String fileName, ConversationState state, String tripDetails, String[] essentialItems, String[] recommendations, String[] optionalItems, String[] accessories):

- Parameters:
    - fileName: The name of the output PDF file.
    - state: The ConversationState object holding user details such as name.
    - tripDetails: A string representing trip details (e.g., locations).
    - essentialItems: An array of essential clothing items.
    - recommendations: An array of recommended items for the trip.
    - optionalItems: An array of optional items.
    - accessories: An array of accessory items for the trip.
- Description:
    - Generates a custom PDF checklist with sections for the trip details, essential items, recommendations, optional items, and accessories.
    - The document is saved to the provided file path (fileName).
- Returns: None.

2. addSectionHeader(PDPageContentStream contentStream, String header, float margin, float yPosition, PDType1Font font, int fontSize):
    - Parameters:
        - contentStream: The stream used to write content to the PDF page.
        - header: The title of the section (e.g., "Trip:", "Essential:", etc.).
        - margin: The left margin for positioning the header.
        - yPosition: The current y-coordinate for positioning.
        - font: The font to use for the header.
        - fontSize: The font size to use for the header.
    - Description:
        - Adds a section header to the PDF.
        - Adjusts the y-position for the next section.
    - Returns: The updated y-position after adding the header.
3. addBulletList(PDPageContentStream contentStream, String[] items, float xPosition, float yPosition, PDType1Font font, int fontSize, float leading):
    - Parameters:
        - contentStream: The stream used to write content to the PDF page.
        - items: An array of items to be listed as bullets.
        - xPosition: The x-coordinate for positioning the list.
        - yPosition: The current y-coordinate for positioning the list.
        - font: The font to use for the list items.
        - fontSize: The font size to use for the list items.
        - leading: The line height or spacing between list items.
    - Description:
        - Adds a bullet-point list to the PDF.
        - Adjusts the y-position after adding each item.
    - Returns: The updated y-position after adding the list.
4. addParagraph(PDPageContentStream contentStream, String text, float xPosition, float yPosition, PDType1Font font, int fontSize, float maxWidth, float leading):
    - Parameters:
        - contentStream: The stream used to write content to the PDF page.
        - text: The text to be added as a paragraph.

        - xPosition: The x-coordinate for positioning the paragraph.
        - yPosition: The current y-coordinate for positioning the paragraph.
        - font: The font to use for the text.
        - fontSize: The font size to use for the text.
        - maxWidth: The maximum width for the paragraph (used for text wrapping).
        - leading: The line height or spacing between lines of text.
    - Description:
        - Adds a paragraph to the PDF with automatic word wrapping.
        - Adjusts the y-position after adding the text.
    - Returns: The updated y-position after adding the paragraph.

Example Usage:

```
String fileName = "TripChecklist.pdf";
ConversationState state = new ConversationState();
state.setUserName("Thales Campos");

String tripDetails = "Trip to Brazil, Italy, and Spain.";
String[] essentialItems = {"T-shirts", "Shorts", "Sunglasses"};
String[] recommendations = {"Comfortable Shoes", "Camera"};
String[] optionalItems = {"Swimwear", "Hat"};
String[] accessories = {"Backpack", "Travel Pillow"};

PDFGenerator.generateChecklist(fileName, state, tripDetails, essentialItems, recommendations,
optionalItems, accessories);
```

This will generate a PDF titled "TripChecklist.pdf" containing a personalized checklist for Thales Campos, detailing trip information and packing essentials.

Security Considerations:

- Ensure that any user-generated input (e.g., tripDetails, essentialItems) is sanitized to prevent injection attacks.
- The file generation process should be done in a secure location to avoid unauthorized access.

TerminalChatbot Class Documentation

Overview:

The TerminalChatbot class serves as the entry point for the chatbot application. It initializes the ConversationController and starts the interaction by invoking the run() method. This setup is typically used for a terminal-based interface where the chatbot engages with the user in a command-line environment.

Dependencies:

- ConversationController: The class responsible for controlling the flow of the conversation, processing user inputs, and managing the chatbot's state.

Methods:

1. main(String[] args):
    - Parameters:
        - args: Command-line arguments (if any).
    - Description:
        - This is the entry point of the application. It creates an instance of the ConversationController and starts the conversation by calling its run() method.
    - Returns: None.

Example Usage:

To run the chatbot, the user simply needs to execute the TerminalChatbot class in a terminal environment.

```
java main.java.com.tripper.TerminalChatbot
```

This command will start the chatbot, which will interact with the user by controlling the conversation flow through the ConversationController.

Flow:

1. Initialization:
    - The main() method initializes a ConversationController object.
2. Conversation Start:
    - The run() method of the ConversationController is invoked, starting the chatbot's interaction with the user.

This class does not handle direct user interaction but relies on the ConversationController to manage the logic and state of the conversation.

TripChecklist Class Documentation

Overview:

The TripChecklist class holds and organizes the various categories of items for a travel checklist. It categorizes the items into essential items, recommendations, optional items, and accessories, which are useful for generating a detailed packing list for the user.

Fields:

- essentialItems: An array of essential items needed for the trip (e.g., passport, tickets).
- recommendations: An array of recommended items (e.g., sunscreen, camera).
- optionalItems: An array of optional items that might be useful but are not strictly necessary (e.g., a book, extra shoes).
- accessories: An array of accessory items (e.g., hats, sunglasses, scarves).

Constructor:

- TripChecklist(String[] essentialItems, String[] recommendations, String[] optionalItems, String[] accessories)
    - Parameters:
        - essentialItems: An array containing items deemed essential for the trip.
        - recommendations: An array of recommended items for the trip.
        - optionalItems: An array of optional items that can be included for the trip.
        - accessories: An array of accessory items for the trip.
    - Description: This constructor initializes the four categories of items, providing a structured way to manage the checklist for the trip.

Getter Methods:

- getEssentialItems(): Returns the array of essential items.
- getRecommendations(): Returns the array of recommended items.
- getOptionalItems(): Returns the array of optional items.
- getAccessories(): Returns the array of accessory items.

Example Usage:

```
// Example of how to create a TripChecklist and access the items
String[] essentials = {"Passport", "Flight tickets", "Travel Insurance"};
String[] recommendations = {"Camera", "Sunscreen", "Guidebook"};
String[] optional = {"Book", "Extra shoes"};
String[] accessories = {"Hat", "Sunglasses", "Scarf"};

TripChecklist checklist = new TripChecklist(essentials, recommendations, optional, accessories);

// Accessing the items
String[] essentialItems = checklist.getEssentialItems();
String[] recommendations = checklist.getRecommendations();
String[] optionalItems = checklist.getOptionalItems();
String[] accessories = checklist.getAccessories();
```

Use Case:

This class is typically used to store and retrieve different categories of items that need to be packed for a trip. It can be passed to other components like PDFGenerator to create packing lists, or to be used within the ConversationController to provide personalized recommendations.

TripChecklistGenerator Class Documentation

Overview:

The TripChecklistGenerator class is responsible for generating a set of checklist items (essential items, recommendations, optional items, and accessories) based on the details of the trip, particularly the travel month. It uses simple heuristics to categorize items based on whether the trip is likely to be in summer, winter, or a neutral season.

Method:

- generateChecklist(TripDetails details)
    - Parameters:
        - details: An instance of TripDetails that contains information about the user's travel month.
    - Returns: A TripChecklist object containing categorized items (essential, recommendations, optional, and accessories).
    - Description: This method generates checklist arrays based on the travel month in the TripDetails object. It uses the travel month to determine which items should be included for the trip, following a set of heuristics:
        - Summer months (June, July, August) or mentions of "summer" lead to lighter clothing items.
        - Winter months (December, January, February) or mentions of "winter" lead to warmer clothing and accessories.
        - If the travel month is unclear, a neutral set of items is chosen.

Logic:

- Summer Travel (e.g., June, July, August):
    - Essential Items: Light T-shirt, Shorts, Comfortable walking shoes, Sunglasses.
    - Recommendations: Hat, Sunscreen, Umbrella (for potential rain).
    - Optional Items: Light Sweater, Extra pair of Socks.
    - Accessories: Crossbody Bag, Travel Adapter, Power Bank.
- Winter Travel (e.g., December, January, February):
    - Essential Items: Warm Jacket, Thermal Wear, Gloves, Scarf, Beanie.
    - Recommendations: Boots, Extra Socks.
    - Optional Items: Lip Balm, Hand Warmers.
    - Accessories: Backpack, Travel Adapter.
- Neutral Travel (if no specific month or season is clear):
    - Essential Items: Versatile T-shirt, Jeans, Comfortable Shoes.
    - Recommendations: Light Jacket, Umbrella.
    - Optional Items: Hat, Sunglasses.
    - Accessories: Backpack, Portable Charger.

Example Usage:

```
java

// Example of how to generate a checklist based on travel month

TripDetails tripDetails = new TripDetails();

tripDetails.setTravelMonth("June");
```

```
TripChecklist checklist = TripChecklistGenerator.generateChecklist(tripDetails);

// Accessing checklist items
String[] essentialItems = checklist.getEssentialItems();
String[] recommendations = checklist.getRecommendations();
String[] optionalItems = checklist.getOptionalItems();
String[] accessories = checklist.getAccessories();
```

Use Case:

This class is typically used in the context of a travel planning application or a chatbot. After determining the user's travel month, the TripChecklistGenerator can create a personalized packing list to help the user prepare for their trip. The generated checklist can be used in various formats (e.g., displayed on the user interface, saved as a PDF, etc.).

TripDetails Class Documentation

Overview:

The TripDetails class holds information about the user's trip. It includes a list of locations (places the user intends to visit) and the travel month (the month during which the user is planning their trip).

Fields:

1. locations (List<String>):
   - A list of locations that the user intends to visit. This could represent cities, countries, or specific landmarks.
2. travelMonth (String):
   - The month when the user is planning to travel, represented as a string (e.g., "June", "December"). This helps in generating a personalized packing list based on seasonal factors.

Methods:

- getLocations():
  - Description: Returns the list of locations for the trip.
  - Return Type: List<String>
- setLocations(List<String> locations):
  - Description: Sets the list of locations for the trip.
  - Parameters: A List<String> containing location names (e.g., "Paris", "Venice").
- getTravelMonth():
  - Description: Returns the travel month for the trip.
  - Return Type: String
- setTravelMonth(String travelMonth):
  - Description: Sets the travel month for the trip.
  - Parameters: A String representing the month of travel (e.g., "June").
- toString():

- Description: Returns a string representation of the TripDetails object, which includes the list of locations and the travel month.
- Return Type: String

Example Usage:

```
java
// Creating a TripDetails instance
TripDetails tripDetails = new TripDetails();

// Setting the travel month and locations
tripDetails.setTravelMonth("June");
tripDetails.setLocations(List.of("Brazil", "Italy", "Spain"));

// Accessing the trip details
String travelMonth = tripDetails.getTravelMonth(); // "June"
List<String> locations = tripDetails.getLocations(); // ["Brazil", "Italy", "Spain"]

// Printing the trip details
System.out.println(tripDetails.toString());
// Output: TripDetails [locations=[Brazil, Italy, Spain], travelMonth=June]
```

Use Case:

The TripDetails class is used to store essential trip information, such as the destinations the user is visiting and when they are traveling. This information is used by other classes, like TripChecklistGenerator, to create personalized checklists and planning resources for the user.

WeatherService Class Documentation

Overview:

The WeatherService class is responsible for fetching weather forecast data from the OpenWeatherMap API. The class uses HTTP requests to fetch weather data for a specific location and returns the data as a WeatherResponse object.

Fields:

- API_KEY (String):
  - A constant holding the API key required to authenticate requests to the OpenWeatherMap API. *(Note: For security reasons, make sure to keep the API key secret.)*
- BASE_URL (String):
  - A constant holding the base URL for the OpenWeatherMap API endpoint.

Method:

- getForecastData(String location):
  - Description: This method fetches the weather forecast for a specified location using the OpenWeatherMap API.
  - Parameters:
    - location (String): The name of the location for which the forecast is needed (e.g., "Paris").
  - Return Type: WeatherResponse
    - Returns a WeatherResponse object containing the parsed forecast data.
  - Throws:
    - Catches and prints exceptions related to network requests or data parsing.

Steps:

5. The method builds the URL to request the forecast data using the location and the API_KEY.
6. Sends an HTTP GET request to OpenWeatherMap API.
7. If the request is successful (status code 200), it reads the response as a string.
8. The response string is parsed into a WeatherResponse object using the Gson library.
9. The method returns the WeatherResponse object containing the forecast data.
10. In case of any errors (e.g., failed request or invalid data), it prints an error message and returns null.

Example Usage:

```
java
// Create a WeatherService instance
WeatherService weatherService = new WeatherService();

// Fetch weather forecast for a given location
WeatherResponse forecast = weatherService.getForecastData("Paris");

if (forecast != null) {
    // Process the forecast data
    System.out.println("Weather forecast: " + forecast);
} else {
    System.out.println("Unable to retrieve weather data.");
}
```

Notes:

1. The API_KEY is stored as a private constant within the class. Be sure to keep it secret, and avoid sharing it in public repositories.
2. The WeatherResponse class is assumed to be a data model for parsing the JSON response from the OpenWeatherMap API. This class would typically contain fields for temperature, weather conditions, etc.
3. Make sure to handle any potential issues with rate limits or incorrect API keys when using this service.

3. UI Implementation

4. Rest API

4.1 Rest API Implementation

5. Weather API

6. External Packages

Gson

The Gson library is a tool from Google used to convert Java objects to JSON representation and vice versa. The goals of this library are to allow customized representation, support complex objects, and generate compact output. According to the official documentation, its performance and scalability provide results more than sufficient for the expectations of this project. The components used are: .JsonArray; .JsonObject; .JsonParser. Their purpose is to interact with objects and arrays and parse strings.

OpenNLP

The OpenNLP library is a tool from Apache used to process natural language text, based on machine learning. The components used are: .DictionaryLemmatizer; .POSModel; .POSTaggerME; .SentenceDetectorME; .SentenceModel; .TokenizerME; .TokenizerModel. Their purpose is to identify sentences, split text, determine grammatical tags for tokens, and convert words.

PDFBox

The PDFBox library is an open source tool from Apache used to create, manipulate, and extract information from PDF documents. The components used are: .PDDocument; .PDPage; .PDPageContentStream; .PDRectangle; .font.PDType1Font; .font.Standard14Fonts. Their purpose is to manage, structure, write, create PDFs, and deal with fonts.

7. Project Setup

8. Milestone 1

8.1 Goals

For the first milestone, the goals involved researching different models and exploring what was available related to the desired target. The next step was to discuss the audience, determine which tools to implement, and identify the possible devices and platforms to work with. After that, we created a persona and developed a visual identity, including a logo and some possible wireframes. The final step was coding a prototype using APIs and packages that could make the project feasible while documenting, researching, and saving all necessary information throughout the process.

8.2 Report

The group decided to start the project by conducting simple research using all the material provided by the lecturers on Moodle and organizing the schedule, steps, and roles. The focus was on developing and delivering a complete and functional project based on each member's previous knowledge, personal experience, and abilities.

Trello was chosen as the platform to organize and share the project's steps, allowing changes, suggestions, and task storage. Since the group had used this tool before, the main goals were to enhance productivity and teamwork.

A Gantt chart was created using a free template available on Google Sheets. As the project progresses, the group can visualize and assess whether time management requirements are being met. This provides a sense of control and helps in making decisions, adjustments, and improvements as needed.

GitHub was selected as the repository platform, enabling authorized members to create, store, manage, share, and comment on all necessary files, code, and documents. As a well-known and secure platform, GitHub facilitates fast, collaborative, and well-documented work, ensuring version control and a history of completed tasks.

OpenAi’s GPT-3.5 Turbo was selected as the REST API to provide a code with an elevated level of communication, This model with 16,385 context window, uses NLP (natural language processing) to interact, generate and summarize text, answer questions, and much more. All these processes are supported in the code using JSON.

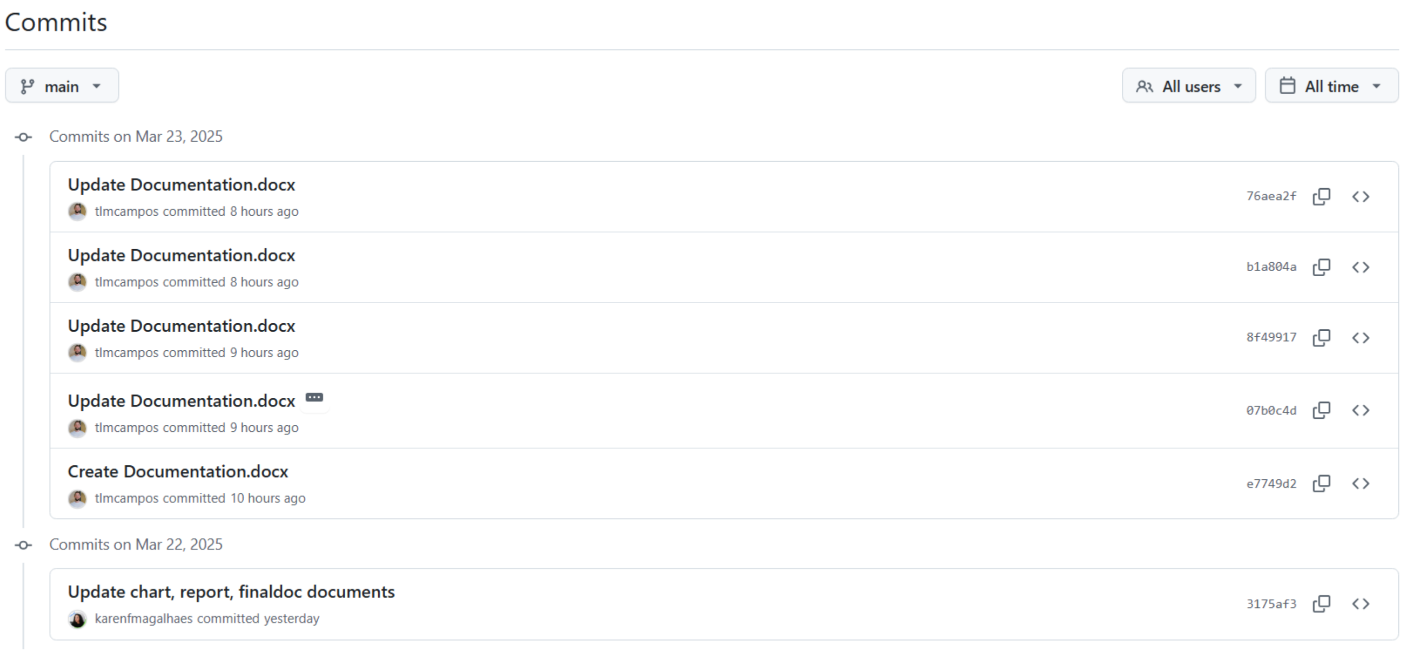
Roles:

Karen – Individual research and discussion, second research, flow and chart, coding tests, define persona, report, fill project final doc.

Thales – Individual research and discussion, trello, wireframes, logo, coding tests, define audience, documentation.

Vitor – Individual research and discussion, requirements, GitHub repository, coding prototype, coding tests, map edge cases, entity mapping.

8.3 Commit Logs


Commits

main

All users

All time

Commits on Mar 23, 2025

Update Documentation.docx
tlmcampos committed 8 hours ago
76aea2f

Update Documentation.docx
tlmcampos committed 8 hours ago
b1a804a

Update Documentation.docx
tlmcampos committed 9 hours ago
8f49917

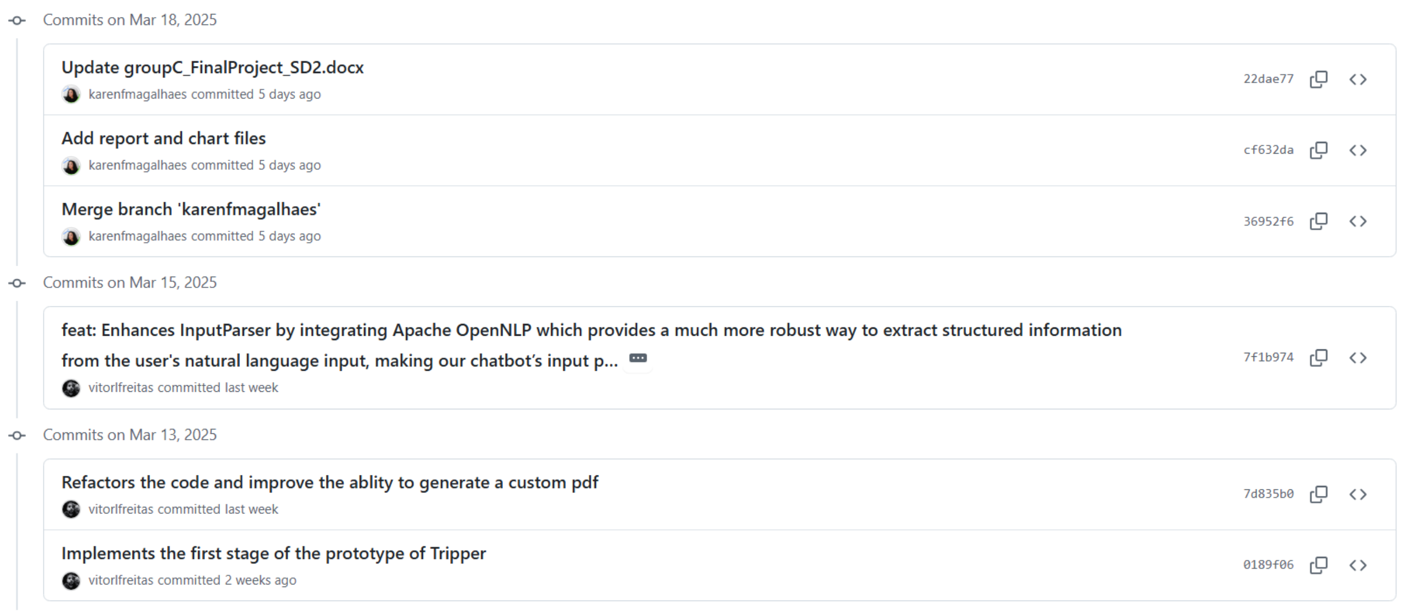
Update Documentation.docx
tlmcampos committed 9 hours ago
07b0c4d

Create Documentation.docx
tlmcampos committed 10 hours ago
e7749d2

Commits on Mar 22, 2025

Update chart, report, finaldoc documents
karenfmagalhaes committed yesterday
3175af3

Figure 5 - GitHub commit history showcasing the project's development process, including code updates, fixes, and feature implementations by the team.


Commits on Mar 18, 2025

Update groupC_FinalProject_SD2.docx
karenfmagalhaes committed 5 days ago — 22dae77

Add report and chart files
karenfmagalhaes committed 5 days ago — cf632da

Merge branch 'karenfmagalhaes'
karenfmagalhaes committed 5 days ago — 36952f6

Commits on Mar 15, 2025

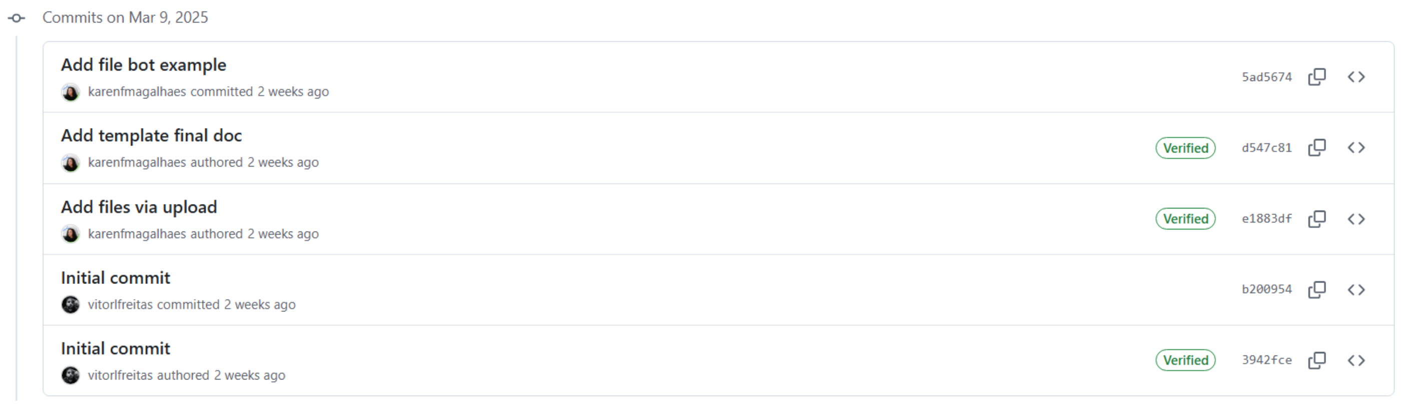
feat: Enhances InputParser by integrating Apache OpenNLP which provides a much more robust way to extract structured information from the user's natural language input, making our chatbot's input p...
vitorlfreitas committed last week — 7f1b974

Commits on Mar 13, 2025

Refactors the code and improve the ablity to generate a custom pdf
vitorlfreitas committed last week — 7d835b0

Implements the first stage of the prototype of Tripper
vitorlfreitas committed 2 weeks ago — 0189f06

Figure 6 - GitHub commit history showcasing the project's development process, including code updates, fixes, and feature implementations by the team.


Commits on Mar 9, 2025

Add file bot example
karenfmagalhaes committed 2 weeks ago — 5ad5674

Add template final doc
karenfmagalhaes authored 2 weeks ago — Verified — d547c81

Add files via upload
karenfmagalhaes authored 2 weeks ago — Verified — e1883df

Initial commit
vitorlfreitas committed 2 weeks ago — b200954

Initial commit
vitorlfreitas authored 2 weeks ago — Verified — 3942fce

Figure 7 - GitHub commit history showcasing the project's development process, including code updates, fixes, and feature implementations by the team.

8.4 Full Log Details

```
stupc@LAPTOP-715EEOOP MINGW64 ~/Documents/GitHub/trip-clothing-planner (main)
$ git log
commit 76aea2f0e15b962e51bd654158ca43844fc4a72b (HEAD -> main, origin/main, orig
in/HEAD)
Author: Thales Campos <159160714+tlmcampos@users.noreply.github.com>
Date:   Sun Mar 23 12:56:01 2025 +0000

    Update Documentation.docx

commit b1a804a54b7bcff1aa7d1df5fc5ffbaf209c2509
Author: Thales Campos <159160714+tlmcampos@users.noreply.github.com>
Date:   Sun Mar 23 12:46:13 2025 +0000

    Update Documentation.docx

commit 8f4991753ae28eb736cbe957d058f458e419ff2a
Author: Thales Campos <159160714+tlmcampos@users.noreply.github.com>
Date:   Sun Mar 23 12:11:41 2025 +0000

    Update Documentation.docx

commit 07b0c4d6829e34d8ff7dd3defa0958e2c0752475
Author: Thales Campos <159160714+tlmcampos@users.noreply.github.com>
```

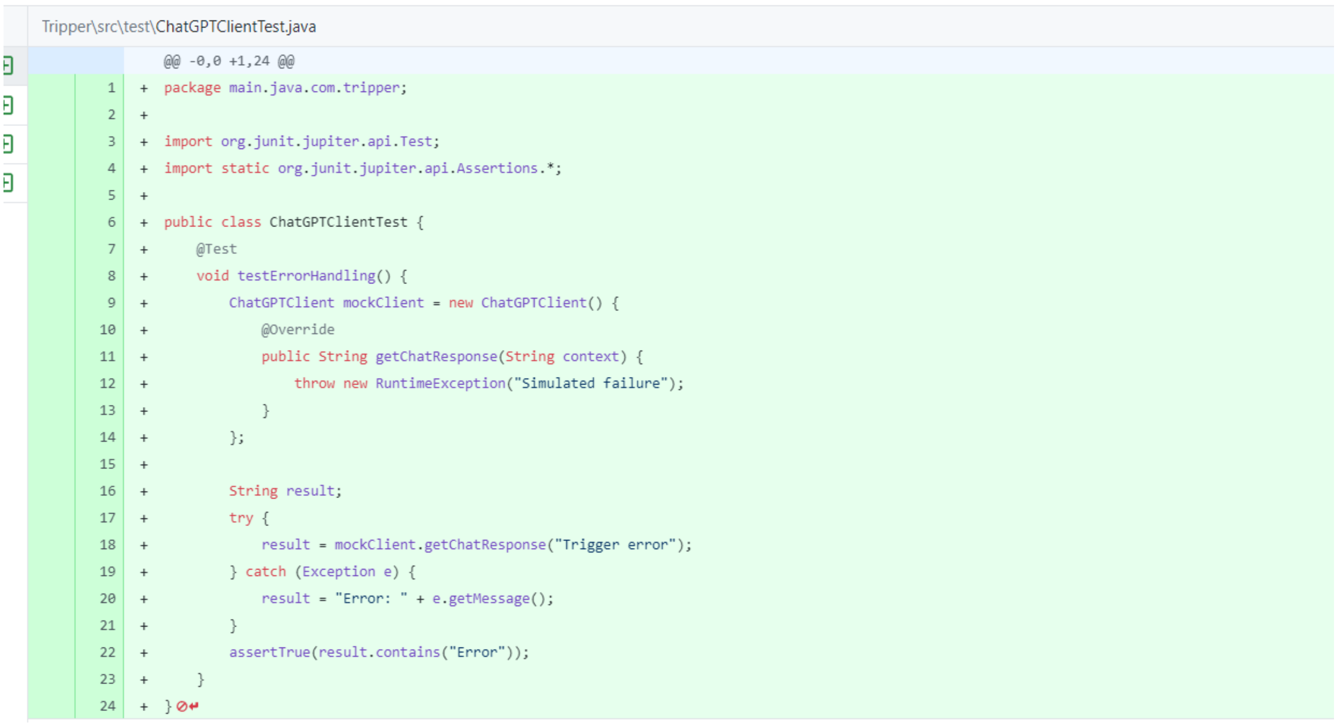
```
Date:   Sun Mar 23 11:44:55 2025 +0000

    Update Documentation.docx

    describe classes ChatGPTClient, ComversationController, and ConversationMana
ger

commit e7749d258d8db49ba650096f082f95771c78acfe
Author: Thales Campos <159160714+tlmcampos@users.noreply.github.com>
Date:   Sun Mar 23 11:18:11 2025 +0000

    Create Documentation.docx

commit 3175af3a470fdeda3142cab1e4d04258c4c4dcdc
Author: Karen F Magalhaes <karenfmagalhaes@gmail.com>
Date:   Sat Mar 22 12:33:59 2025 +0000

    Update chart, report, finaldoc documents

commit 22dae7766e11b23b9d009c4195f7bb4652f2933d
Author: Karen F Magalhaes <karenfmagalhaes@gmail.com>
Date:   Tue Mar 18 16:58:45 2025 +0000

    Update groupC_FinalProject_SD2.docx

commit cf632da0df3be834da04a5c7f07413923196bf2c
Author: Karen F Magalhaes <karenfmagalhaes@gmail.com>
Date:   Tue Mar 18 14:30:41 2025 +0000

    Add report and chart files

commit 36952f6fbfdfdf397891e944ae66b22f43aa27ec
Merge: 7f1b974 5ad5674
Author: Karen F Magalhaes <karenfmagalhaes@gmail.com>
Date:   Tue Mar 18 13:25:10 2025 +0000

    Merge branch 'karenfmagalhaes'

commit 7f1b9747757958ac2c9f823ed7264cceeb11786b
Author: vitorlfreitas <vitor.lucfreitas@gmail.com>
Date:   Sat Mar 15 00:22:42 2025 +0000

    feat: Enhances InputParser by integrating Apache OpenNLP which provides a mu
ch more robust way to extract structured information from the user's natural lan
guage input, making our chatbot’s input parsing far stronger and  Adds SLF4J dep
endencies to fix OpenNLP logging issue and enable UD EWT models

commit d547c815c70c2939d6732bccad90325f70706b84
Author: Karen F Magalhaes <59734660+karenfmagalhaes@users.noreply.github.com>
Date:   Sun Mar 9 22:33:29 2025 +0000

    Add template final doc

commit e1883df1c1a891202486a2111e166489ed7d22bf
Author: Karen F Magalhaes <59734660+karenfmagalhaes@users.noreply.github.com>
Date:   Sun Mar 9 21:53:09 2025 +0000

    Add files via upload

commit b200954425837219dc5e565fd2bfa800f76bdc9d (origin/thales)
Author: vitorlfreitas <vitor.lucfreitas@gmail.com>
Date:   Sun Mar 9 21:29:32 2025 +0000

    Initial commit

commit 3942fcea01f4010467702137c9ea70d74e4476be
Author: Vitor Freitas <130233950+vitorlfreitas@users.noreply.github.com>
Date:   Sun Mar 9 21:23:20 2025 +0000

    Initial commit
```

```
(END)
```

9. Milestone 2

9.1 Goals

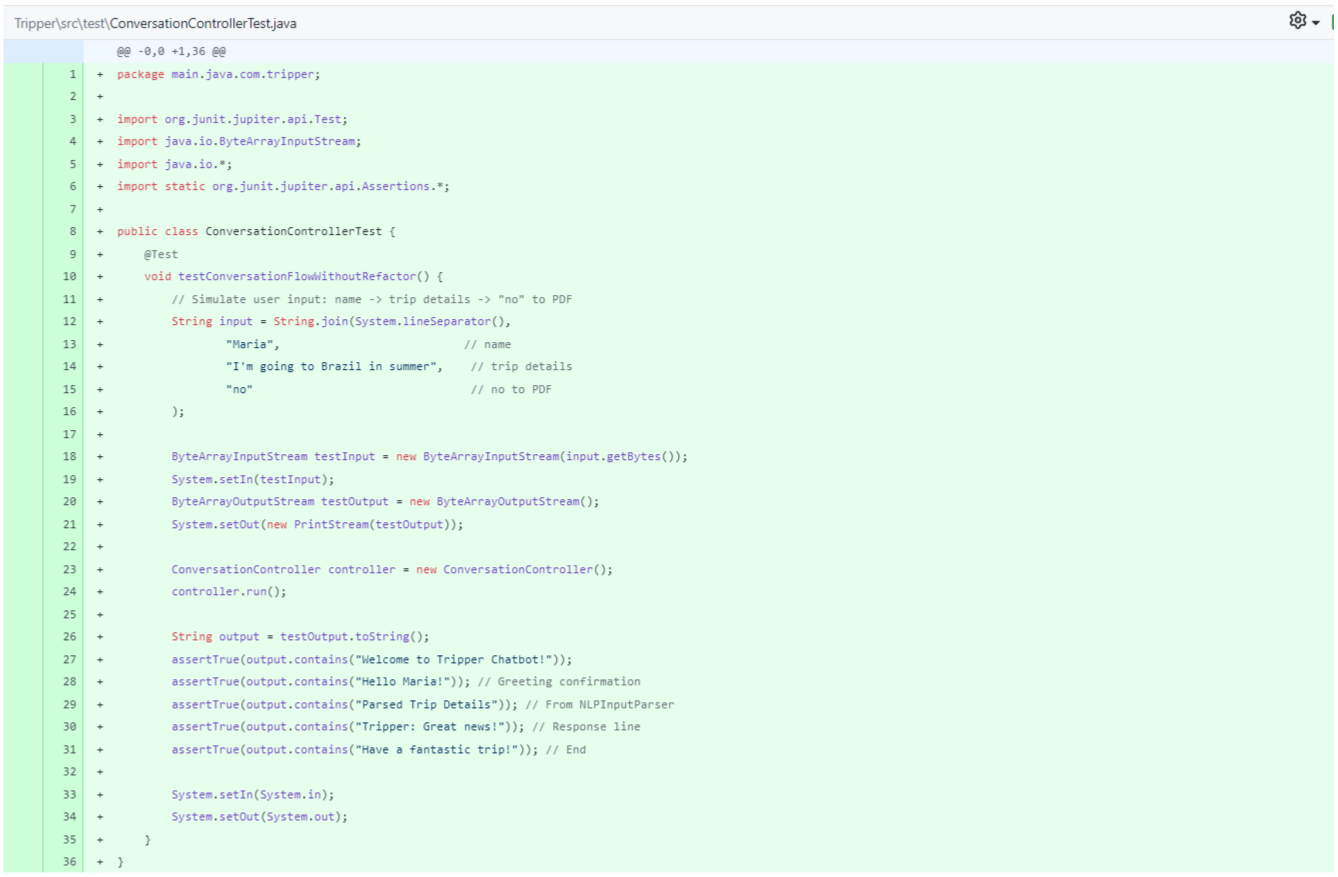
9.2 Junit Tests Integration



Tripper\src\test\ChatGPTClientTest.java

```
@@ -0,0 +1,24 @@
+ package main.java.com.tripper;
+
+ import org.junit.jupiter.api.Test;
+ import static org.junit.jupiter.api.Assertions.*;
+
+ public class ChatGPTClientTest {
+     @Test
+     void testErrorHandling() {
+         ChatGPTClient mockClient = new ChatGPTClient() {
+             @Override
+             public String getChatResponse(String context) {
+                 throw new RuntimeException("Simulated failure");
+             }
+         };
+
+         String result;
+         try {
+             result = mockClient.getChatResponse("Trigger error");
+         } catch (Exception e) {
+             result = "Error: " + e.getMessage();
+         }
+         assertTrue(result.contains("Error"));
+     }
+ }
```

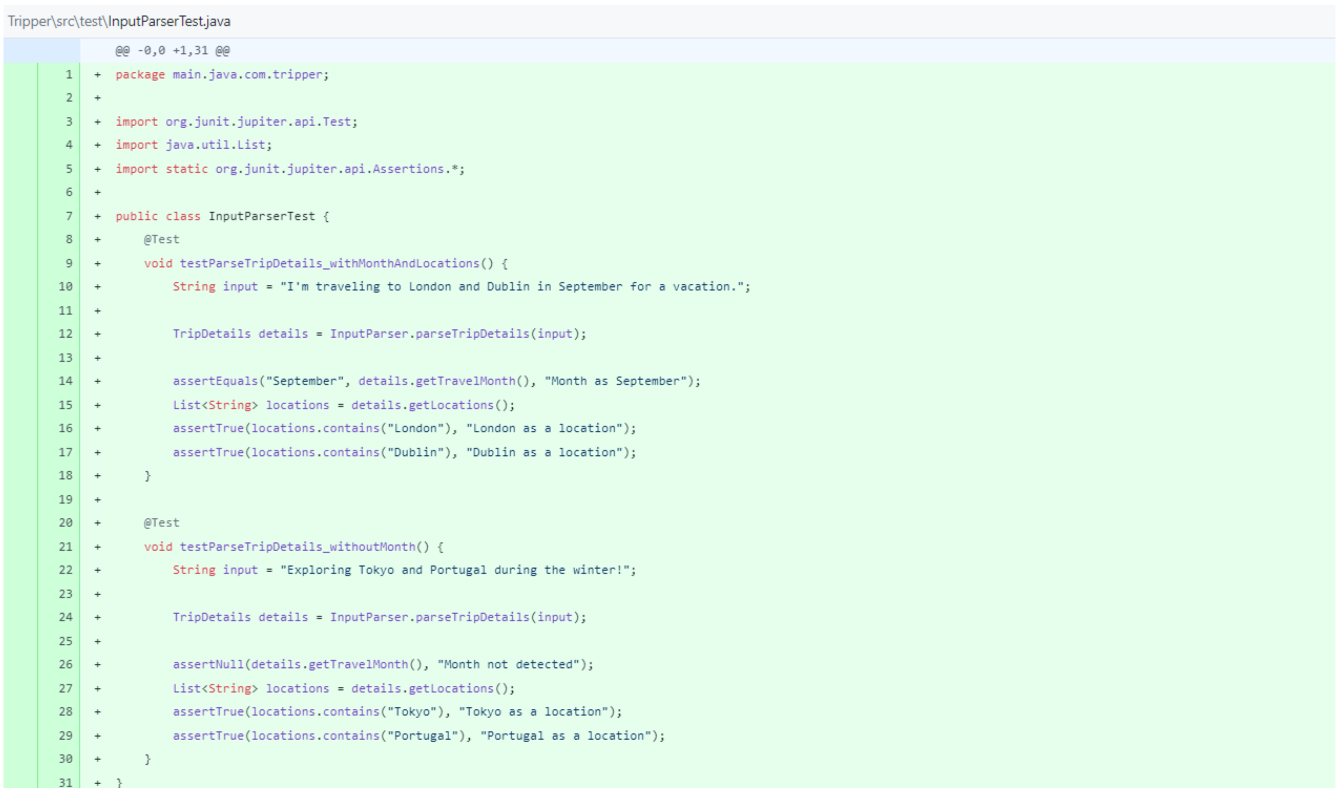
Figure 8 - Unit test for ChatGPTClient class



Tripper\src\test\ConversationControllerTest.java

```
@@ -0,0 +1,36 @@
+ package main.java.com.tripper;
+
+ import org.junit.jupiter.api.Test;
+ import java.io.ByteArrayInputStream;
+ import java.io.*;
+ import static org.junit.jupiter.api.Assertions.*;
+
+ public class ConversationControllerTest {
+     @Test
+     void testConversationFlowWithoutRefactor() {
+         // Simulate user input: name -> trip details -> "no" to PDF
+         String input = String.join(System.lineSeparator(),
+                 "Maria",                           // name
+                 "I'm going to Brazil in summer",    // trip details
+                 "no"                                // no to PDF
+         );
+
+         ByteArrayInputStream testInput = new ByteArrayInputStream(input.getBytes());
+         System.setIn(testInput);
+         ByteArrayOutputStream testOutput = new ByteArrayOutputStream();
+         System.setOut(new PrintStream(testOutput));
+
+         ConversationController controller = new ConversationController();
+         controller.run();
+
+         String output = testOutput.toString();
+         assertTrue(output.contains("Welcome to Tripper Chatbot!"));
+         assertTrue(output.contains("Hello Maria!")); // Greeting confirmation
+         assertTrue(output.contains("Parsed Trip Details")); // From NLPInputParser
+         assertTrue(output.contains("Tripper: Great news!")); // Response line
+         assertTrue(output.contains("Have a fantastic trip!")); // End
+
+         System.setIn(System.in);
+         System.setOut(System.out);
+     }
+ }
```

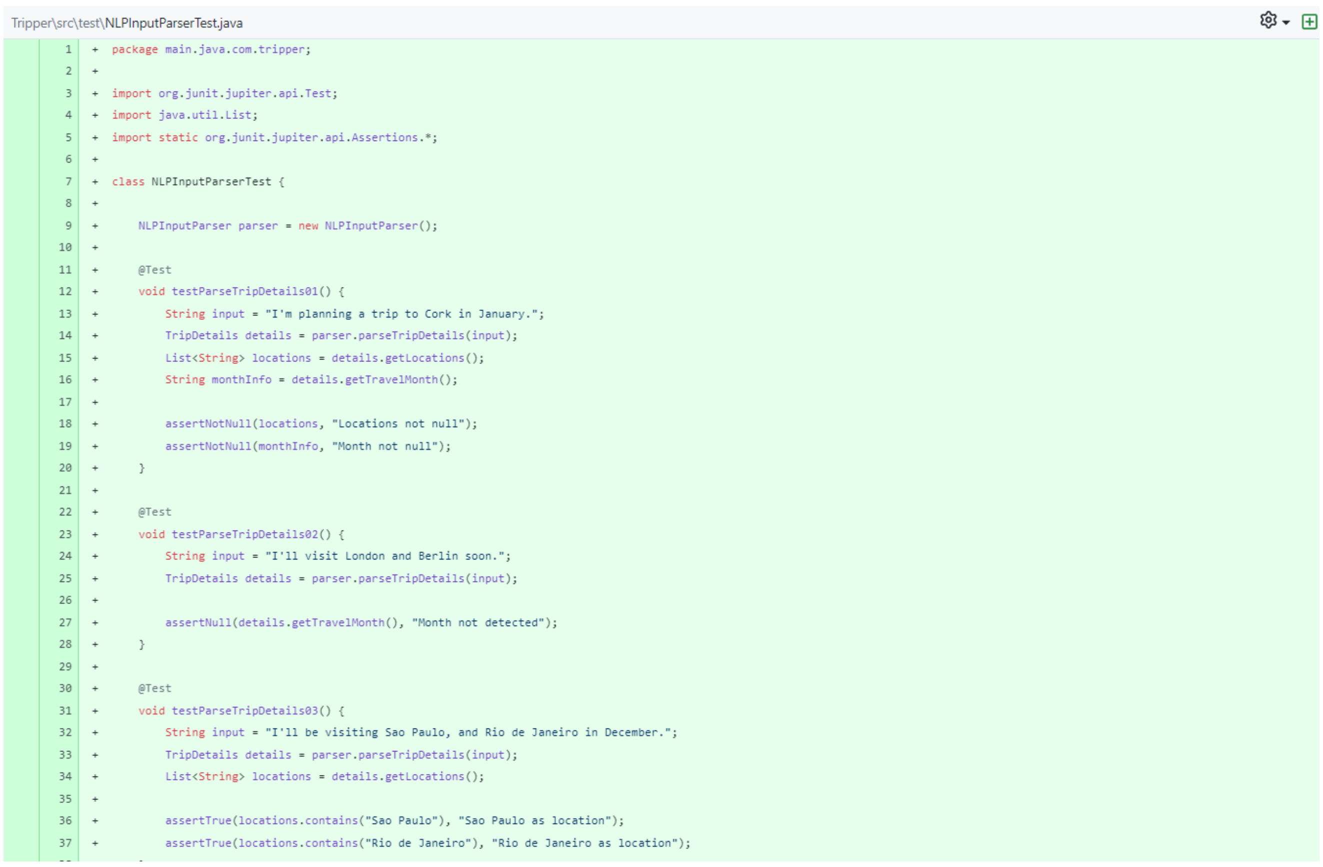
Figure 9 - Unit test for ConversationController class



Tripper\src\test\InputParserTest.java

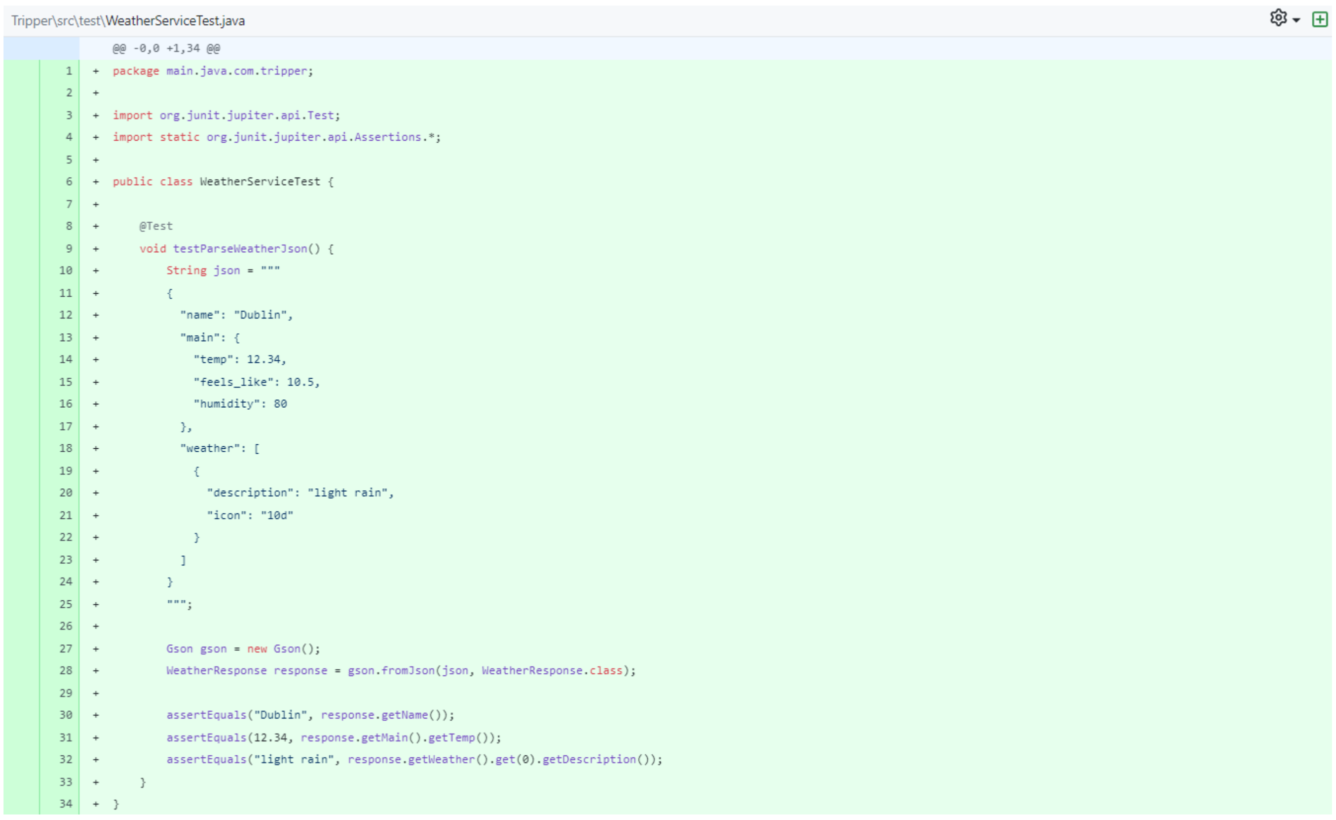
```
@@ -0,0 +1,31 @@
+ package main.java.com.tripper;
+
+ import org.junit.jupiter.api.Test;
+ import java.util.List;
+ import static org.junit.jupiter.api.Assertions.*;
+
+ public class InputParserTest {
+     @Test
+     void testParseTripDetails_withMonthAndLocations() {
+         String input = "I'm traveling to London and Dublin in September for a vacation.";
+
+         TripDetails details = InputParser.parseTripDetails(input);
+
+         assertEquals("September", details.getTravelMonth(), "Month as September");
+         List<String> locations = details.getLocations();
+         assertTrue(locations.contains("London"), "London as a location");
+         assertTrue(locations.contains("Dublin"), "Dublin as a location");
+     }
+
+     @Test
+     void testParseTripDetails_withoutMonth() {
+         String input = "Exploring Tokyo and Portugal during the winter!";
+
+         TripDetails details = InputParser.parseTripDetails(input);
+
+         assertNull(details.getTravelMonth(), "Month not detected");
+         List<String> locations = details.getLocations();
+         assertTrue(locations.contains("Tokyo"), "Tokyo as a location");
+         assertTrue(locations.contains("Portugal"), "Portugal as a location");
+     }
+ }
```

Figure 10 - Unit test for InputParser class


Tripper\src\test\NLPInputParserTest.java

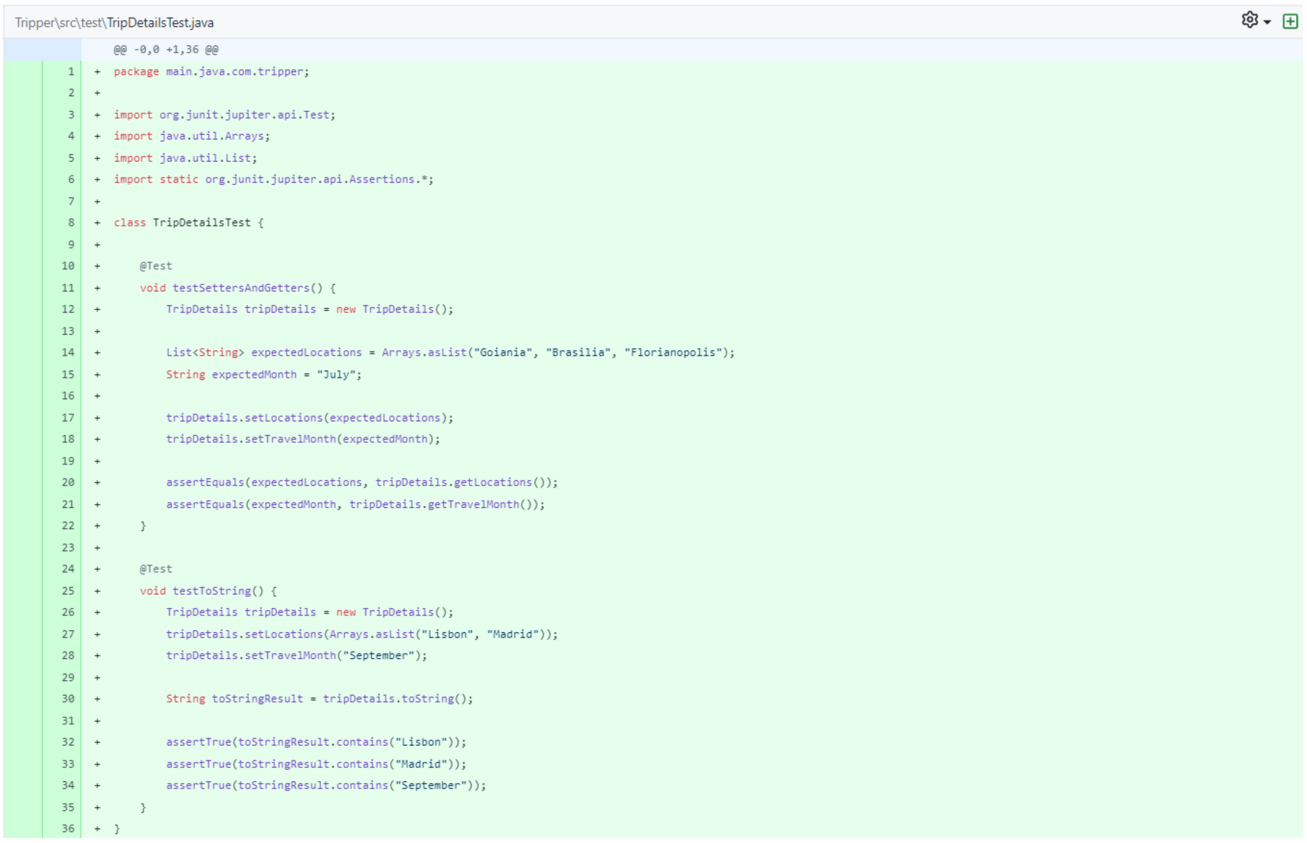
```
+ package main.java.com.tripper;
+
+ import org.junit.jupiter.api.Test;
+ import java.util.List;
+ import static org.junit.jupiter.api.Assertions.*;
+
+ class NLPInputParserTest {
+
+     NLPInputParser parser = new NLPInputParser();
+
+     @Test
+     void testParseTripDetails01() {
+         String input = "I'm planning a trip to Cork in January.";
+         TripDetails details = parser.parseTripDetails(input);
+         List<String> locations = details.getLocations();
+         String monthInfo = details.getTravelMonth();
+
+         assertNotNull(locations, "Locations not null");
+         assertNotNull(monthInfo, "Month not null");
+     }
+
+     @Test
+     void testParseTripDetails02() {
+         String input = "I'll visit London and Berlin soon.";
+         TripDetails details = parser.parseTripDetails(input);
+
+         assertNull(details.getTravelMonth(), "Month not detected");
+     }
+
+     @Test
+     void testParseTripDetails03() {
+         String input = "I'll be visiting Sao Paulo, and Rio de Janeiro in December.";
+         TripDetails details = parser.parseTripDetails(input);
+         List<String> locations = details.getLocations();
+
+         assertTrue(locations.contains("Sao Paulo"), "Sao Paulo as location");
+         assertTrue(locations.contains("Rio de Janeiro"), "Rio de Janeiro as location");
```

Figure 11 - Unit test for NLPInputParser class


Tripper\src\test\WeatherServiceTest.java

```
@@ -0,0 +1,34 @@
+ package main.java.com.tripper;
+
+ import org.junit.jupiter.api.Test;
+ import static org.junit.jupiter.api.Assertions.*;
+
+ public class WeatherServiceTest {
+
+     @Test
+     void testParseWeatherJson() {
+         String json = """
+         {
+           "name": "Dublin",
+           "main": {
+             "temp": 12.34,
+             "feels_like": 10.5,
+             "humidity": 80
+           },
+           "weather": [
+             {
+               "description": "light rain",
+               "icon": "10d"
+             }
+           ]
+         }
+         """;
+
+         Gson gson = new Gson();
+         WeatherResponse response = gson.fromJson(json, WeatherResponse.class);
+
+         assertEquals("Dublin", response.getName());
+         assertEquals(12.34, response.getMain().getTemp());
+         assertEquals("light rain", response.getWeather().get(0).getDescription());
+     }
+ }
```

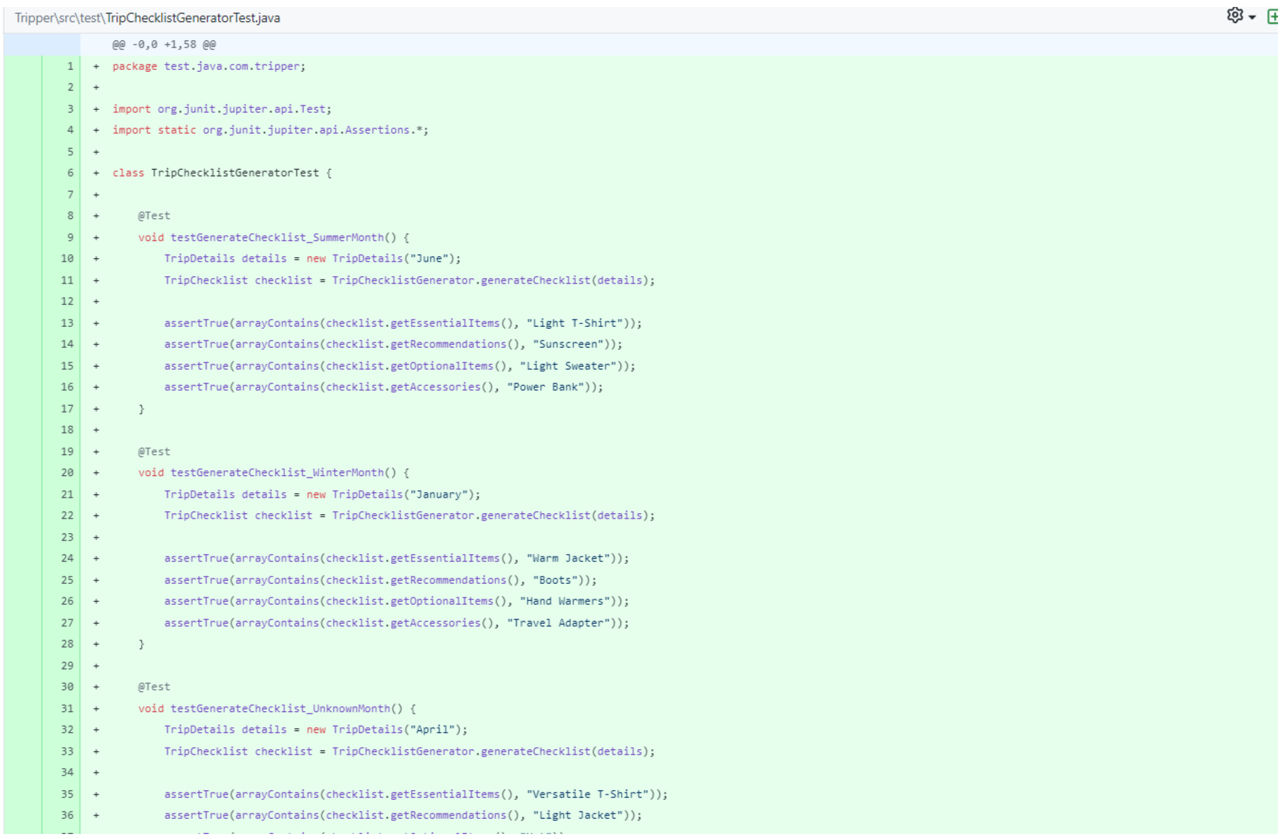
Figure 12 - Unit test for WeatherService class



Tripper\src\test\TripDetailsTest.java

```
@@ -0,0 +1,36 @@
+ package main.java.com.tripper;
+
+ import org.junit.jupiter.api.Test;
+ import java.util.Arrays;
+ import java.util.List;
+ import static org.junit.jupiter.api.Assertions.*;
+
+ class TripDetailsTest {
+
+     @Test
+     void testSettersAndGetters() {
+         TripDetails tripDetails = new TripDetails();
+
+         List<String> expectedLocations = Arrays.asList("Goiania", "Brasilia", "Florianopolis");
+         String expectedMonth = "July";
+
+         tripDetails.setLocations(expectedLocations);
+         tripDetails.setTravelMonth(expectedMonth);
+
+         assertEquals(expectedLocations, tripDetails.getLocations());
+         assertEquals(expectedMonth, tripDetails.getTravelMonth());
+     }
+
+     @Test
+     void testToString() {
+         TripDetails tripDetails = new TripDetails();
+         tripDetails.setLocations(Arrays.asList("Lisbon", "Madrid"));
+         tripDetails.setTravelMonth("September");
+
+         String toStringResult = tripDetails.toString();
+
+         assertTrue(toStringResult.contains("Lisbon"));
+         assertTrue(toStringResult.contains("Madrid"));
+         assertTrue(toStringResult.contains("September"));
+     }
+ }
```

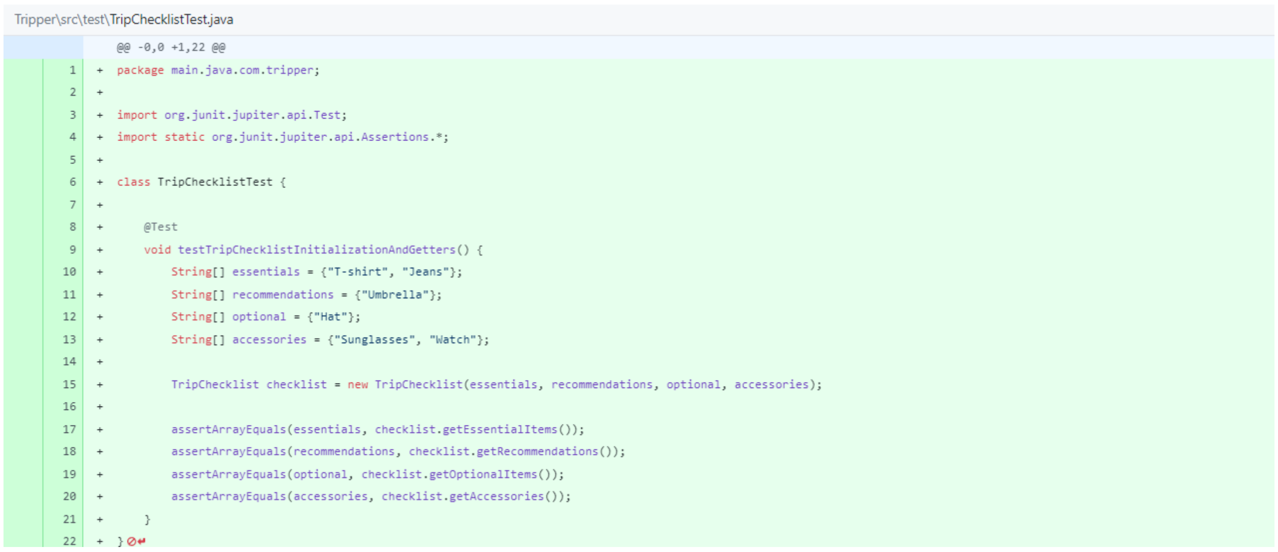
Figure 13 - Unit test for TripDetails class



Tripper\src\test\TripChecklistGeneratorTest.java

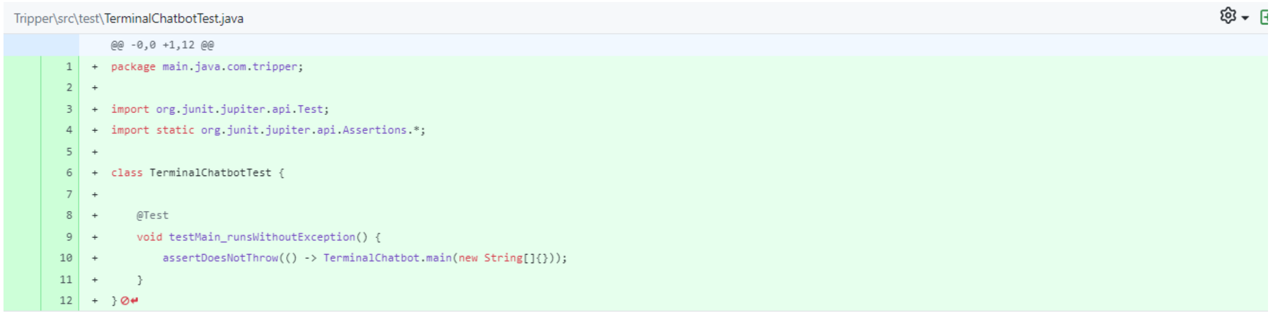
```
@@ -0,0 +1,58 @@
+ package test.java.com.tripper;
+
+ import org.junit.jupiter.api.Test;
+ import static org.junit.jupiter.api.Assertions.*;
+
+ class TripChecklistGeneratorTest {
+
+     @Test
+     void testGenerateChecklist_SummerMonth() {
+         TripDetails details = new TripDetails("June");
+         TripChecklist checklist = TripChecklistGenerator.generateChecklist(details);
+
+         assertTrue(arrayContains(checklist.getEssentialItems(), "Light T-Shirt"));
+         assertTrue(arrayContains(checklist.getRecommendations(), "Sunscreen"));
+         assertTrue(arrayContains(checklist.getOptionalItems(), "Light Sweater"));
+         assertTrue(arrayContains(checklist.getAccessories(), "Power Bank"));
+     }
+
+     @Test
+     void testGenerateChecklist_WinterMonth() {
+         TripDetails details = new TripDetails("January");
+         TripChecklist checklist = TripChecklistGenerator.generateChecklist(details);
+
+         assertTrue(arrayContains(checklist.getEssentialItems(), "Warm Jacket"));
+         assertTrue(arrayContains(checklist.getRecommendations(), "Boots"));
+         assertTrue(arrayContains(checklist.getOptionalItems(), "Hand Warmers"));
+         assertTrue(arrayContains(checklist.getAccessories(), "Travel Adapter"));
+     }
+
+     @Test
+     void testGenerateChecklist_UnknownMonth() {
+         TripDetails details = new TripDetails("April");
+         TripChecklist checklist = TripChecklistGenerator.generateChecklist(details);
+
+         assertTrue(arrayContains(checklist.getEssentialItems(), "Versatile T-Shirt"));
+         assertTrue(arrayContains(checklist.getRecommendations(), "Light Jacket"));
```

Figure 14 - Unit test for TripChecklistGenerator class



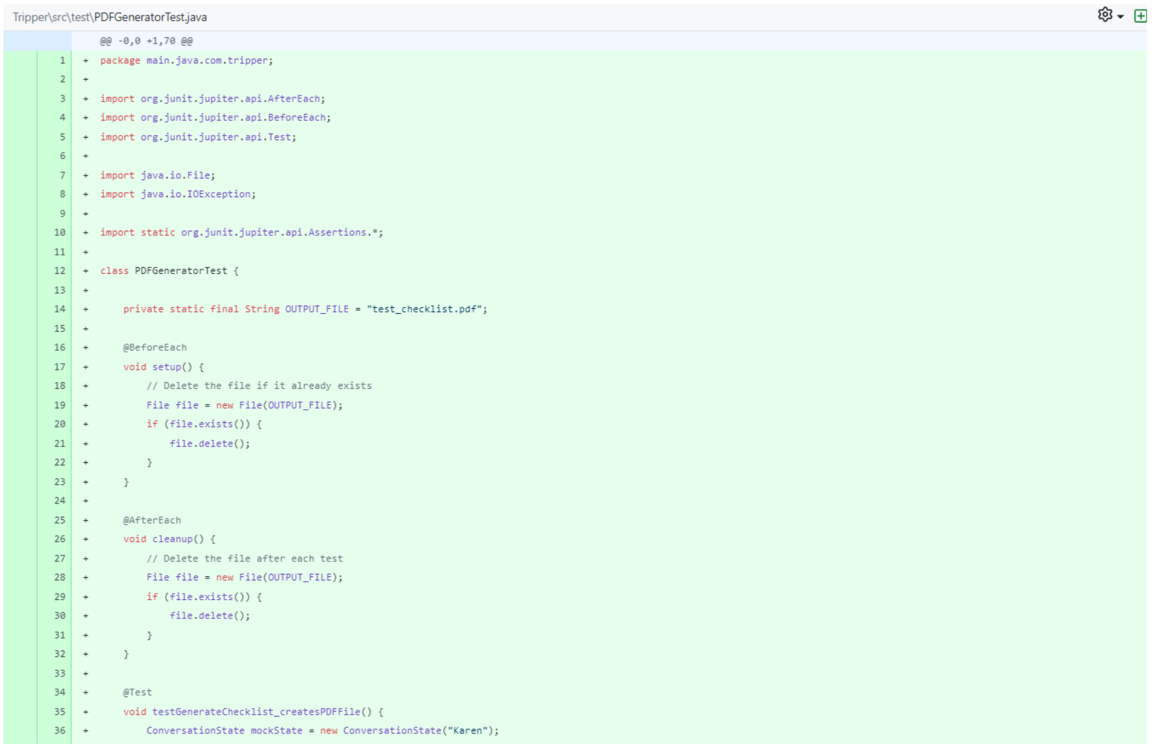
```
Tripper\src\test\TripChecklistTest.java
@@ -0,0 +1,22 @@
+ package main.java.com.tripper;
+
+ import org.junit.jupiter.api.Test;
+ import static org.junit.jupiter.api.Assertions.*;
+
+ class TripChecklistTest {
+
+     @Test
+     void testTripChecklistInitializationAndGetters() {
+         String[] essentials = {"T-shirt", "Jeans"};
+         String[] recommendations = {"Umbrella"};
+         String[] optional = {"Hat"};
+         String[] accessories = {"Sunglasses", "Watch"};
+
+         TripChecklist checklist = new TripChecklist(essentials, recommendations, optional, accessories);
+
+         assertArrayEquals(essentials, checklist.getEssentialItems());
+         assertArrayEquals(recommendations, checklist.getRecommendations());
+         assertArrayEquals(optional, checklist.getOptionalItems());
+         assertArrayEquals(accessories, checklist.getAccessories());
+     }
+ }
```

Figure 15 - Unit test for TripChecklist class



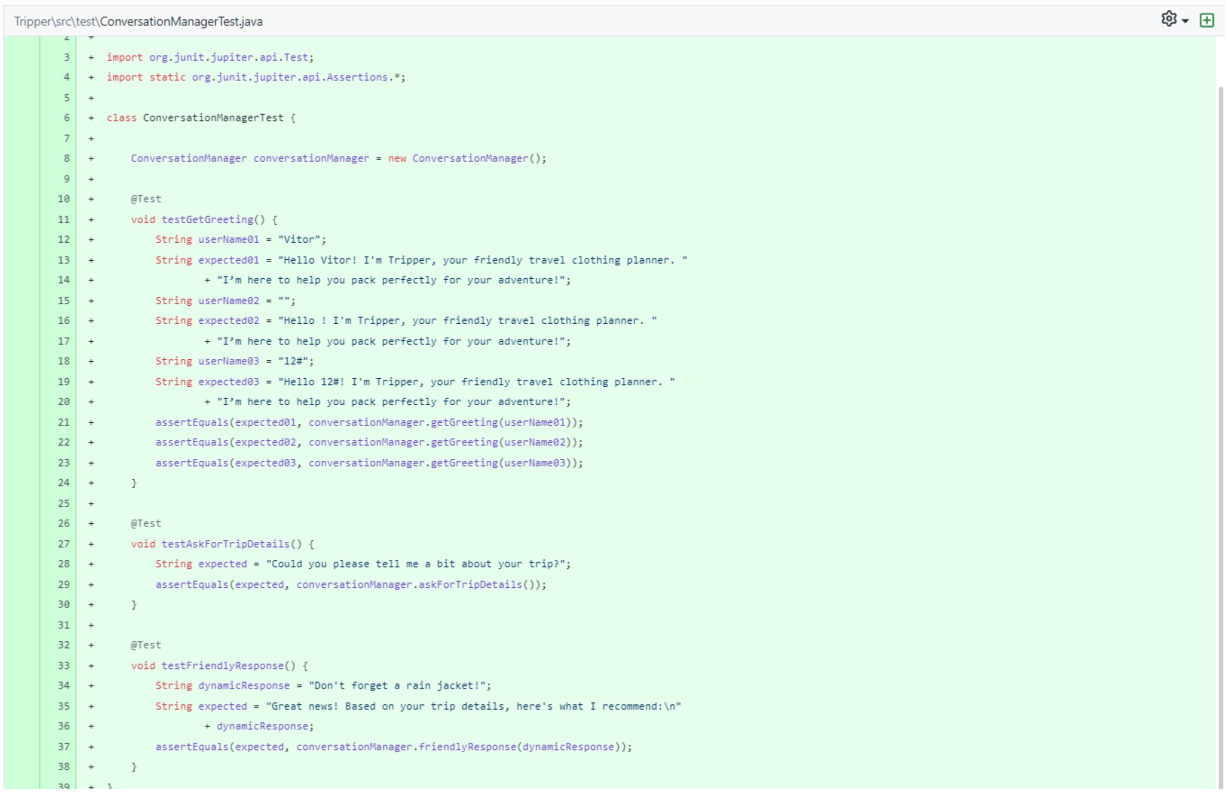
```
Tripper\src\test\TerminalChatbotTest.java
@@ -0,0 +1,12 @@
+ package main.java.com.tripper;
+
+ import org.junit.jupiter.api.Test;
+ import static org.junit.jupiter.api.Assertions.*;
+
+ class TerminalChatbotTest {
+
+     @Test
+     void testMain_runsWithoutException() {
+         assertDoesNotThrow(() -> TerminalChatbot.main(new String[]{}));
+     }
+ }
```

Figure 16 - Unit test for TerminalChatbot class



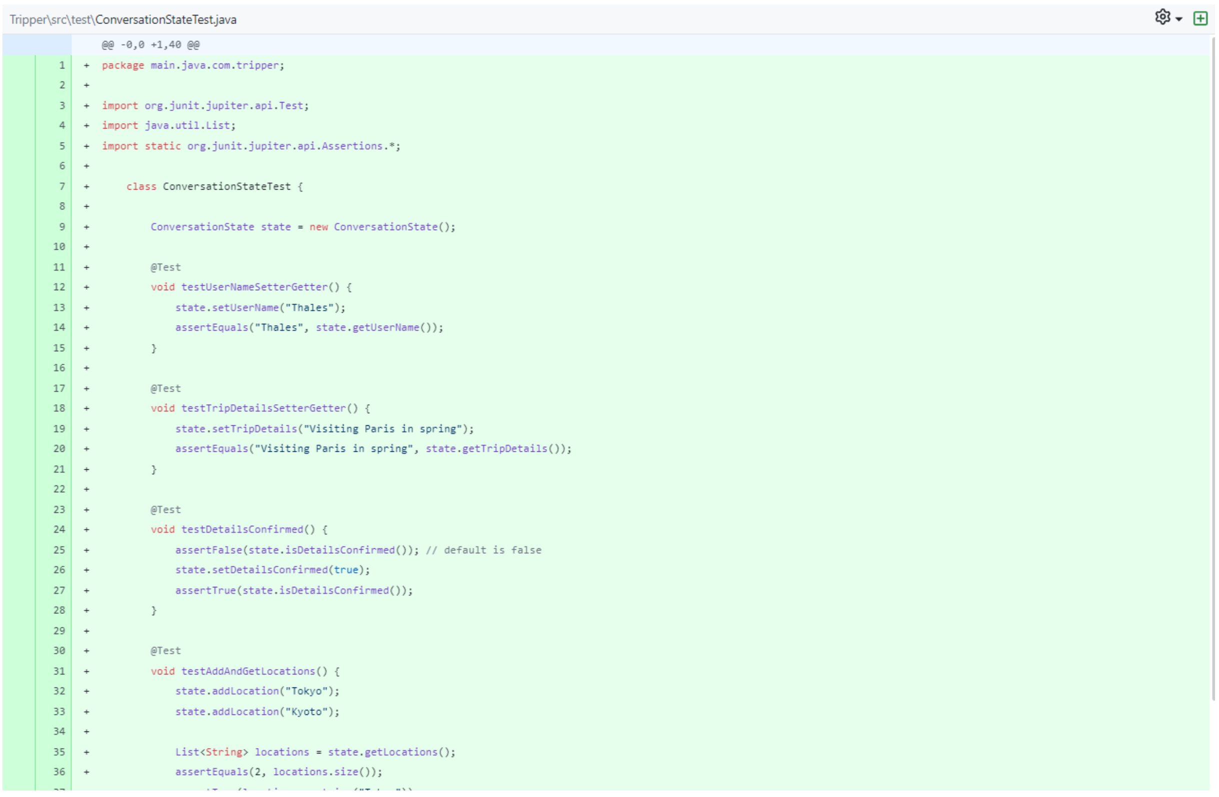
```
Tripper\src\test\PDFGeneratorTest.java
@@ -0,0 +1,70 @@
+ package main.java.com.tripper;
+
+ import org.junit.jupiter.api.AfterEach;
+ import org.junit.jupiter.api.BeforeEach;
+ import org.junit.jupiter.api.Test;
+
+ import java.io.File;
+ import java.io.IOException;
+
+ import static org.junit.jupiter.api.Assertions.*;
+
+ class PDFGeneratorTest {
+
+     private static final String OUTPUT_FILE = "test_checklist.pdf";
+
+     @BeforeEach
+     void setup() {
+         // Delete the file if it already exists
+         File file = new File(OUTPUT_FILE);
+         if (file.exists()) {
+             file.delete();
+         }
+     }
+
+     @AfterEach
+     void cleanup() {
+         // Delete the file after each test
+         File file = new File(OUTPUT_FILE);
+         if (file.exists()) {
+             file.delete();
+         }
+     }
+
+     @Test
+     void testGenerateChecklist_createsPDFFile() {
+         ConversationState mockState = new ConversationState("Karen");
```

Figure 17 - Unit test for PDFGenerator class


Tripper\src\test\ConversationManagerTest.java

```
+
+ import org.junit.jupiter.api.Test;
+ import static org.junit.jupiter.api.Assertions.*;
+
+ class ConversationManagerTest {
+
+     ConversationManager conversationManager = new ConversationManager();
+
+     @Test
+     void testGetGreeting() {
+         String userName01 = "Vitor";
+         String expected01 = "Hello Vitor! I'm Tripper, your friendly travel clothing planner. "
+                 + "I’m here to help you pack perfectly for your adventure!";
+         String userName02 = "";
+         String expected02 = "Hello ! I'm Tripper, your friendly travel clothing planner. "
+                 + "I’m here to help you pack perfectly for your adventure!";
+         String userName03 = "12#";
+         String expected03 = "Hello 12#! I'm Tripper, your friendly travel clothing planner. "
+                 + "I’m here to help you pack perfectly for your adventure!";
+         assertEquals(expected01, conversationManager.getGreeting(userName01));
+         assertEquals(expected02, conversationManager.getGreeting(userName02));
+         assertEquals(expected03, conversationManager.getGreeting(userName03));
+     }
+
+     @Test
+     void testAskForTripDetails() {
+         String expected = "Could you please tell me a bit about your trip?";
+         assertEquals(expected, conversationManager.askForTripDetails());
+     }
+
+     @Test
+     void testFriendlyResponse() {
+         String dynamicResponse = "Don't forget a rain jacket!";
+         String expected = "Great news! Based on your trip details, here's what I recommend:\n"
+                 + dynamicResponse;
+         assertEquals(expected, conversationManager.friendlyResponse(dynamicResponse));
+     }
+ }
```

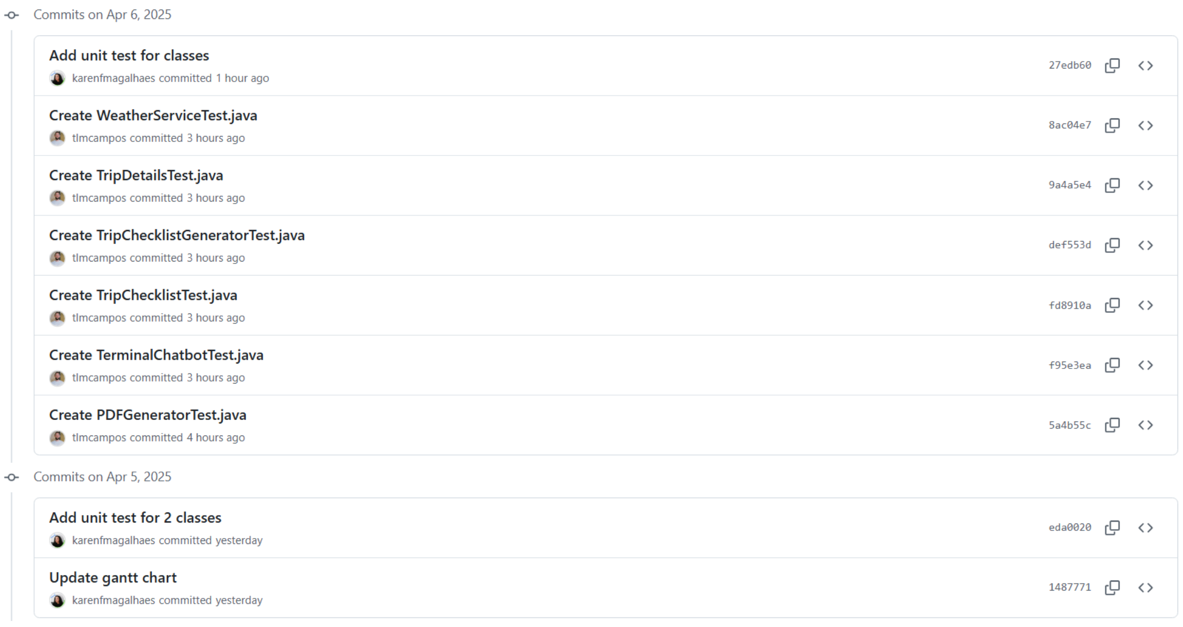
Figure 18 - Unit test for ConversationManager class


Tripper\src\test\ConversationStateTest.java

```
@@ -0,0 +1,40 @@
+ package main.java.com.tripper;
+
+ import org.junit.jupiter.api.Test;
+ import java.util.List;
+ import static org.junit.jupiter.api.Assertions.*;
+
+     class ConversationStateTest {
+
+         ConversationState state = new ConversationState();
+
+         @Test
+         void testUserNameSetterGetter() {
+             state.setUserName("Thales");
+             assertEquals("Thales", state.getUserName());
+         }
+
+         @Test
+         void testTripDetailsSetterGetter() {
+             state.setTripDetails("Visiting Paris in spring");
+             assertEquals("Visiting Paris in spring", state.getTripDetails());
+         }
+
+         @Test
+         void testDetailsConfirmed() {
+             assertFalse(state.isDetailsConfirmed()); // default is false
+             state.setDetailsConfirmed(true);
+             assertTrue(state.isDetailsConfirmed());
+         }
+
+         @Test
+         void testAddAndGetLocations() {
+             state.addLocation("Tokyo");
+             state.addLocation("Kyoto");
+
+             List<String> locations = state.getLocations();
+             assertEquals(2, locations.size());
```

Figure 19 - Unit test for ConversationState class

9.3 Commit List & Branches Tree



Commits on Apr 6, 2025

Add unit test for classes
karenfmagalhaes committed 1 hour ago — 27edb60

Create WeatherServiceTest.java
tlmcampos committed 3 hours ago — 8ac04e7

Create TripDetailsTest.java
tlmcampos committed 3 hours ago — 9a4a5e4

Create TripChecklistGeneratorTest.java
tlmcampos committed 3 hours ago — def553d

Create TripChecklistTest.java
tlmcampos committed 3 hours ago — fd8910a

Create TerminalChatbotTest.java
tlmcampos committed 3 hours ago — f95e3ea

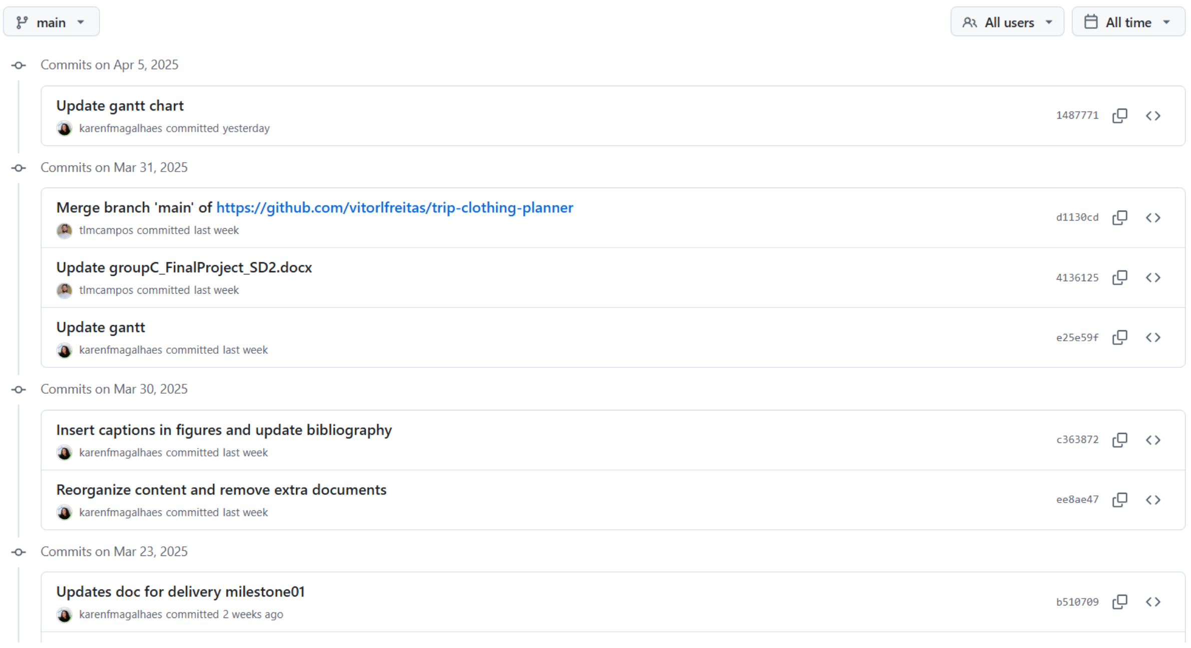
Create PDFGeneratorTest.java
tlmcampos committed 4 hours ago — 5a4b55c

Commits on Apr 5, 2025

Add unit test for 2 classes
karenfmagalhaes committed yesterday — eda0020

Update gantt chart
karenfmagalhaes committed yesterday — 1487771

Figure 20 - Commits branch tests



Commits on Apr 5, 2025

Update gantt chart
karenfmagalhaes committed yesterday — 1487771

Commits on Mar 31, 2025

Merge branch 'main' of https://github.com/vitorlfreitas/trip-clothing-planner
tlmcampos committed last week — d1130cd

Update groupC_FinalProject_SD2.docx
tlmcampos committed last week — 4136125

Update gantt
karenfmagalhaes committed last week — e25e59f

Commits on Mar 30, 2025

Insert captions in figures and update bibliography
karenfmagalhaes committed last week — c363872

Reorganize content and remove extra documents
karenfmagalhaes committed last week — ee8ae47

Commits on Mar 23, 2025

Updates doc for delivery milestone01
karenfmagalhaes committed 2 weeks ago — b510709

Figure 21 - Commits branch main

9.4 Full Log Details

```
commit 27edb601e08bca5a459118282bf139aab5be266f (HEAD -> karen, origin/karen)
Author: Karen F Magalhaes <karenfmagalhaes@gmail.com>
Date:   Sun Apr 6 13:19:32 2025 +0100

    Add unit test for classes

commit 8ac04e754d59998c57fb3a3d957ccf4e2bb7b342
Author: Thales Campos <159160714+tlmcampos@users.noreply.github.com>
Date:   Sun Apr 6 11:07:21 2025 +0100

    Create WeatherServiceTest.java

commit 9a4a5e4d7ea9568b16d90f7835c313c4d1a03159
```

```
Author: Thales Campos <159160714+tlmcampos@users.noreply.github.com>
Date:   Sun Apr 6 11:04:22 2025 +0100

    Create TripDetailsTest.java

commit def553d5a1f60693bc26af9ca04b68e06177fc7e
Author: Thales Campos <159160714+tlmcampos@users.noreply.github.com>
Date:   Sun Apr 6 10:58:27 2025 +0100

    Create TripChecklistGeneratorTest.java
:...skipping...
commit 27edb601e08bca5a459118282bf139aab5be266f (HEAD -> karen, origin/karen)
Author: Karen F Magalhaes <karenfmagalhaes@gmail.com>
Date:   Sun Apr 6 13:19:32 2025 +0100

    Add unit test for classes

commit 8ac04e754d59998c57fb3a3d957ccf4e2bb7b342
Author: Thales Campos <159160714+tlmcampos@users.noreply.github.com>
Date:   Sun Apr 6 11:07:21 2025 +0100

    Create WeatherServiceTest.java

commit 9a4a5e4d7ea9568b16d90f7835c313c4d1a03159
Author: Thales Campos <159160714+tlmcampos@users.noreply.github.com>
Date:   Sun Apr 6 11:04:22 2025 +0100

    Create TripDetailsTest.java

commit def553d5a1f60693bc26af9ca04b68e06177fc7e
Author: Thales Campos <159160714+tlmcampos@users.noreply.github.com>
Date:   Sun Apr 6 10:58:27 2025 +0100

    Create TripChecklistGeneratorTest.java

commit fd8910a5736d7434bedb1305e7169df03873827a
Author: Thales Campos <159160714+tlmcampos@users.noreply.github.com>
Date:   Sun Apr 6 10:54:31 2025 +0100

    Create TripChecklistTest.java

commit f95e3ea437de5542bd307ef08aa9f0a491db3646
Author: Thales Campos <159160714+tlmcampos@users.noreply.github.com>
Date:   Sun Apr 6 10:52:04 2025 +0100

    Create TerminalChatbotTest.java

commit 5a4b55cc48975685f06b144dbee051ca578b4dfd
Author: Thales Campos <159160714+tlmcampos@users.noreply.github.com>
Date:   Sun Apr 6 10:43:12 2025 +0100

    Create PDFGeneratorTest.java

commit eda0020c612ef2b4547c98859eb37552c7a530ba
Author: Karen F Magalhaes <karenfmagalhaes@gmail.com>
Date:   Sat Apr 5 17:12:35 2025 +0100

    Add unit test for 2 classes

commit 1487771e5c6138c51453ea8ae6c4c5b903a223a1 (origin/main, origin/HEAD, main)
Author: Karen F Magalhaes <karenfmagalhaes@gmail.com>
Date:   Sat Apr 5 11:17:01 2025 +0100

    Update gantt chart

commit d1130cd45541f1ab3d73b3fbe66059b08e9a6f03
Merge: 4136125 e25e59f
Author: Thales Campos <159160714+tlmcampos@users.noreply.github.com>
Date:   Mon Mar 31 10:32:41 2025 +0100

    Merge branch 'main' of https://github.com/vitorlfreitas/trip-clothing-planner

commit 4136125196503bed5ca283f28731fffcb41fb0b7
Author: Thales Campos <159160714+tlmcampos@users.noreply.github.com>
Date:   Mon Mar 31 10:32:22 2025 +0100

    Update groupC_FinalProject_SD2.docx

commit e25e59fdb9001327419c0cf605d5501fa9458678
Author: Karen F Magalhaes <karenfmagalhaes@gmail.com>
```

```
Date:   Mon Mar 31 10:30:16 2025 +0100

    Update gantt

commit c36387230b07d8703818c8662192ca8b17d99806
Author: Karen F Magalhaes <karenfmagalhaes@gmail.com>
Date:   Sun Mar 30 10:50:45 2025 +0100

    Insert captions in figures and update bibliography

commit ee8ae4709d559185cc9a084e0f47a2c496f6047c
Author: Karen F Magalhaes <karenfmagalhaes@gmail.com>
Date:   Sun Mar 30 10:43:08 2025 +0100

    Reorganize content and remove extra documents

commit b5107099e04c366d61b0dc0ebca3f6117828d301
Author: Karen F Magalhaes <karenfmagalhaes@gmail.com>
Date:   Sun Mar 23 21:38:57 2025 +0000

    Updates doc for delivery milestone01
```

10. Milestone 3

10.1 Goals

Tripper App — Milestone 03

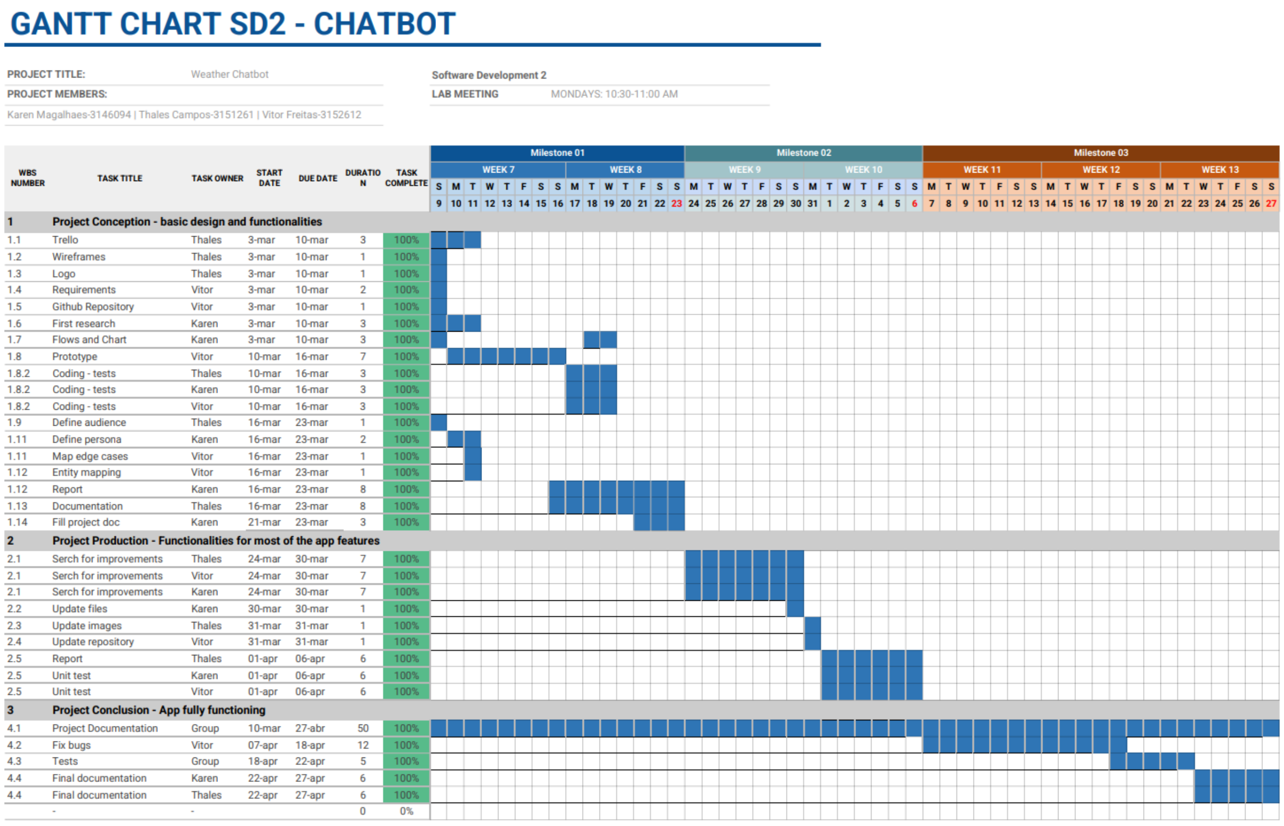
This project is a Java application built with the Spring Boot framework, named "Tripper". The Tripper App's goal is to create and deliver a simple and smart trip planning assistant that suggests clothing recommendations, tips about the weather, and brief information based on weather conditions at their travel destination. The app integrates Natural Language Processing (NLP) to understand user input. It uses real-time data, such as location validation and weather APIs, to provide personalized advice and allow the user to download it in PDF. Our objective was to create an intuitive, fully functioning chatbot application that can help travellers in an efficient and user-friendly way.

To make it work, some requirements were followed as: full integration with Google Maps API for location validation, implementation of OpenNLP for parsing user inputs and extracting important travel details (like destination and travel month), development of a responsive backend using Spring Boot, database integration using JPA and MySQL for storing user queries and trip data, PDF generation feature using iText library to export trip recommendations, error handling and logging improvements for better maintainability. It contains various dependencies to provide features such as web development (Spring MVC), data persistence (JPA with MySQL), and natural language processing. The application also integrates tools like Lombok to reduce boilerplate code and MapStruct for object mapping.

Some changes made since milestone 02: integrated geocoding validation to verify user-provided locations before processing trip data, added input parsing improvements to better detect dates and multi-word location names, improved error handling when external APIs fail (such as the Maps API), finalized project documentation and cleaned up the codebase for better structure and clarity.

10.2 Updates

Figure 22 - Chatbot logo updated using ChatGPT



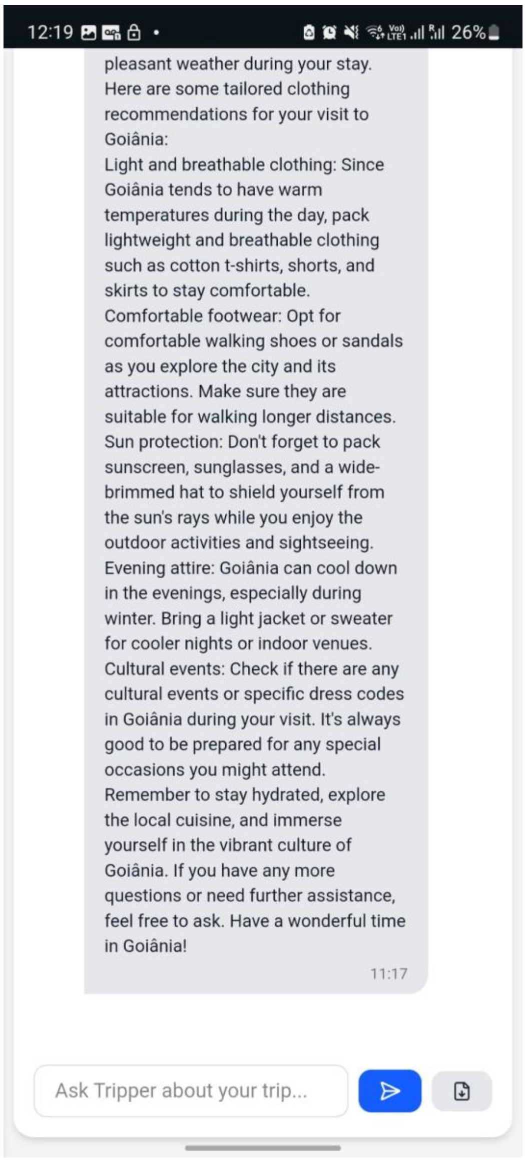
GANTT CHART SD2 - CHATBOT

PROJECT TITLE:	Weather Chatbot	Software Development 2	
PROJECT MEMBERS:		LAB MEETING	MONDAYS: 10:30-11:00 AM
Karen Magalhaes-3146094 | Thales Campos-3151261 | Vitor Freitas-3152612			

WBS NUMBER	TASK TITLE	TASK OWNER	START DATE	DUE DATE	DURATION	TASK COMPLETE
1	**Project Conception - basic design and functionalities**					
1.1	Trello	Thales	3-mar	10-mar	3	100%
1.2	Wireframes	Thales	3-mar	10-mar	1	100%
1.3	Logo	Thales	3-mar	10-mar	1	100%
1.4	Requirements	Vitor	3-mar	10-mar	2	100%
1.5	Github Repository	Vitor	3-mar	10-mar	1	100%
1.6	First research	Karen	3-mar	10-mar	3	100%
1.7	Flows and Chart	Karen	3-mar	10-mar	3	100%
1.8	Prototype	Vitor	10-mar	16-mar	7	100%
1.8.2	Coding - tests	Thales	10-mar	16-mar	3	100%
1.8.2	Coding - tests	Karen	10-mar	16-mar	3	100%
1.8.2	Coding - tests	Vitor	10-mar	16-mar	3	100%
1.9	Define audience	Thales	16-mar	23-mar	1	100%
1.11	Define persona	Karen	16-mar	23-mar	2	100%
1.11	Map edge cases	Vitor	16-mar	23-mar	1	100%
1.12	Entity mapping	Vitor	16-mar	23-mar	1	100%
1.12	Report	Karen	16-mar	23-mar	8	100%
1.13	Documentation	Thales	16-mar	23-mar	8	100%
1.14	Fill project doc	Karen	21-mar	23-mar	3	100%
2	**Project Production - Functionalities for most of the app features**					
2.1	Serch for improvements	Thales	24-mar	30-mar	7	100%
2.1	Serch for improvements	Vitor	24-mar	30-mar	7	100%
2.1	Serch for improvements	Karen	24-mar	30-mar	7	100%
2.2	Update files	Karen	30-mar	30-mar	1	100%
2.3	Update images	Thales	31-mar	31-mar	1	100%
2.4	Update repository	Vitor	31-mar	31-mar	1	100%
2.5	Report	Thales	01-apr	06-apr	6	100%
2.5	Unit test	Karen	01-apr	06-apr	6	100%
2.5	Unit test	Vitor	01-apr	06-apr	6	100%
3	**Project Conclusion - App fully functioning**					
4.1	Project Documentation	Group	10-mar	27-abr	50	100%
4.2	Fix bugs	Vitor	07-apr	18-apr	12	100%
4.3	Tests	Group	18-apr	22-apr	5	100%
4.4	Final documentation	Karen	22-apr	27-apr	6	100%
4.4	Final documentation	Thales	22-apr	27-apr	6	100%
	-	-			0	0%

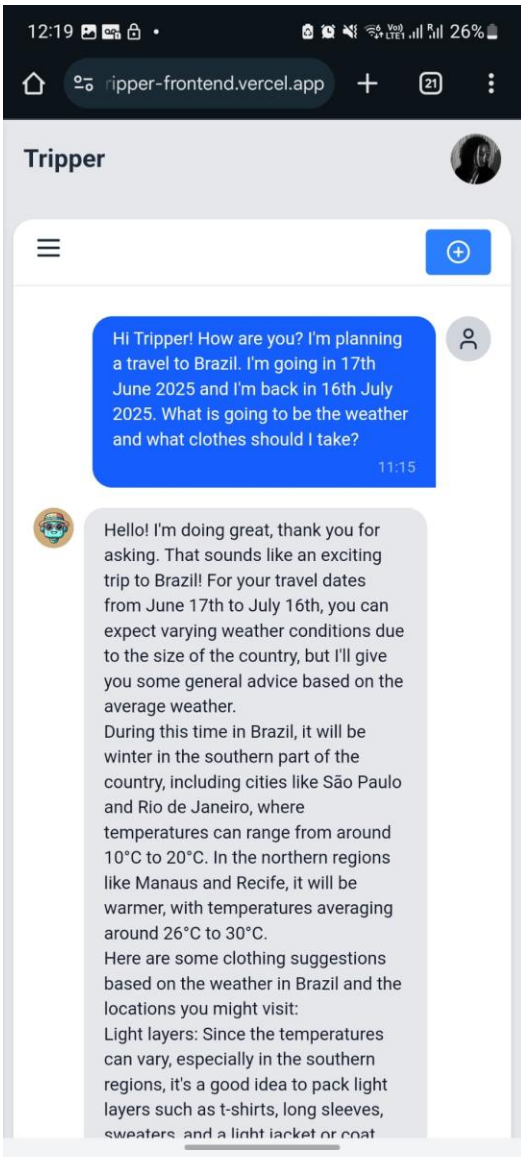
Figure 23 - Final gantt chart

10.3 App tests / Interface



12:19 26%

pleasant weather during your stay.
Here are some tailored clothing recommendations for your visit to Goiânia:
Light and breathable clothing: Since Goiânia tends to have warm temperatures during the day, pack lightweight and breathable clothing such as cotton t-shirts, shorts, and skirts to stay comfortable.
Comfortable footwear: Opt for comfortable walking shoes or sandals as you explore the city and its attractions. Make sure they are suitable for walking longer distances.
Sun protection: Don't forget to pack sunscreen, sunglasses, and a wide-brimmed hat to shield yourself from the sun's rays while you enjoy the outdoor activities and sightseeing.
Evening attire: Goiânia can cool down in the evenings, especially during winter. Bring a light jacket or sweater for cooler nights or indoor venues.
Cultural events: Check if there are any cultural events or specific dress codes in Goiânia during your visit. It's always good to be prepared for any special occasions you might attend.
Remember to stay hydrated, explore the local cuisine, and immerse yourself in the vibrant culture of Goiânia. If you have any more questions or need further assistance, feel free to ask. Have a wonderful time in Goiânia!
11:17



12:19 26%

Hi Tripper! How are you? I'm planning a travel to Brazil. I'm going in 17th June 2025 and I'm back in 16th July 2025. What is going to be the weather and what clothes should I take?
11:15

Hello! I'm doing great, thank you for asking. That sounds like an exciting trip to Brazil! For your travel dates from June 17th to July 16th, you can expect varying weather conditions due to the size of the country, but I'll give you some general advice based on the average weather.
During this time in Brazil, it will be winter in the southern part of the country, including cities like São Paulo and Rio de Janeiro, where temperatures can range from around 10°C to 20°C. In the northern regions like Manaus and Recife, it will be warmer, with temperatures averaging around 26°C to 30°C.
Here are some clothing suggestions based on the weather in Brazil and the locations you might visit:
Light layers: Since the temperatures can vary, especially in the southern regions, it's a good idea to pack light layers such as t-shirts, long sleeves, sweaters and a light jacket or coat

Figure 24 - Application test

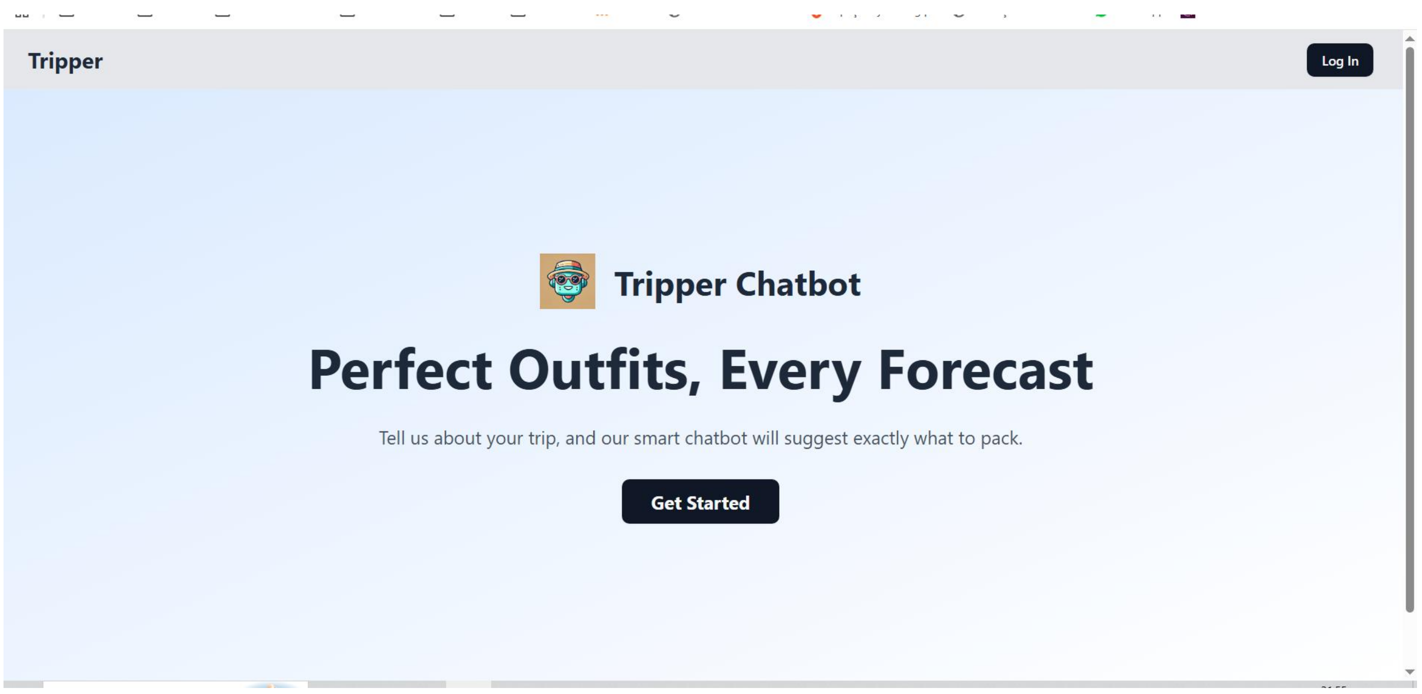

Tripper
Log In
Tripper Chatbot
Perfect Outfits, Every Forecast
Tell us about your trip, and our smart chatbot will suggest exactly what to pack.
Get Started

Figure 25 - Landing page

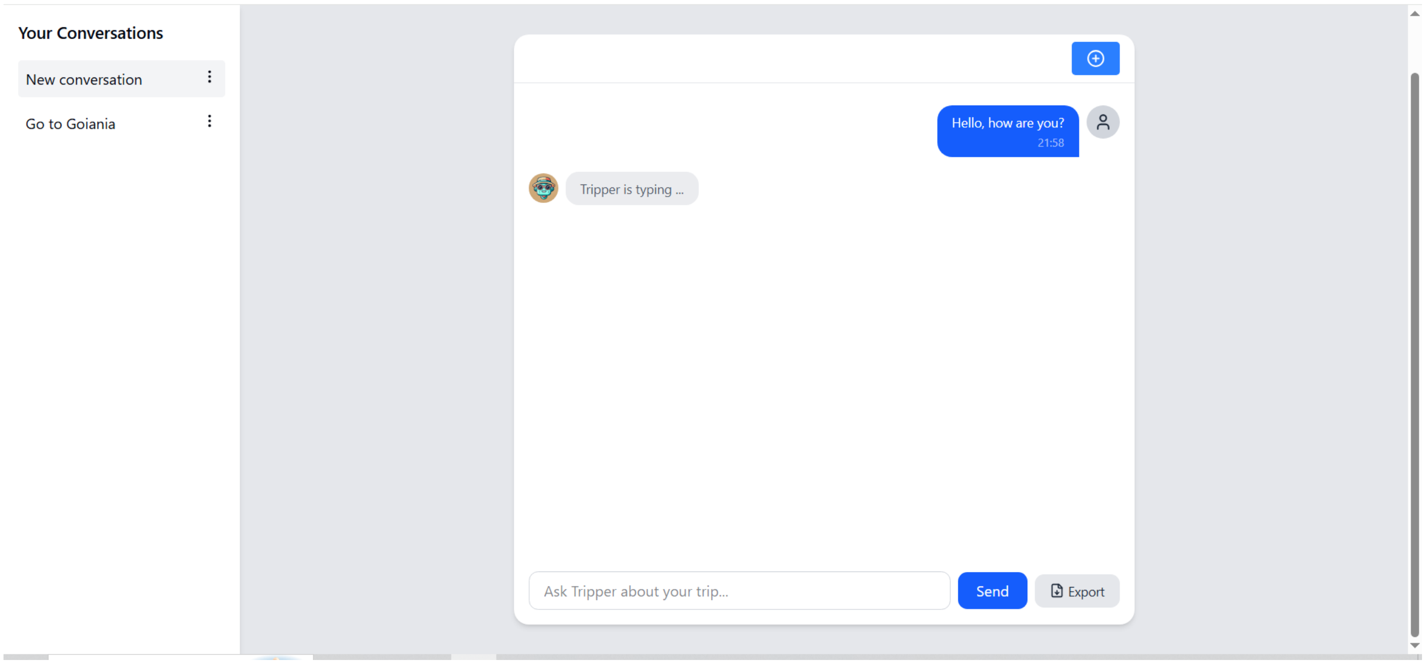
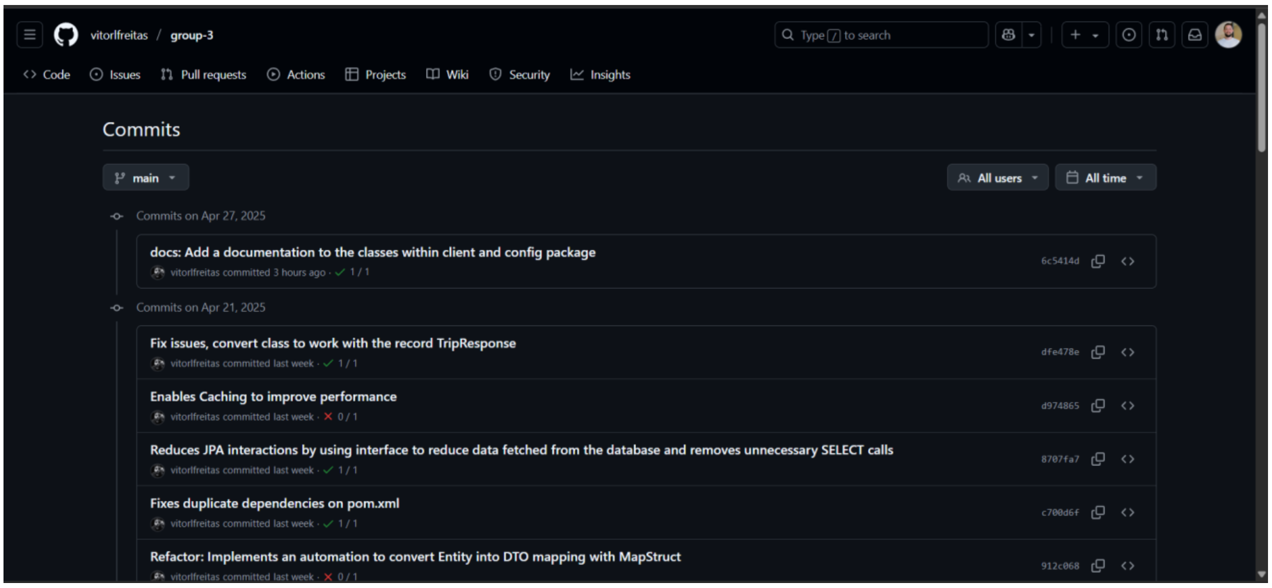


Figure 26 - User interaction

10.4 Commit List & Branches Tree


vitorlfreitas / group-3

Commits

main — All users — All time

Commits on Apr 27, 2025

docs: Add a documentation to the classes within client and config package
vitorlfreitas committed 3 hours ago · ✓ 1 / 1 — 6c5414d

Commits on Apr 21, 2025

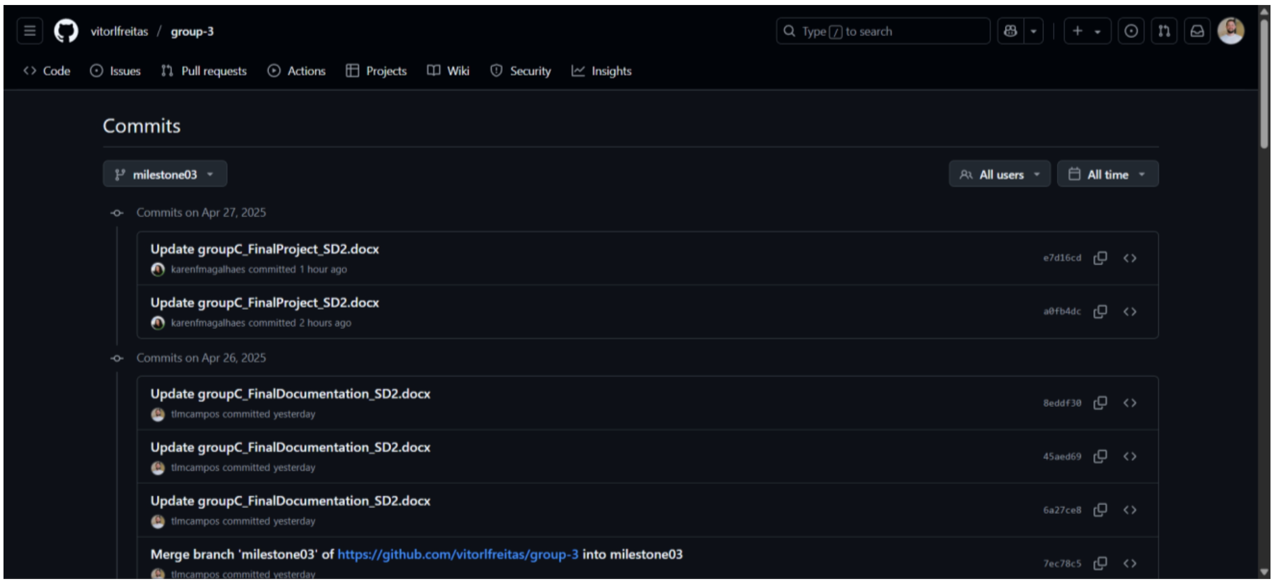
Fix issues, convert class to work with the record TripResponse
vitorlfreitas committed last week · ✓ 1 / 1 — dfe478e

Enables Caching to improve performance
vitorlfreitas committed last week · ✗ 0 / 1 — d974865

Reduces JPA interactions by using interface to reduce data fetched from the database and removes unnecessary SELECT calls
vitorlfreitas committed last week · ✓ 1 / 1 — 8707fa7

Fixes duplicate dependencies on pom.xml
vitorlfreitas committed last week · ✓ 1 / 1 — c700d6f

Refactor: Implements an automation to convert Entity into DTO mapping with MapStruct
vitorlfreitas committed last week · ✗ 0 / 1 — 912c068


vitorlfreitas / group-3

Commits

milestone03 — All users — All time

Commits on Apr 27, 2025

Update groupC_FinalProject_SD2.docx
karenfmagalhaes committed 1 hour ago — e7d16cd

Update groupC_FinalProject_SD2.docx
karenfmagalhaes committed 2 hours ago — a0fb4dc

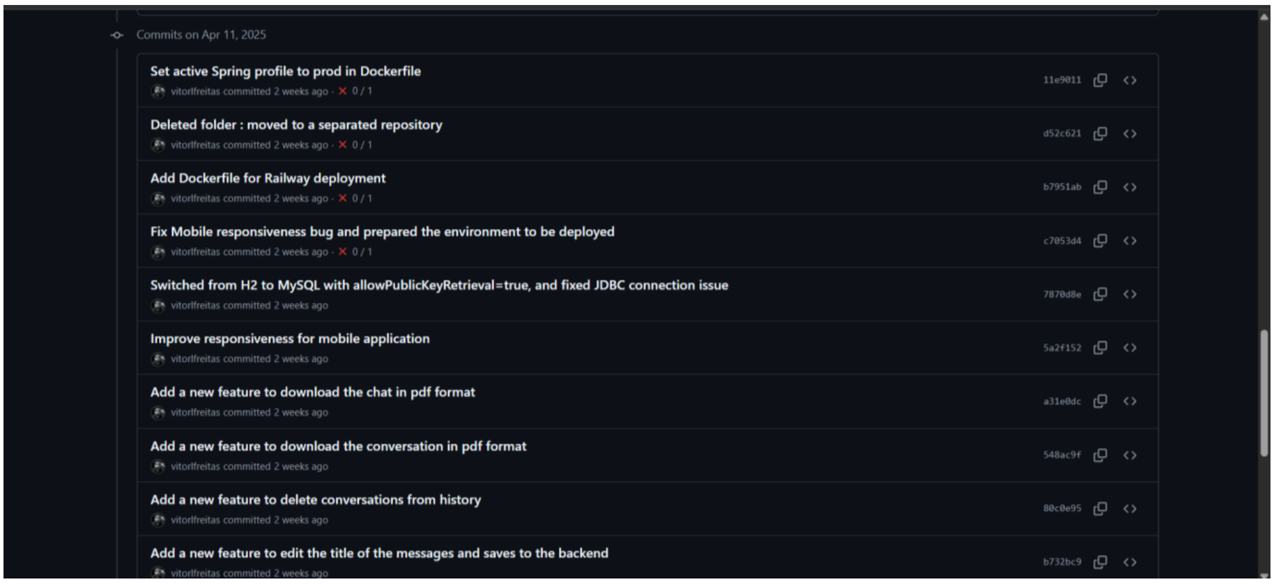
Commits on Apr 26, 2025

Update groupC_FinalDocumentation_SD2.docx
tlmcampos committed yesterday — 8eddf30

Update groupC_FinalDocumentation_SD2.docx
tlmcampos committed yesterday — 45aed69

Update groupC_FinalDocumentation_SD2.docx
tlmcampos committed yesterday — 6a27ce8

Merge branch 'milestone03' of https://github.com/vitorlfreitas/group-3 into milestone03
tlmcampos committed yesterday — 7ec78c5


Commits on Apr 11, 2025

Set active Spring profile to prod in Dockerfile
vitorlfreitas committed 2 weeks ago · ✗ 0 / 1 — 11e9011

Deleted folder : moved to a separated repository
vitorlfreitas committed 2 weeks ago · ✗ 0 / 1 — d52c621

Add Dockerfile for Railway deployment
vitorlfreitas committed 2 weeks ago · ✗ 0 / 1 — b7951ab

Fix Mobile responsiveness bug and prepared the environment to be deployed
vitorlfreitas committed 2 weeks ago · ✗ 0 / 1 — c7053d4

Switched from H2 to MySQL with allowPublicKeyRetrieval=true, and fixed JDBC connection issue
vitorlfreitas committed 2 weeks ago — 7870d8e

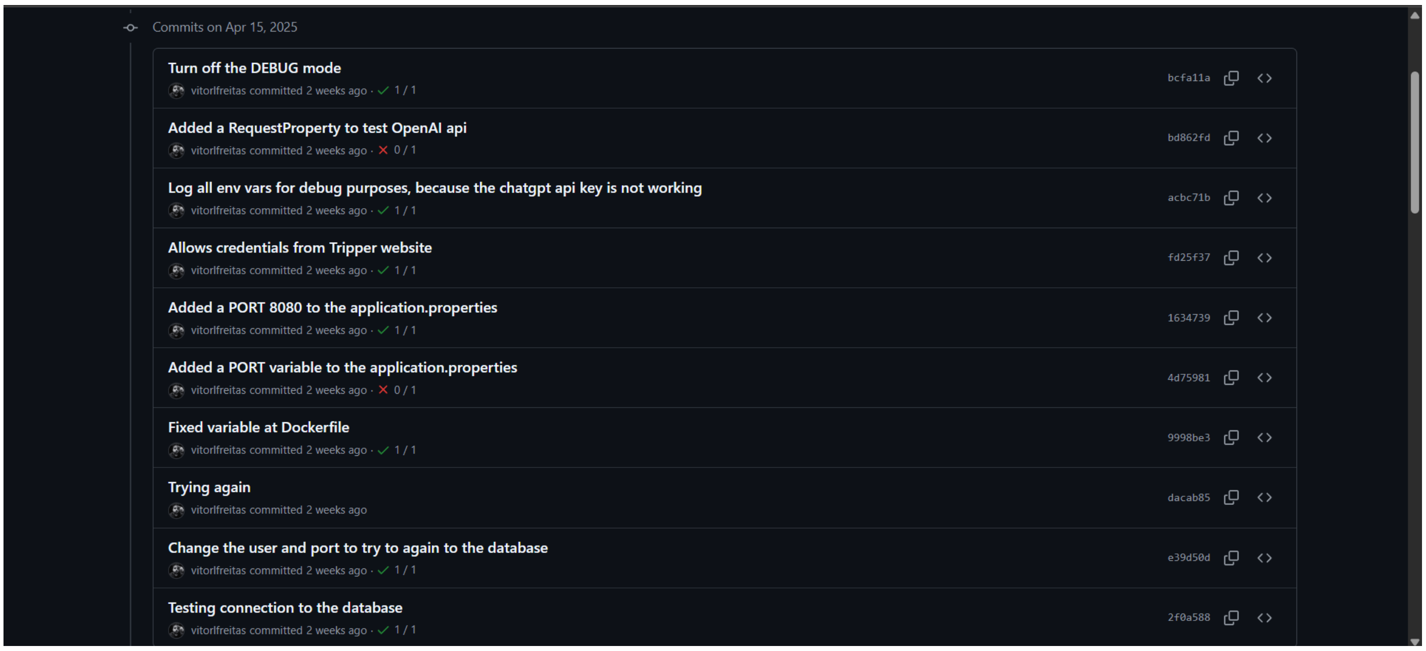
Improve responsiveness for mobile application
vitorlfreitas committed 2 weeks ago — 5a2f152

Add a new feature to download the chat in pdf format
vitorlfreitas committed 2 weeks ago — a31e0dc

Add a new feature to download the conversation in pdf format
vitorlfreitas committed 2 weeks ago — 548ac9f

Add a new feature to delete conversations from history
vitorlfreitas committed 2 weeks ago — 80c0e95

Add a new feature to edit the title of the messages and saves to the backend
vitorlfreitas committed 2 weeks ago — b732bc9


Commits on Apr 15, 2025

Turn off the DEBUG mode — bcfa11a
vitorlfreitas committed 2 weeks ago · ✓ 1 / 1

Added a RequestProperty to test OpenAI api — bd862fd
vitorlfreitas committed 2 weeks ago · ✗ 0 / 1

Log all env vars for debug purposes, because the chatgpt api key is not working — acbc71b
vitorlfreitas committed 2 weeks ago · ✓ 1 / 1

Allows credentials from Tripper website — fd25f37
vitorlfreitas committed 2 weeks ago · ✓ 1 / 1

Added a PORT 8080 to the application.properties — 1634739
vitorlfreitas committed 2 weeks ago · ✓ 1 / 1

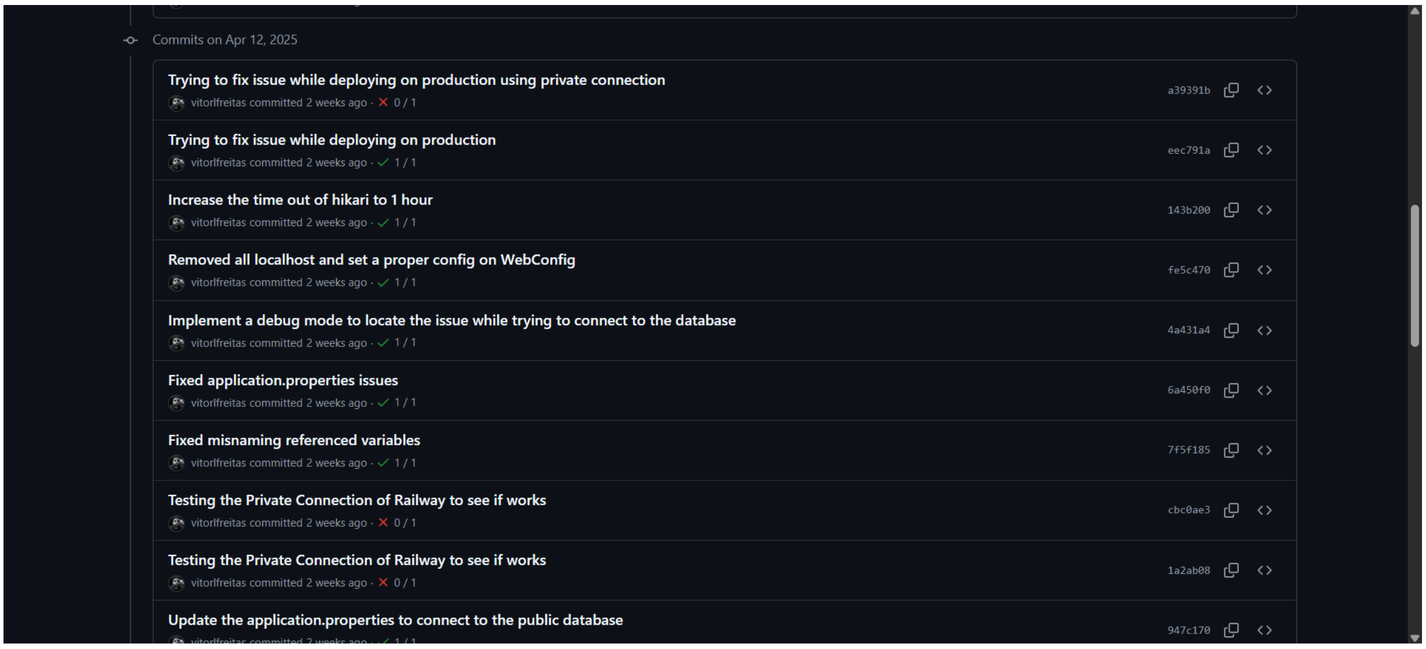
Added a PORT variable to the application.properties — 4d75981
vitorlfreitas committed 2 weeks ago · ✗ 0 / 1

Fixed variable at Dockerfile — 9998be3
vitorlfreitas committed 2 weeks ago · ✓ 1 / 1

Trying again — dacab85
vitorlfreitas committed 2 weeks ago

Change the user and port to try to again to the database — e39d50d
vitorlfreitas committed 2 weeks ago · ✓ 1 / 1

Testing connection to the database — 2f0a588
vitorlfreitas committed 2 weeks ago · ✓ 1 / 1


Commits on Apr 12, 2025

Trying to fix issue while deploying on production using private connection — a39391b
vitorlfreitas committed 2 weeks ago · ✗ 0 / 1

Trying to fix issue while deploying on production — eec791a
vitorlfreitas committed 2 weeks ago · ✓ 1 / 1

Increase the time out of hikari to 1 hour — 143b200
vitorlfreitas committed 2 weeks ago · ✓ 1 / 1

Removed all localhost and set a proper config on WebConfig — fe5c470
vitorlfreitas committed 2 weeks ago · ✓ 1 / 1

Implement a debug mode to locate the issue while trying to connect to the database — 4a431a4
vitorlfreitas committed 2 weeks ago · ✓ 1 / 1

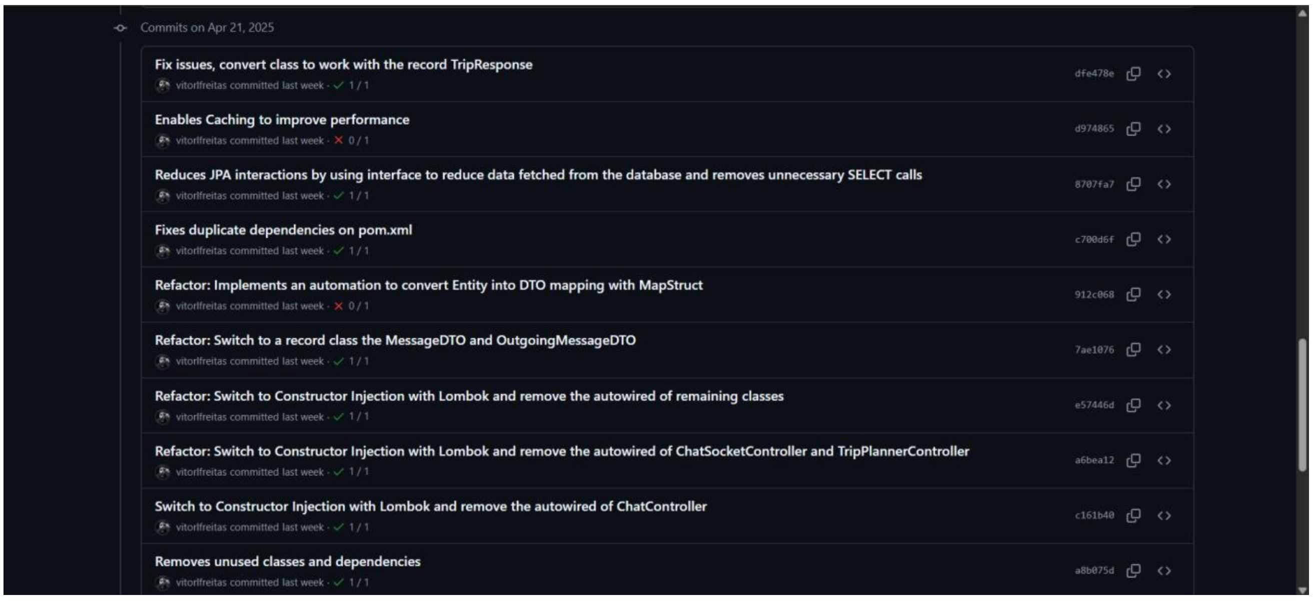
Fixed application.properties issues — 6a450f0
vitorlfreitas committed 2 weeks ago · ✓ 1 / 1

Fixed misnaming referenced variables — 7f5f185
vitorlfreitas committed 2 weeks ago · ✓ 1 / 1

Testing the Private Connection of Railway to see if works — cbc0ae3
vitorlfreitas committed 2 weeks ago · ✗ 0 / 1

Testing the Private Connection of Railway to see if works — 1a2ab08
vitorlfreitas committed 2 weeks ago · ✗ 0 / 1

Update the application.properties to connect to the public database — 947c170


Commits on Apr 21, 2025

Fix issues, convert class to work with the record TripResponse — dfe478e
vitorlfreitas committed last week · ✓ 1 / 1

Enables Caching to improve performance — d974865
vitorlfreitas committed last week · ✗ 0 / 1

Reduces JPA interactions by using interface to reduce data fetched from the database and removes unnecessary SELECT calls — 8707fa7
vitorlfreitas committed last week · ✓ 1 / 1

Fixes duplicate dependencies on pom.xml — c700d6f
vitorlfreitas committed last week · ✓ 1 / 1

Refactor: Implements an automation to convert Entity into DTO mapping with MapStruct — 912c068
vitorlfreitas committed last week · ✗ 0 / 1

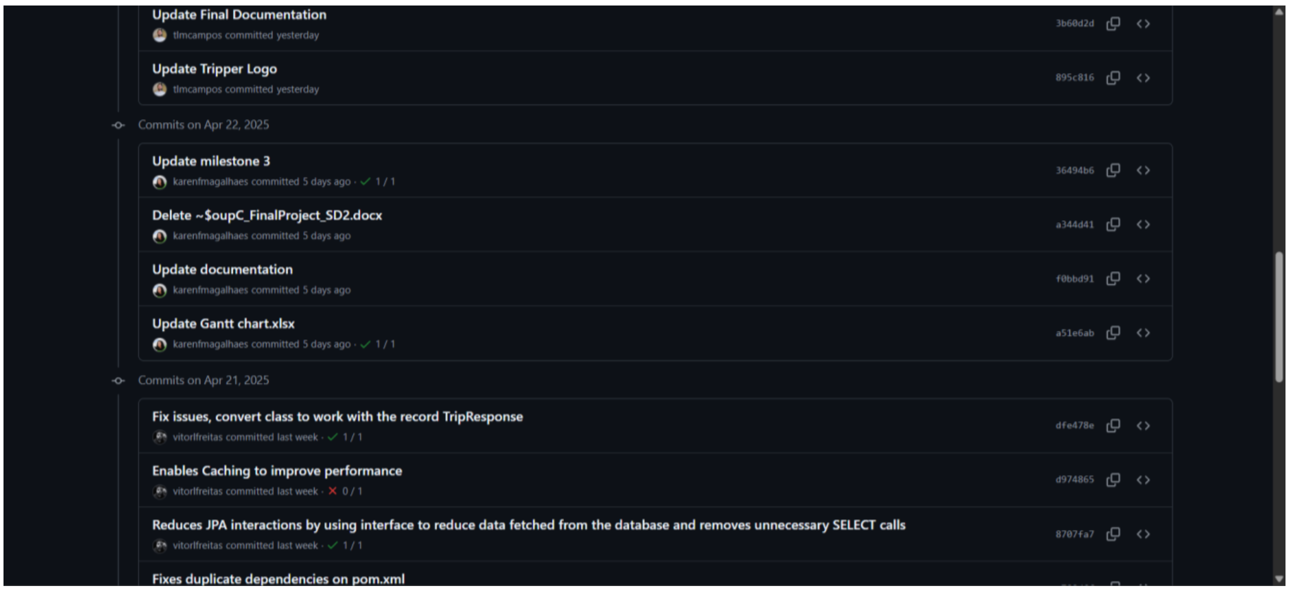
Refactor: Switch to a record class the MessageDTO and OutgoingMessageDTO — 7ae1076
vitorlfreitas committed last week · ✓ 1 / 1

Refactor: Switch to Constructor Injection with Lombok and remove the autowired of remaining classes — e57446d
vitorlfreitas committed last week · ✓ 1 / 1

Refactor: Switch to Constructor Injection with Lombok and remove the autowired of ChatSocketController and TripPlannerController — a6bea12
vitorlfreitas committed last week · ✓ 1 / 1

Switch to Constructor Injection with Lombok and remove the autowired of ChatController — c161b40
vitorlfreitas committed last week · ✓ 1 / 1

Removes unused classes and dependencies — a8b075d
vitorlfreitas committed last week · ✓ 1 / 1


Update Final Documentation
tlmcampos committed yesterday 3b60d2d

Update Tripper Logo
tlmcampos committed yesterday 895c816

Commits on Apr 22, 2025

Update milestone 3
karenfmagalhaes committed 5 days ago · ✓ 1 / 1 36494b6

Delete ~$oupC_FinalProject_SD2.docx
karenfmagalhaes committed 5 days ago a344d41

Update documentation
karenfmagalhaes committed 5 days ago f0bbd91

Update Gantt chart.xlsx
karenfmagalhaes committed 5 days ago · ✓ 1 / 1 a51e6ab

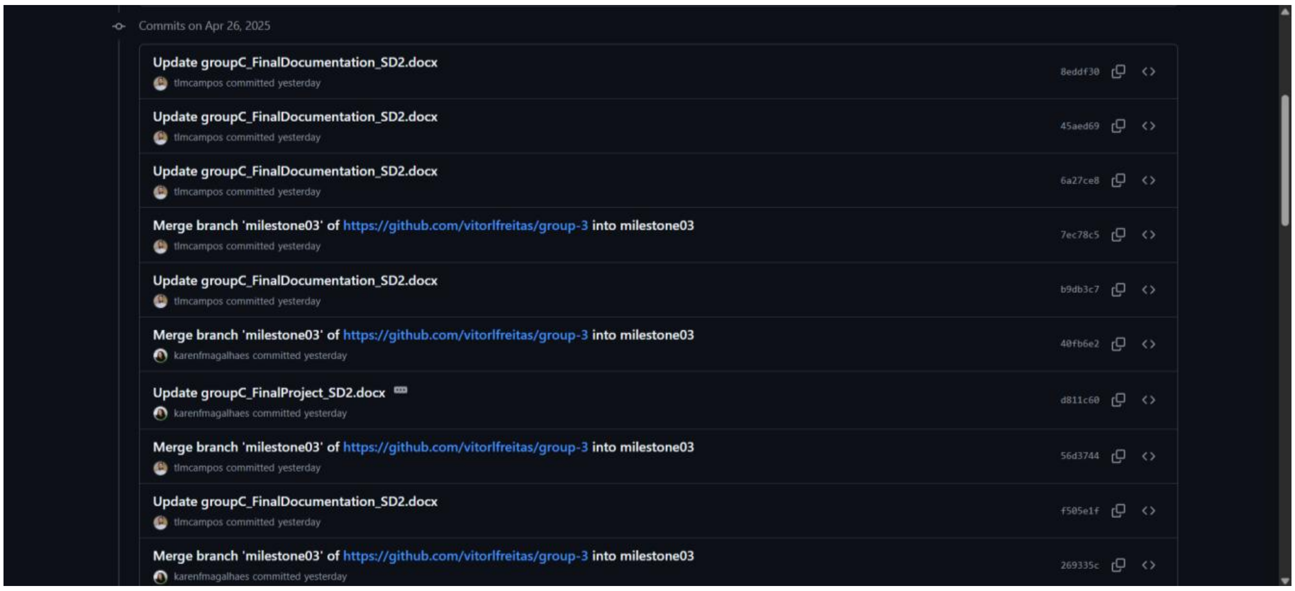
Commits on Apr 21, 2025

Fix issues, convert class to work with the record TripResponse
vitorlfreitas committed last week · ✓ 1 / 1 dfe478e

Enables Caching to improve performance
vitorlfreitas committed last week · ✗ 0 / 1 d974865

Reduces JPA interactions by using interface to reduce data fetched from the database and removes unnecessary SELECT calls
vitorlfreitas committed last week · ✓ 1 / 1 8707fa7

Fixes duplicate dependencies on pom.xml


Commits on Apr 26, 2025

Update groupC_FinalDocumentation_SD2.docx
tlmcampos committed yesterday 8eddf30

Update groupC_FinalDocumentation_SD2.docx
tlmcampos committed yesterday 45aed69

Update groupC_FinalDocumentation_SD2.docx
tlmcampos committed yesterday 6a27ce8

Merge branch 'milestone03' of https://github.com/vitorlfreitas/group-3 into milestone03
tlmcampos committed yesterday 7ec78c5

Update groupC_FinalDocumentation_SD2.docx
tlmcampos committed yesterday b9db3c7

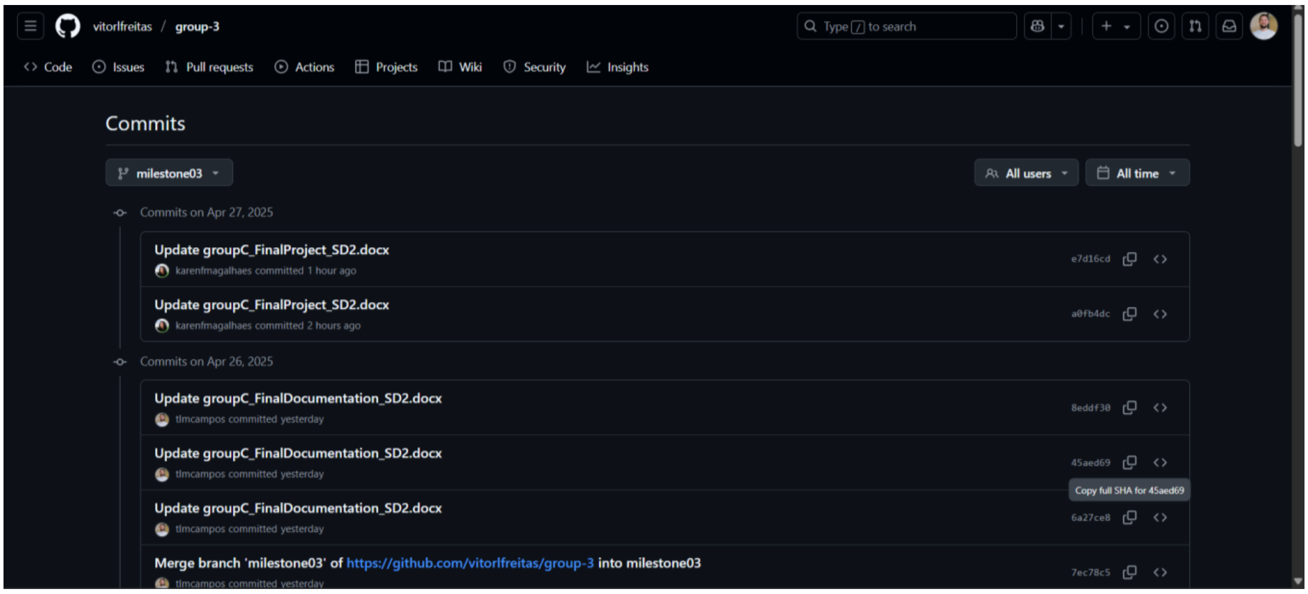
Merge branch 'milestone03' of https://github.com/vitorlfreitas/group-3 into milestone03
karenfmagalhaes committed yesterday 40fb6e2

Update groupC_FinalProject_SD2.docx
karenfmagalhaes committed yesterday d811c60

Merge branch 'milestone03' of https://github.com/vitorlfreitas/group-3 into milestone03
tlmcampos committed yesterday 56d3744

Update groupC_FinalDocumentation_SD2.docx
tlmcampos committed yesterday f505e1f

Merge branch 'milestone03' of https://github.com/vitorlfreitas/group-3 into milestone03
karenfmagalhaes committed yesterday 269335c



Commits

milestone03 · All users · All time

Commits on Apr 27, 2025

Update groupC_FinalProject_SD2.docx
karenfmagalhaes committed 1 hour ago e7d16cd

Update groupC_FinalProject_SD2.docx
karenfmagalhaes committed 2 hours ago a0fb4dc

Commits on Apr 26, 2025

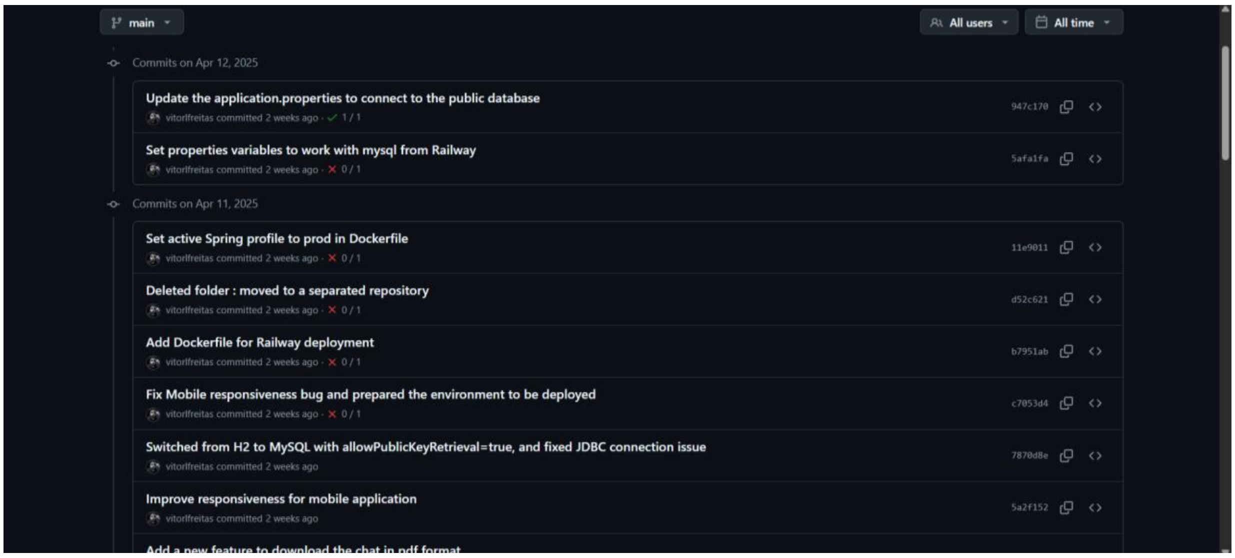
Update groupC_FinalDocumentation_SD2.docx
tlmcampos committed yesterday 8eddf30

Update groupC_FinalDocumentation_SD2.docx
tlmcampos committed yesterday 45aed69

Copy full SHA for 45aed69

Update groupC_FinalDocumentation_SD2.docx
tlmcampos committed yesterday 6a27ce8

Merge branch 'milestone03' of https://github.com/vitorlfreitas/group-3 into milestone03
tlmcampos committed yesterday 7ec78c5



main · All users · All time

Commits on Apr 12, 2025

Update the application.properties to connect to the public database
vitorlfreitas committed 2 weeks ago · ✓ 1 / 1 — 947c170

Set properties variables to work with mysql from Railway
vitorlfreitas committed 2 weeks ago · ✗ 0 / 1 — 5afa1fa

Commits on Apr 11, 2025

Set active Spring profile to prod in Dockerfile
vitorlfreitas committed 2 weeks ago · ✗ 0 / 1 — 11e9011

Deleted folder : moved to a separated repository
vitorlfreitas committed 2 weeks ago · ✗ 0 / 1 — d52c621

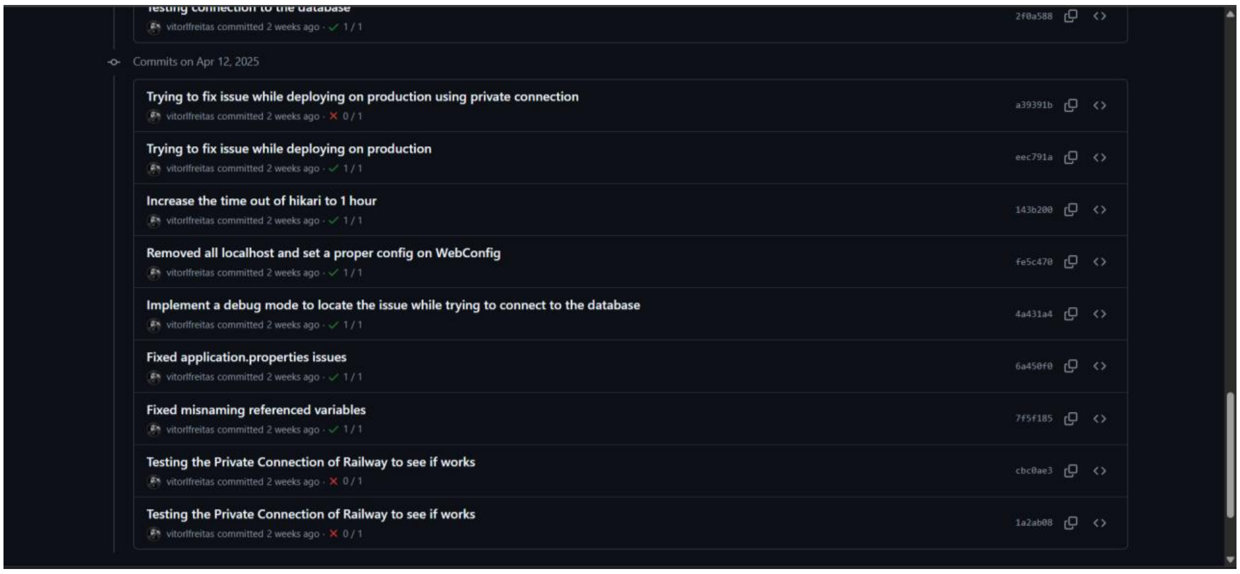
Add Dockerfile for Railway deployment
vitorlfreitas committed 2 weeks ago · ✗ 0 / 1 — b7951ab

Fix Mobile responsiveness bug and prepared the environment to be deployed
vitorlfreitas committed 2 weeks ago · ✗ 0 / 1 — c7053d4

Switched from H2 to MySQL with allowPublicKeyRetrieval=true, and fixed JDBC connection issue
vitorlfreitas committed 2 weeks ago — 7870d8e

Improve responsiveness for mobile application
vitorlfreitas committed 2 weeks ago — 5a2f152

Add a new feature to download the chat in pdf format



Testing connection to the database
vitorlfreitas committed 2 weeks ago · ✓ 1 / 1 — 2f0a588

Commits on Apr 12, 2025

Trying to fix issue while deploying on production using private connection
vitorlfreitas committed 2 weeks ago · ✗ 0 / 1 — a39391b

Trying to fix issue while deploying on production
vitorlfreitas committed 2 weeks ago · ✓ 1 / 1 — eec791a

Increase the time out of hikari to 1 hour
vitorlfreitas committed 2 weeks ago · ✓ 1 / 1 — 143b200

Removed all localhost and set a proper config on WebConfig
vitorlfreitas committed 2 weeks ago · ✓ 1 / 1 — fe5c470

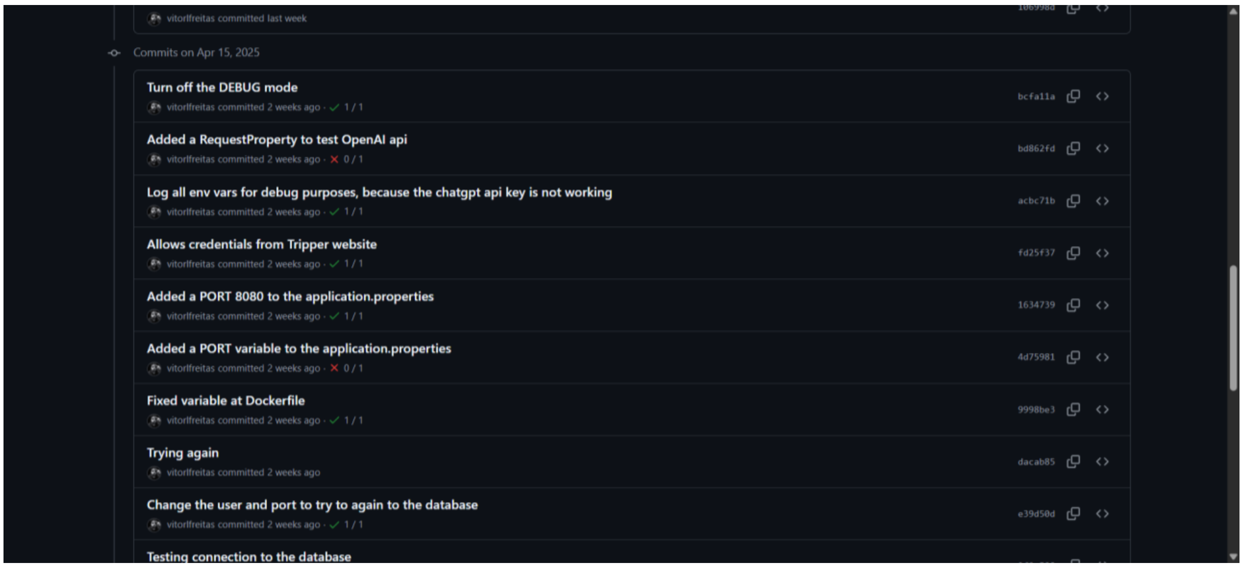
Implement a debug mode to locate the issue while trying to connect to the database
vitorlfreitas committed 2 weeks ago · ✓ 1 / 1 — 4a431a4

Fixed application.properties issues
vitorlfreitas committed 2 weeks ago · ✓ 1 / 1 — 6a450f0

Fixed misnaming referenced variables
vitorlfreitas committed 2 weeks ago · ✓ 1 / 1 — 7f5f185

Testing the Private Connection of Railway to see if works
vitorlfreitas committed 2 weeks ago · ✗ 0 / 1 — cbc0ae3

Testing the Private Connection of Railway to see if works
vitorlfreitas committed 2 weeks ago · ✗ 0 / 1 — 1a2ab08



vitorlfreitas committed last week — 106998d

Commits on Apr 15, 2025

Turn off the DEBUG mode
vitorlfreitas committed 2 weeks ago · ✓ 1 / 1 — bcfa11a

Added a RequestProperty to test OpenAI api
vitorlfreitas committed 2 weeks ago · ✗ 0 / 1 — bd862fd

Log all env vars for debug purposes, because the chatgpt api key is not working
vitorlfreitas committed 2 weeks ago · ✓ 1 / 1 — acbc71b

Allows credentials from Tripper website
vitorlfreitas committed 2 weeks ago · ✓ 1 / 1 — fd25f37

Added a PORT 8080 to the application.properties
vitorlfreitas committed 2 weeks ago · ✓ 1 / 1 — 1634739

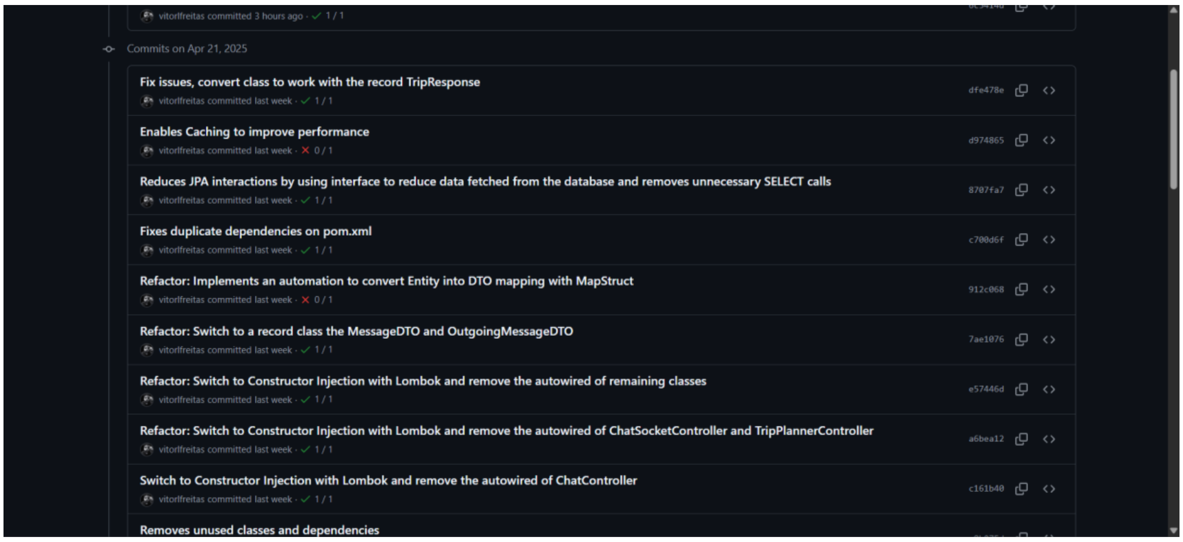
Added a PORT variable to the application.properties
vitorlfreitas committed 2 weeks ago · ✗ 0 / 1 — 4d75981

Fixed variable at Dockerfile
vitorlfreitas committed 2 weeks ago · ✓ 1 / 1 — 9998be3

Trying again
vitorlfreitas committed 2 weeks ago — dacab85

Change the user and port to try to again to the database
vitorlfreitas committed 2 weeks ago · ✓ 1 / 1 — e39d50d

Testing connection to the database


vitorlfreitas committed 3 hours ago · ✓ 1 / 1

Commits on Apr 21, 2025

Fix issues, convert class to work with the record TripResponse
vitorlfreitas committed last week · ✓ 1 / 1 dfe478e

Enables Caching to improve performance
vitorlfreitas committed last week · ✗ 0 / 1 d974865

Reduces JPA interactions by using interface to reduce data fetched from the database and removes unnecessary SELECT calls
vitorlfreitas committed last week · ✓ 1 / 1 8707fa7

Fixes duplicate dependencies on pom.xml
vitorlfreitas committed last week · ✓ 1 / 1 c700d6f

Refactor: Implements an automation to convert Entity into DTO mapping with MapStruct
vitorlfreitas committed last week · ✗ 0 / 1 912c068

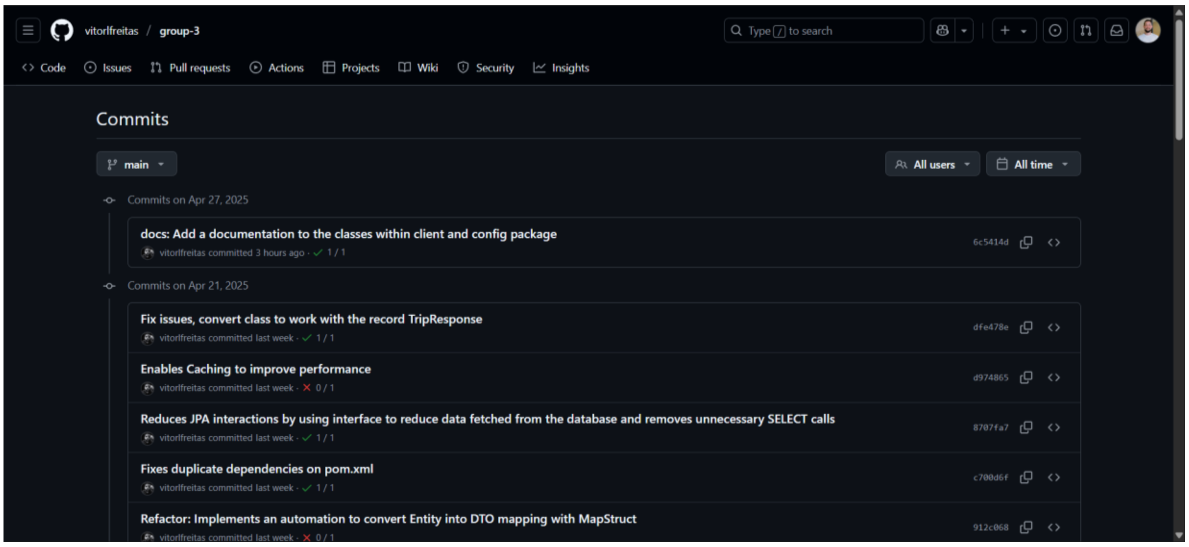
Refactor: Switch to a record class the MessageDTO and OutgoingMessageDTO
vitorlfreitas committed last week · ✓ 1 / 1 7ae1076

Refactor: Switch to Constructor Injection with Lombok and remove the autowired of remaining classes
vitorlfreitas committed last week · ✓ 1 / 1 e57446d

Refactor: Switch to Constructor Injection with Lombok and remove the autowired of ChatSocketController and TripPlannerController
vitorlfreitas committed last week · ✓ 1 / 1 a6bea12

Switch to Constructor Injection with Lombok and remove the autowired of ChatController
vitorlfreitas committed last week · ✓ 1 / 1 c161b40

Removes unused classes and dependencies


vitorlfreitas / group-3

Code Issues Pull requests Actions Projects Wiki Security Insights

Commits

main All users All time

Commits on Apr 27, 2025

docs: Add a documentation to the classes within client and config package
vitorlfreitas committed 3 hours ago · ✓ 1 / 1 6c5414d

Commits on Apr 21, 2025

Fix issues, convert class to work with the record TripResponse
vitorlfreitas committed last week · ✓ 1 / 1 dfe478e

Enables Caching to improve performance
vitorlfreitas committed last week · ✗ 0 / 1 d974865

Reduces JPA interactions by using interface to reduce data fetched from the database and removes unnecessary SELECT calls
vitorlfreitas committed last week · ✓ 1 / 1 8707fa7

Fixes duplicate dependencies on pom.xml
vitorlfreitas committed last week · ✓ 1 / 1 c700d6f

Refactor: Implements an automation to convert Entity into DTO mapping with MapStruct
vitorlfreitas committed last week · ✗ 0 / 1 912c068

10.5 Full Log Details

```
commit 1e3d80750d70c500b3bfb237496edc34973013b2 (HEAD -> main, origin/main, origin/HEAD)
Merge: 81a812b 36494b6
Author: vitorlfreitas <vitor.lucfreitas@gmail.com>
Date:   Sun Apr 27 20:19:54 2025 +0100

    Merge remote-tracking branch 'origin/main'

commit 81a812be083abe869fea4f7f218443926c9c1e11
Author: vitorlfreitas <vitor.lucfreitas@gmail.com>
Date:   Sun Apr 27 20:19:23 2025 +0100

    Refactors the code by removing unused code (added into the first versions of the code)

commit f1aeb501c4eb387c7ab8b19cb4866649a70aea26
Author: vitorlfreitas <vitor.lucfreitas@gmail.com>
Date:   Sun Apr 27 19:59:22 2025 +0100

    docs: Add documentatio to all classes within dto package

commit be5db93612de8bbc4e357ef5e363f462ac0eaa69
Author: vitorlfreitas <vitor.lucfreitas@gmail.com>
Date:   Sun Apr 27 19:53:49 2025 +0100
```

```
    docs: Adds a documentation to TripPlannerController.java

commit e1a4fdbc20e72a4ba85a44ce634415e86a1a820d
Author: vitorlfreitas <vitor.lucfreitas@gmail.com>
Date:   Sun Apr 27 19:51:27 2025 +0100

    docs: Adds a documentation to ChatSocketController

commit 6c5414d73ba0589fb7d8c4f7224e54d924f98a73
Author: vitorlfreitas <vitor.lucfreitas@gmail.com>
Date:   Sun Apr 27 19:45:23 2025 +0100

    docs: Add a documentation to the classes within client and config package



commit 8eddf306f8896595c713ec8558c426da1e12a832 (HEAD -> milestone03, origin/milestone03)
Author: Thales Campos <159160714+tlmcampos@users.noreply.github.com>
Date:   Sat Apr 26 23:56:26 2025 +0100

    Update groupC_FinalDocumentation_SD2.docx

commit 45aed69f8f848a0a2af14b39ad4dc374c96e5669
Author: Thales Campos <159160714+tlmcampos@users.noreply.github.com>
Date:   Sat Apr 26 23:22:11 2025 +0100

    Update groupC_FinalDocumentation_SD2.docx

commit 6a27ce8397665bc67c87d1f34bf61fc1ca9522a5
Author: Thales Campos <159160714+tlmcampos@users.noreply.github.com>
Date:   Sat Apr 26 23:00:03 2025 +0100

    Update groupC_FinalDocumentation_SD2.docx

commit 7ec78c56baa99d54abbd2bb304a091f3e30209a0
Merge: b9db3c7 40fb6e2
Author: Thales Campos <159160714+tlmcampos@users.noreply.github.com>
Date:   Sat Apr 26 22:32:12 2025 +0100
:...skipping...
commit 8eddf306f8896595c713ec8558c426da1e12a832 (HEAD -> milestone03, origin/milestone03)
Author: Thales Campos <159160714+tlmcampos@users.noreply.github.com>
Date:   Sat Apr 26 23:56:26 2025 +0100

    Update groupC_FinalDocumentation_SD2.docx

commit 45aed69f8f848a0a2af14b39ad4dc374c96e5669
Author: Thales Campos <159160714+tlmcampos@users.noreply.github.com>
Date:   Sat Apr 26 23:22:11 2025 +0100

    Update groupC_FinalDocumentation_SD2.docx

commit 6a27ce8397665bc67c87d1f34bf61fc1ca9522a5
Author: Thales Campos <159160714+tlmcampos@users.noreply.github.com>
Date:   Sat Apr 26 23:00:03 2025 +0100

    Update groupC_FinalDocumentation_SD2.docx

commit 7ec78c56baa99d54abbd2bb304a091f3e30209a0
Merge: b9db3c7 40fb6e2
Author: Thales Campos <159160714+tlmcampos@users.noreply.github.com>
Date:   Sat Apr 26 22:32:12 2025 +0100

    Merge branch 'milestone03' of https://github.com/vitorlfreitas/group-3 into milestone03

commit b9db3c7511b94fef49d29f446722493c503a0854
Author: Thales Campos <159160714+tlmcampos@users.noreply.github.com>
Date:   Sat Apr 26 22:32:06 2025 +0100

    Update groupC_FinalDocumentation_SD2.docx

commit 40fb6e23184cc6631e7383ecda91033b49dabc1f
Merge: d811c60 56d3744
Author: Karen F Magalhaes <karenfmagalhaes@gmail.com>
Date:   Sat Apr 26 22:06:40 2025 +0100

    Merge branch 'milestone03' of https://github.com/vitorlfreitas/group-3 into milestone03

commit d811c60bd86806382d6c3b95959064209f440df0
Author: Karen F Magalhaes <karenfmagalhaes@gmail.com>
```

```
Date:   Sat Apr 26 22:06:34 2025 +0100

    Update groupC_FinalProject_SD2.docx

    Add images

commit 56d3744108e771fc41ea7bbd017d7bd177174dae
Merge: f505e1f 269335c
Author: Thales Campos <159160714+tlmcampos@users.noreply.github.com>
Date:   Sat Apr 26 21:59:49 2025 +0100

    Merge branch 'milestone03' of https://github.com/vitorlfreitas/group-3 into milestone03

commit f505e1f99d1385a84495a4fc9e1124cd93c2a1a1
Author: Thales Campos <159160714+tlmcampos@users.noreply.github.com>
Date:   Sat Apr 26 21:59:24 2025 +0100

    Update groupC_FinalDocumentation_SD2.docx

commit 269335ced52d9fd75361d2a2b9b9d49b9c7ef1d5
Merge: 48a40c3 8951424
Author: Karen F Magalhaes <karenfmagalhaes@gmail.com>
Date:   Sat Apr 26 21:32:37 2025 +0100

    Merge branch 'milestone03' of https://github.com/vitorlfreitas/group-3 into milestone03

commit 48a40c37e80ef7a37e42d7105a4156d2cb7318e2
Author: Karen F Magalhaes <karenfmagalhaes@gmail.com>
Date:   Sat Apr 26 21:32:14 2025 +0100

    Update groupC_FinalProject_SD2.docx

commit 8951424dad24679e42265531d7162e446ddf0ded
Author: Thales Campos <159160714+tlmcampos@users.noreply.github.com>
Date:   Sat Apr 26 21:32:12 2025 +0100

    Create groupC_FinalDocumentation_SD2.docx

commit 3b60d2d5be06bcfa1ca2e394c5ff65ef0e132fcc
Author: Thales Campos <159160714+tlmcampos@users.noreply.github.com>
Date:   Sat Apr 26 21:07:06 2025 +0100

    Update Final Documentation

commit 895c816c2633462b693c5b780e8c2b2876c29c08
Author: Thales Campos <159160714+tlmcampos@users.noreply.github.com>
Date:   Sat Apr 26 20:40:51 2025 +0100

    Update Tripper Logo

commit 36494b6f644c14e663d5fa70199a252eacdbd3b9
Author: Karen F Magalhaes <karenfmagalhaes@gmail.com>
Date:   Tue Apr 22 19:03:21 2025 +0100

    Update milestone 3

commit a344d415ad2ca73af1a269378b678b83a547973a
Author: Karen F Magalhaes <karenfmagalhaes@gmail.com>
Date:   Tue Apr 22 18:10:10 2025 +0100

    Delete ~$oupC_FinalProject_SD2.docx

commit f0bbd91e894c70e50c17c9b49b0cbf76cd736a84
Author: Karen F Magalhaes <karenfmagalhaes@gmail.com>
Date:   Tue Apr 22 18:09:51 2025 +0100

    Update documentation

commit a51e6ab3203d16814d2a8ddec51c4a34b64bda50
Author: Karen F Magalhaes <karenfmagalhaes@gmail.com>
Date:   Tue Apr 22 18:06:01 2025 +0100

    Update Gantt chart.xlsx

commit dfe478e8b1dba75740f86ee3130d84d369ff7355
Author: vitorlfreitas <vitor.lucfreitas@gmail.com>
Date:   Mon Apr 21 12:37:58 2025 +0100

    Fix issues, convert class to work with the record TripResponse
```

```
commit d974865ed0c7972ca38bb4cb6e2dfbb12e90010f
Author: vitorlfreitas <vitor.lucfreitas@gmail.com>
Date:   Mon Apr 21 12:12:21 2025 +0100

    Enables Caching to improve performance

commit 8707fa7703c948eef73a7b3f8e6fa4610ef6fda5
Author: vitorlfreitas <vitor.lucfreitas@gmail.com>
Date:   Mon Apr 21 11:49:53 2025 +0100

    Reduces JPA interactions by using interface to reduce data fetched from the database and
removes unnecessary SELECT calls

commit c700d6f0ead45662710300e0447326421c7c2948
Author: vitorlfreitas <vitor.lucfreitas@gmail.com>
Date:   Mon Apr 21 11:14:47 2025 +0100

    Fixes duplicate dependencies on pom.xml

commit 912c0683bdfe3f4fe260c233fa209c41dd3418ea
Author: vitorlfreitas <vitor.lucfreitas@gmail.com>
Date:   Mon Apr 21 11:03:44 2025 +0100

    Refactor: Implements an automation to convert Entity into DTO mapping with MapStruct

commit 7ae10763682ba3ffa051d4b28c7bb2f1d027b89e
Author: vitorlfreitas <vitor.lucfreitas@gmail.com>
Date:   Mon Apr 21 02:30:27 2025 +0100

    Refactor: Switch to a record class the MessageDTO and OutgoingMessageDTO

commit e57446de501318065add0088f8a701817e6cbc93
Author: vitorlfreitas <vitor.lucfreitas@gmail.com>
Date:   Mon Apr 21 02:14:11 2025 +0100

    Refactor: Switch to Constructor Injection with Lombok and remove the autowired of remaining
classes

commit a6bea12c565af4c9f8772fdd3ad56e85f70001d5
Author: vitorlfreitas <vitor.lucfreitas@gmail.com>
Date:   Mon Apr 21 02:04:29 2025 +0100

    Refactor: Switch to Constructor Injection with Lombok and remove the autowired of
ChatSocketController and TripPlannerController

commit c161b40d9ebf2384bbd45a7f93346b18e0e4c1e9
Author: vitorlfreitas <vitor.lucfreitas@gmail.com>
Date:   Mon Apr 21 01:58:21 2025 +0100

    Switch to Constructor Injection with Lombok and remove the autowired of ChatController

commit a8b075dac6a1e5e76b23ab8f2e2a0ce88449b697
Author: vitorlfreitas <vitor.lucfreitas@gmail.com>
Date:   Mon Apr 21 01:50:07 2025 +0100

    Removes unused classes and dependencies

commit 2e77a35778de3bbb93ba14bd2b3e3c14f6876215
Author: vitorlfreitas <vitor.lucfreitas@gmail.com>
Date:   Mon Apr 21 01:27:47 2025 +0100

    Refactors the ChatController removing boilerplate code

commit 6651a17e7a8bb1cf75e5cfa5db2758b720bf07bb
Author: vitorlfreitas <vitor.lucfreitas@gmail.com>
Date:   Mon Apr 21 00:57:26 2025 +0100

    Add comments to the classes within Config package

commit 2ef0ccf991163157ee7b0a7ab334f15f1daf083e
Author: vitorlfreitas <vitor.lucfreitas@gmail.com>
Date:   Sun Apr 20 05:09:19 2025 +0100

    Fix NLPInputParser class as a bean component

commit a8e31e9b8d826b6b81cb733e977b88c7463b6064
Author: vitorlfreitas <vitor.lucfreitas@gmail.com>
Date:   Sun Apr 20 04:39:46 2025 +0100

    Adds a new config class to manage the weatherservice
```

```
commit 106998d86a3a65139a2ac783bf05c81d93949cc9
Author: vitorlfreitas <vitor.lucfreitas@gmail.com>
Date:   Sun Apr 20 04:38:20 2025 +0100

    Refactor the WeatherService and add comments to the client classes

commit bcfa11ab236e41302768e2f9c4ebcb7f7f0264b7
Author: vitorlfreitas <vitor.lucfreitas@gmail.com>
Date:   Tue Apr 15 06:02:27 2025 +0100

    Turn off the DEBUG mode

commit bd862fda77ceca591549bcc44fb52d233a6e7724
Author: vitorlfreitas <vitor.lucfreitas@gmail.com>
Date:   Tue Apr 15 05:52:49 2025 +0100

    Added a RequestProperty to test OpenAI api

commit acbc71b06995aceb0972e50848d4ab82f957acf9
Author: vitorlfreitas <vitor.lucfreitas@gmail.com>
Date:   Tue Apr 15 05:18:07 2025 +0100

    Log all env vars for debug purposes, because the chatgpt api key is not working

commit fd25f37ae4d4feb4f9f52d059da11aafeb133244
Author: vitorlfreitas <vitor.lucfreitas@gmail.com>
Date:   Tue Apr 15 04:42:44 2025 +0100

    Allows credentials from Tripper website

commit 1634739db15679b13d0efbc42bc892f903526b88
Author: vitorlfreitas <vitor.lucfreitas@gmail.com>
Date:   Tue Apr 15 04:37:32 2025 +0100

    Added a PORT 8080 to the application.properties

commit 4d7598181607a0648155293cb700261c5e91dab7
Author: vitorlfreitas <vitor.lucfreitas@gmail.com>
Date:   Tue Apr 15 04:25:29 2025 +0100

    Added a PORT variable to the application.properties

commit 9998be3eecdaba6245970e99ea9c9a3ea39ab5e8
Author: vitorlfreitas <vitor.lucfreitas@gmail.com>
Date:   Tue Apr 15 04:21:28 2025 +0100

    Fixed variable at Dockerfile

commit dacab85f8d9f294a22280179899e381a61fcf695
Author: vitorlfreitas <vitor.lucfreitas@gmail.com>
Date:   Tue Apr 15 03:39:57 2025 +0100

    Trying again

commit e39d50dc0aea1c18e7560a1dbac34de5cfe3f5d4
Author: vitorlfreitas <vitor.lucfreitas@gmail.com>
Date:   Tue Apr 15 03:23:26 2025 +0100

    Change the user and port to try to again to the database

commit 2f0a5882198bd22c2b4a8ced379fab1cacecb5c7
Author: vitorlfreitas <vitor.lucfreitas@gmail.com>
Date:   Tue Apr 15 03:18:29 2025 +0100

    Testing connection to the database

commit a39391bb2da36562d44f9ec6a88fc189e3c3e914
Author: vitorlfreitas <vitor.lucfreitas@gmail.com>
Date:   Sat Apr 12 06:50:35 2025 +0100

    Trying to fix issue while deploying on production using private connection

commit eec791a3a998bd62aeb9c453958a280fc1d3092e
Author: vitorlfreitas <vitor.lucfreitas@gmail.com>
Date:   Sat Apr 12 06:45:43 2025 +0100

    Trying to fix issue while deploying on production

commit 143b200af1eb2f0e975bd22b78fe158b2e6f2642
```

```
Author: vitorlfreitas <vitor.lucfreitas@gmail.com>
Date:   Sat Apr 12 06:11:06 2025 +0100

    Increase the time out of hikari to 1 hour

commit fe5c470605c896f41b9f86097a6237edb9b6ad51
Author: vitorlfreitas <vitor.lucfreitas@gmail.com>
Date:   Sat Apr 12 06:07:05 2025 +0100

    Removed all localhost and set a proper config on WebConfig

commit 4a431a499f97f2bf46dce58c17d60f22760c4971
Author: vitorlfreitas <vitor.lucfreitas@gmail.com>
Date:   Sat Apr 12 05:53:50 2025 +0100

    Implement a debug mode to locate the issue while trying to connect to the database

commit 6a450f0e89392e7b8c91c740e243f97117191e29
Author: vitorlfreitas <vitor.lucfreitas@gmail.com>
Date:   Sat Apr 12 05:30:56 2025 +0100

    Fixed application.properties issues

commit 7f5f185e1c3792aa5c7c74fee19d5fc3adac7afc
Author: vitorlfreitas <vitor.lucfreitas@gmail.com>
Date:   Sat Apr 12 05:21:24 2025 +0100

    Fixed misnaming referenced variables

commit cbc0ae37b00311681883d73e1178ca6b196b1d57
Author: vitorlfreitas <vitor.lucfreitas@gmail.com>
Date:   Sat Apr 12 05:06:54 2025 +0100

    Testing the Private Connection of Railway to see if works

commit 1a2ab0886feaa4c0770106ab7b539c986bd1c544
Author: vitorlfreitas <vitor.lucfreitas@gmail.com>
Date:   Sat Apr 12 04:40:10 2025 +0100

    Testing the Private Connection of Railway to see if works

commit 947c170f7d71b3f3c11af0158506c887a1f4989e
Author: vitorlfreitas <vitor.lucfreitas@gmail.com>
Date:   Sat Apr 12 04:28:23 2025 +0100

    Update the application.properties to connect to the public database

commit 5afa1faef89fb6e3eebc76e35f69e14c88669e0d
Author: vitorlfreitas <vitor.lucfreitas@gmail.com>
Date:   Sat Apr 12 03:34:32 2025 +0100

    Set properties variables to work with mysql from Railway

commit 11e90118b9785c09e2e643a8b28a05944ca43fe8
Author: vitorlfreitas <vitor.lucfreitas@gmail.com>
Date:   Fri Apr 11 09:42:57 2025 +0100

    Set active Spring profile to prod in Dockerfile

commit d52c6219bc5e26cdf563e0ed260623a5f607c2e4
Author: vitorlfreitas <vitor.lucfreitas@gmail.com>
Date:   Fri Apr 11 08:46:49 2025 +0100

    Deleted folder : moved to a separated repository

commit b7951ab8e29b43caf270de80f6bc4ac116dcb8aa
Author: vitorlfreitas <vitor.lucfreitas@gmail.com>
Date:   Fri Apr 11 07:55:21 2025 +0100

    Add Dockerfile for Railway deployment

commit c7053d451de446b2aa1b5289ed8cec31f961df74
Author: vitorlfreitas <vitor.lucfreitas@gmail.com>
Date:   Fri Apr 11 07:44:57 2025 +0100

    Fix Mobile responsiveness bug and prepared the environment to be deployed

commit 7870d8e973f5c47db438bee69540c63264649d67
Author: vitorlfreitas <vitor.lucfreitas@gmail.com>
Date:   Fri Apr 11 06:25:16 2025 +0100
```

```
    Switched from H2 to MySQL with allowPublicKeyRetrieval=true, and fixed JDBC connection issue

commit 5a2f15271cb692db54f455f6eba39968355bc886
Author: vitorlfreitas <vitor.lucfreitas@gmail.com>
Date:   Fri Apr 11 05:48:33 2025 +0100

    Improve responsiveness for mobile application

commit a31e0dcd819f629ca8c5e4742cfb8cfe86b3bdf8
Author: vitorlfreitas <vitor.lucfreitas@gmail.com>
Date:   Fri Apr 11 04:44:16 2025 +0100

    Add a new feature to download the chat in pdf format

commit 548ac9f8b4b1191905d252a3f5fe935ab9806493
Author: vitorlfreitas <vitor.lucfreitas@gmail.com>
Date:   Fri Apr 11 04:43:21 2025 +0100

    Add a new feature to download the conversation in pdf format

commit 80c0e95a29056bf7a04a4beab82e9cba567892b7
Author: vitorlfreitas <vitor.lucfreitas@gmail.com>
Date:   Fri Apr 11 02:42:46 2025 +0100

    Add a new feature to delete conversations from history

commit b732bc95b3374d23c25de7d012c91d8da2b7d67c
Author: vitorlfreitas <vitor.lucfreitas@gmail.com>
Date:   Fri Apr 11 02:29:43 2025 +0100

    Add a new feature to edit the title of the messages and saves to the backend

commit 906810e10483bfc1d12b3be2edd64bc8933f79fb
Author: vitorlfreitas <vitor.lucfreitas@gmail.com>
Date:   Fri Apr 11 02:18:58 2025 +0100

    Add persistent chat history, conversation sidebar, and message serialization fix

commit 3e5bc390556f44bd590e3bb0485ba6d473220340
Author: vitorlfreitas <vitor.lucfreitas@gmail.com>
Date:   Fri Apr 11 00:48:58 2025 +0100

    Add a new page /chat and the index of the application, and implements the authentication (at
the moment only by Google) to be able to use the chatbot

commit 1e82f5fbba4753ca393b918ca57328eb93823187
Author: vitorlfreitas <vitor.lucfreitas@gmail.com>
Date:   Wed Apr 9 05:03:07 2025 +0100

    Finalizes and polish the conversation, it includes weather conditions if found using the
Weather API and improves the generated answer of the Tripper

commit 4e8a6c6135968d2bd18fbe656b52a04b6f7e4bf3
Author: vitorlfreitas <vitor.lucfreitas@gmail.com>
Date:   Tue Apr 8 22:36:30 2025 +0100

    Enables WebSocket communication

commit 75c2463039275a8e523ea4f8590c5dcf8bcfb54d
Author: vitorlfreitas <vitor.lucfreitas@gmail.com>
Date:   Tue Apr 8 22:24:17 2025 +0100

    Converts the backend to handle stateful conversations

commit e29de3245b2425ce9eeea6309fe119a7102a7971
Author: vitorlfreitas <vitor.lucfreitas@gmail.com>
Date:   Tue Apr 8 21:54:02 2025 +0100

    Implements basic models, Conversation and Message, and Repositories for both. Besides a
service to manage chat state

commit c3defd5118b8580cc1b551d8b0cc6673e253bd70
Author: vitorlfreitas <vitor.lucfreitas@gmail.com>
Date:   Tue Apr 8 21:16:35 2025 +0100

    Implement the first step to Web applications where focused to create a connection between the
Spring boot application and Next.js. (I have never done this before, so I am using chatgpt to help
me through th
is process)
```

```
commit 7556b984172403940edfe63cdac8255a29671bf7
Author: vitorlfreitas <vitor.lucfreitas@gmail.com>
Date:   Mon Apr 7 04:56:01 2025 +0100

    Refactors and Improve the ability to generate PDF

commit 6e73ff150d4f9d561ef6a4303414425218b64ef9
Author: vitorlfreitas <vitor.lucfreitas@gmail.com>
Date:   Mon Apr 7 01:13:20 2025 +0100

    Improves the GPT contextual travel advised powered by OpenNLP AND built a /nlp/trip-
recommendations REST endpoint

commit 7c2680c8eb31d4be015a0af3e2571d4fe5afa8e3
Author: vitorlfreitas <vitor.lucfreitas@gmail.com>
Date:   Mon Apr 7 01:02:43 2025 +0100

    Implements the TripInfoExtractionService which extracts locations and dates from natural
language

commit 92aea1ea500eaafe06eb74c6537d28d181dbb22f
Author: vitorlfreitas <vitor.lucfreitas@gmail.com>
Date:   Mon Apr 7 00:38:51 2025 +0100

    feat: Convert project into Spring Boot API and implement NLP endpoints

    - Reorganized project structure into Spring Boot application
    - Added Maven support and configured pom.xml with required dependencies (OpenNLP, Gson,
PDFBox, etc.)
    - Integrated sentence detection model (OpenNLP) and placed model file in resources
    - Implemented REST controller with /nlp/tokenize and /nlp/sentences endpoints
    - Created NLP services using OpenNLP for tokenization and sentence detection
    - Configured embedded Tomcat and enabled automatic startup on port 8080
    - Removed duplicate SLF4J bindings to fix logging warnings
    - Verified application startup, logs, and endpoint responses successfully

commit 9aa049e6edf359c8e45d1d3a353063553322443c
Author: vitorlfreitas <vitor.lucfreitas@gmail.com>
Date:   Mon Apr 7 00:38:07 2025 +0100

    feat: Convert project into Spring Boot API and implement NLP endpoints

    - Reorganized project structure into Spring Boot application
    - Added Maven support and configured pom.xml with required dependencies (OpenNLP, Gson,
PDFBox, etc.)
    - Integrated sentence detection model (OpenNLP) and placed model file in resources
    - Implemented REST controller with /nlp/tokenize and /nlp/sentences endpoints
    - Created NLP services using OpenNLP for tokenization and sentence detection
    - Configured embedded Tomcat and enabled automatic startup on port 8080
    - Removed duplicate SLF4J bindings to fix logging warnings
    - Verified application startup, logs, and endpoint responses successfully
```

11. Bibliography

Apache pdfbox® - A java PDF library (no date) *Apache PDFBox | A Java PDF Library*. Available at: https://pdfbox.apache.org/ (Accessed: 18 March 2025).

Com.google.gson module summary - GSON 2.12.1 javadoc. Available at: https://javadoc.io/doc/com.google.code.gson/gson/latest/com.google.gson/module-summary.html (Accessed: 18 March 2025).

Googlemaps. *GitHub - googlemaps/google-maps-services-java: Java client library for Google Maps API Web Services*. Available at: https://github.com/googlemaps/google-maps-services-java. (Accessed: 23 April 2025).

GSON user guide gson. Available at: https://google.github.io/gson/UserGuide.html (Accessed: 18 March 2025).

IText 5.5.13.3 API. Available at: https://api.itextpdf.com/iText5/java/5.5.13.3/. (Accessed: 22 April 2025).

OpenAI's GPT-3.5-turbo model. Available at: https://platform.openai.com/docs/models/gpt-3-5-turbo (Accessed: 22 March 2025).

Package java.net (2024) *java.net (Java Platform SE 8)*. Available at: https://docs.oracle.com/javase/8/docs/api/java/net/package-summary.html (Accessed: 18 March 2025).

Project Lombok. Available at: https://projectlombok.org/. (Accessed: 22 April 2025).

Shevat, A., 2017. *Designing bots: Creating conversational experiences*. " O'Reilly Media, Inc.".

Spring Framework. Available at: https://spring.io/projects/spring-framework. (Accessed: 22 April 2025).

Tokenizermodel (apache opennlp tools 2.1.1 API). Available at: https://opennlp.apache.org/docs/2.1.1/apidocs/opennlp-tools/opennlp/tools/tokenize/TokenizerModel.html (Accessed: 18 March 2025).