Project documentation

Introduction - project goal

The aim of the project is to design an information system that performs statistical analysis and data computations sourced from the NBP API platform. This includes determining the number of upward, downward, and unchanged sessions, as well as calculating statistical measures such as median, mode, standard deviation, and coefficient of variation. Additionally, the system generates monthly and quarterly distribution of changes. All these functionalities are computed and displayed for a user-selected time period.

Project team

5D:

- Jakub Matuszak, 240749
- Kacper Michalec, 240751
- Piotr Jurek, 240693
- Weronika Kretowicz, 235903
- Mateusz Magdziński, 240742

System architecture description

The system architecture for this project is based on a modern web development stack utilizing React with several key libraries and technologies. The project leverages TypeScript for type safety, Vite for fast builds, Axios for HTTP requests, Chart.js for data visualization, TailwindCSS for styling, and GH-Pages for deployment. Vitest for Unit testing.

Component Descriptions

ExchangeRateProvider.tsx

- **Purpose:** Provides exchange rate data to other components within the application.
- **Functionality:** Fetches exchange rates using Axios, stores the data, and makes it accessible via context.

SessionStatisticsProvider.tsx

- **Purpose:** Manages and provides session statistics to the application.
- **Functionality:** Aggregates data on user sessions, stores it in a context, and provides it to consuming components.

StatisticsProvider.tsx

- **Purpose:** Supplies general statistical data to the application.
- **Functionality:** Fetches, processes, and stores statistical data, making it available through context.

CurrencyProvider.tsx

- **Purpose:** Manages currency-related data and operations.
- **Functionality:** Provides current exchange rates and currency conversion functionalities via context.

dateUtils.ts

- **Purpose:** Utility functions for date manipulations.
- **Functionality:** Provides helper functions for formatting and manipulating dates.

SessionAnalysis.tsx

- **Purpose:** Analyzes user session data and provides insights.
- **Functionality:** Processes session data to extract meaningful statistics and trends, visualized using Chart.js.

ChangeDistribution.tsx

- **Purpose:** Displays distribution of changes over a period.
- **Functionality:** Visualizes change data using Chart.js to show distribution patterns.

Sidebar.tsx

- **Purpose:** Provides navigation and contextual information.
- **Functionality:** Implements a sidebar menu for navigating different sections of the application.

StatisticalMeasures.tsx

- **Purpose:** Displays key statistical measures.
- **Functionality:** Calculates and displays various statistical measures such as mean, median, mode, etc.

Dashboard.tsx

- **Purpose:** Main dashboard displaying an overview of data and statistics.
- **Functionality:** Aggregates and displays key information using various child components and Chart.js for visualization.

Used technologies

- **React:** Core library for building user interfaces.
- **Vite with Vitest:** Fast build tool and testing framework.
- **Axios:** Library for making HTTP requests.
- **Chart.js:** Library for creating charts and visualizations.
- **TailwindCSS:** Utility-first CSS framework for styling.
- **GH-Pages:** Service for deploying static sites.
- **TypeScript:** Superset of JavaScript for type safety and enhanced developer experience.

CI/CD System

The CI/CD system uses GitHub Actions for automated testing and deployment. This ensures code reliability and quality through continuous integration and continuous deployment.

Build Workflow

- **Purpose:** Ensures new changes do not break the build. Triggered by pull requests to develop, release, and main branches.
- **Steps:**
  1. **Checkout Code:** Retrieves the latest code from the repository.
  2. **Setup Node.js:** Configures the Node.js environment.
  3. **Install Dependencies:** Uses Yarn to install project dependencies.
  4. **Build Project:** Compiles the project.
  5. **Run Tests:** Executes tests to verify the code.

Deployment Workflow

- **Purpose:** Automates deployment to GitHub Pages when changes are pushed to the release branch.
- **Steps:**
  1. **Checkout Code:** Retrieves the latest code from the release branch.
  2. **Setup Node.js:** Configures the Node.js environment.

3. **Install Dependencies:** Uses Yarn to install project dependencies.
4. **Build Project:** Compiles the project.
5. **Deploy to GitHub Pages:** Deploys the project using Yarn.
6. **Manage Tags and Releases:** Fetches tags, calculates new tag, and creates a GitHub release.

Trigger Events

- **Build Workflow:** Triggered by pull requests to develop, release, and main branches.
- **Deployment Workflow:** Triggered by pushes to the release branch.

CI/CD Process

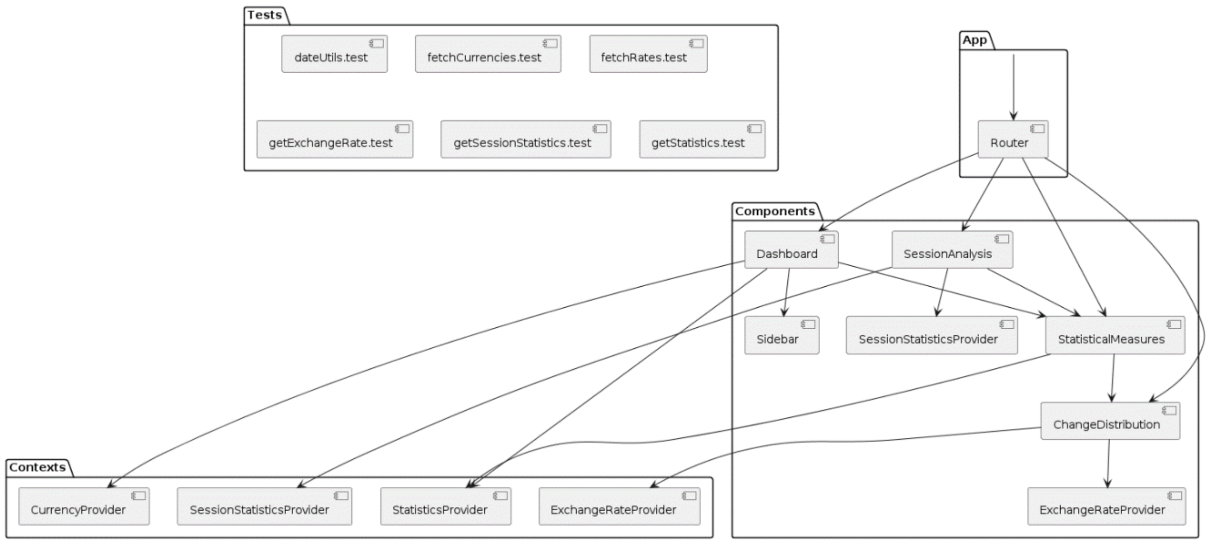
1. **Continuous Integration (CI):** Validates changes via the Build Workflow to ensure code integrity before merging.
2. **Continuous Deployment (CD):** Deploys updated code to GitHub Pages via the Deployment Workflow, managing version tags and releases.

CI Reports

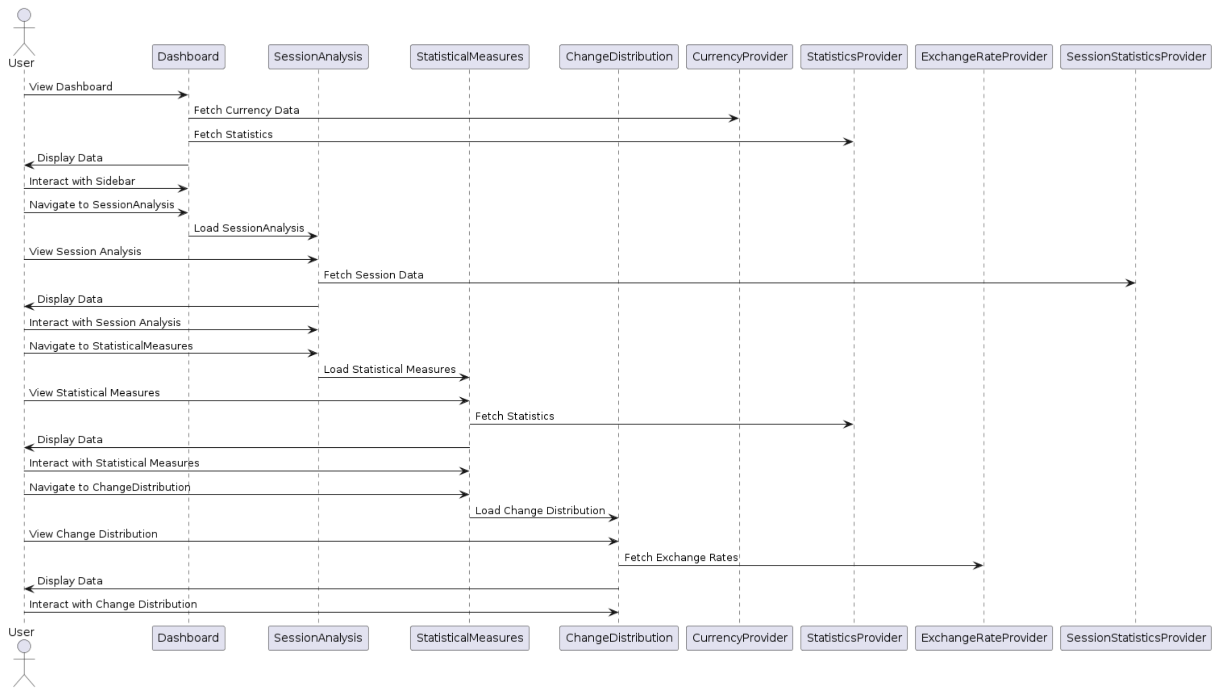
Access CI reports and workflow details on the GitHub Actions page. This provides a comprehensive view of the CI/CD activities and statuses.

For further assistance, refer to the workflow files in the .github/workflows directory or contact the project maintainers.

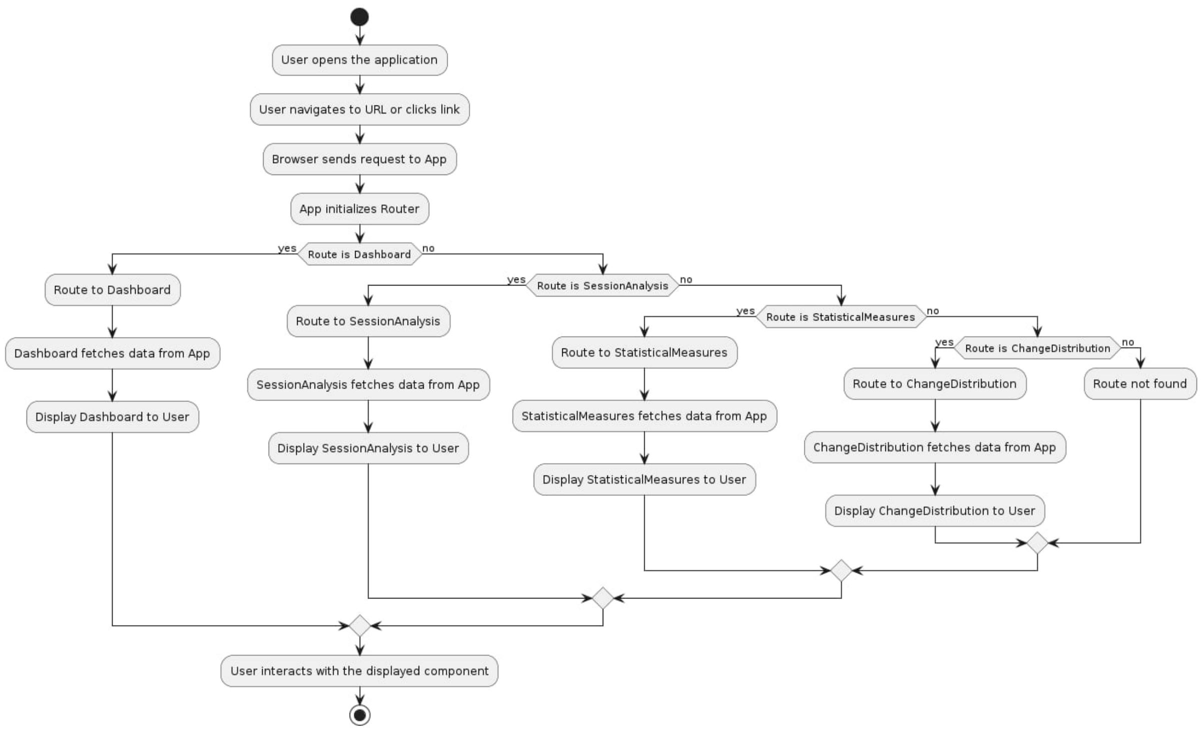
Component diagram



Sequence diagram



Activity diagram


User opens the application
User navigates to URL or clicks link
Browser sends request to App
App initializes Router
Route is Dashboard
yes
no
Route to Dashboard
Dashboard fetches data from App
Display Dashboard to User
Route is SessionAnalysis
yes
no
Route to SessionAnalysis
SessionAnalysis fetches data from App
Display SessionAnalysis to User
Route is StatisticalMeasures
yes
no
Route to StatisticalMeasures
StatisticalMeasures fetches data from App
Display StatisticalMeasures to User
Route is ChangeDistribution
yes
no
Route to ChangeDistribution
ChangeDistribution fetches data from App
Display ChangeDistribution to User
Route not found
User interacts with the displayed component