Sprint 3 Submission

CS5500 Project

Group Name: Fire Emblem

Group Members: Xinyi Zhang, Zeyu Shen

***GitHub Project Board*: https://github.com/users/xinyisherryz/projects/1**

***Git Repo*: https://github.com/xinyisherryz/cs5500-fire-emblem**

***Trello Board*: https://trello.com/b/Rk1GKh9W/cs5500-fire-emblem**

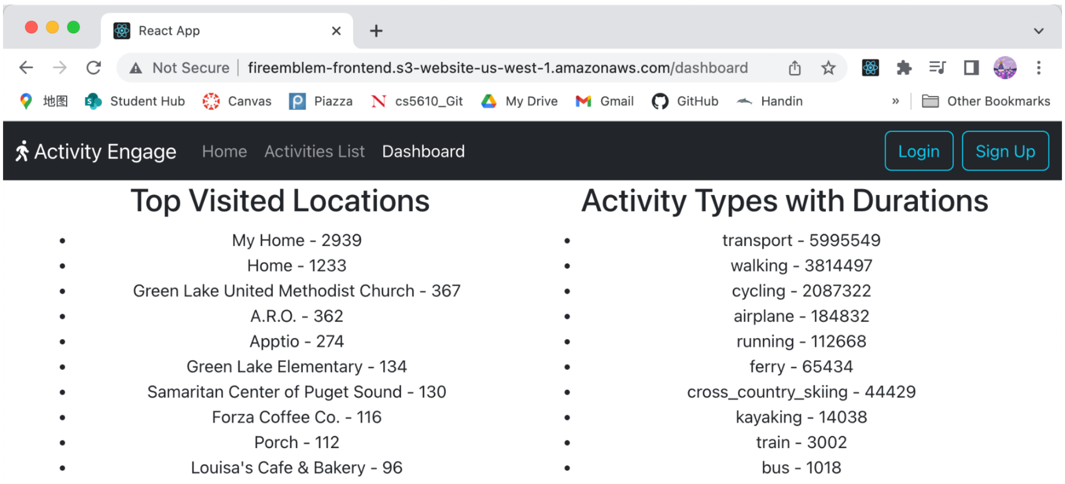
*** Summary of User Stories Delivered:**

1. Designed an engaging landing page, showcasing an overview of the website's features and enticing users to explore further.
2. Developed a comprehensive activities page, displaying a complete list of activities along with key metrics for easy tracking.
3. Enabled users to view activity details for a specific day, allowing for in-depth analysis of daily performance.
4. Implemented a dashboard page, presenting a detailed breakdown and analytics of user activities for a holistic view.
5. Integrated user authentication, ensuring a secure and personalized user experience.
6. Deployed the application on the cloud, providing users with seamless and convenient access to their data.
7. Installed CI pipeline for Java Maven on Github.

*** Functionality of Deliverable:**

For Sprint 3, we have delivered a React-based frontend, user authentication using JWT, and set up CORS and Security configurations. We have also prepared the application for deployment to AWS S3 and AWS Elastic Beanstalk.

1. The user interface of the web application has been constructed using React, which prominently features a user dashboard and various other components for an interactive experience.


React App
Not Secure | fireemblem-frontend.s3-website-us-west-1.amazonaws.com/dashboard
地图 Student Hub Canvas Piazza cs5610_Git My Drive Gmail GitHub Handin
Other Bookmarks
Activity Engage
Home
Activities List
Dashboard
Login
Sign Up

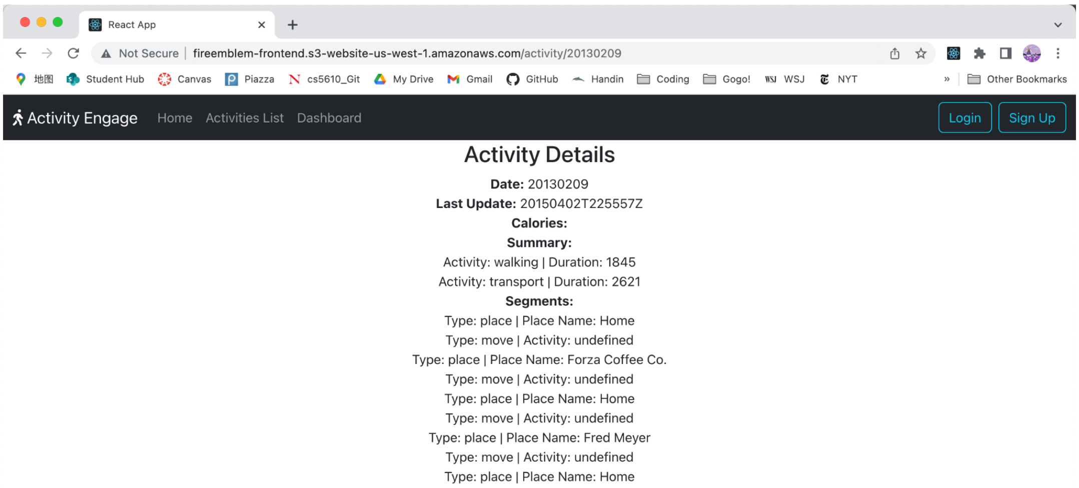
Top Visited Locations

- My Home - 2939
- Home - 1233
- Green Lake United Methodist Church - 367
- A.R.O. - 362
- Apptio - 274
- Green Lake Elementary - 134
- Samaritan Center of Puget Sound - 130
- Forza Coffee Co. - 116
- Porch - 112
- Louisa's Cafe & Bakery - 96

Activity Types with Durations

- transport - 5995549
- walking - 3814497
- cycling - 2087322
- airplane - 184832
- running - 112668
- ferry - 65434
- cross_country_skiing - 44429
- kayaking - 14038
- train - 3002
- bus - 1018

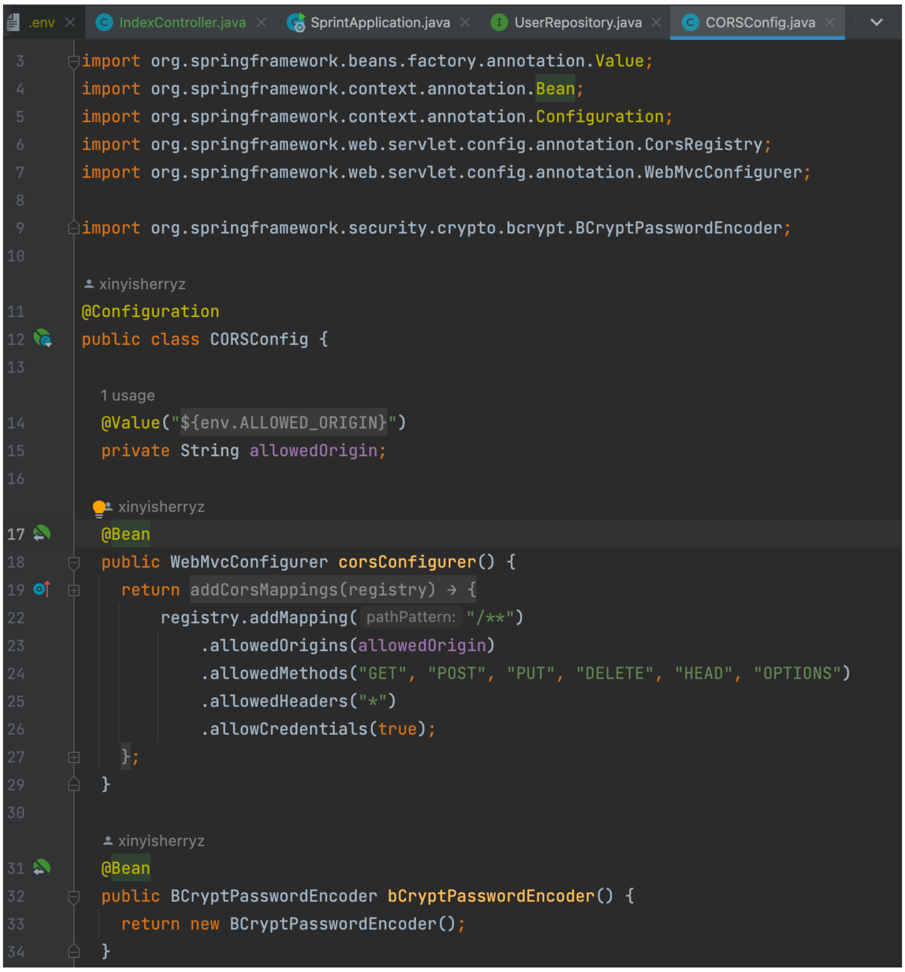
2. The application presents lists and detailed views of data points, providing users with interaction options such as sorting and viewing individual activities.


React App
Not Secure | fireemblem-frontend.s3-website-us-west-1.amazonaws.com/activity/20130209
地图 Student Hub Canvas Piazza cs5610_Git My Drive Gmail GitHub Handin Coding Gogo! WSJ NYT
Other Bookmarks
Activity Engage
Home
Activities List
Dashboard
Login
Sign Up

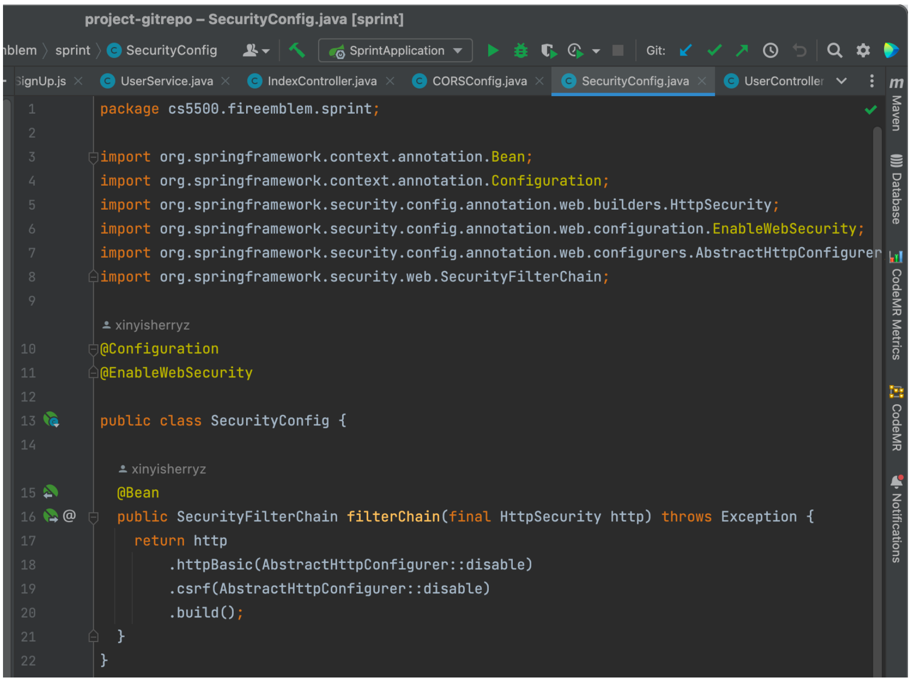
Activity Details

Date: 20130209
Last Update: 20150402T225557Z
Calories:
Summary:
Activity: walking | Duration: 1845
Activity: transport | Duration: 2621
Segments:
Type: place | Place Name: Home
Type: move | Activity: undefined
Type: place | Place Name: Forza Coffee Co.
Type: move | Activity: undefined
Type: place | Place Name: Home
Type: move | Activity: undefined
Type: place | Place Name: Fred Meyer
Type: move | Activity: undefined
Type: place | Place Name: Home

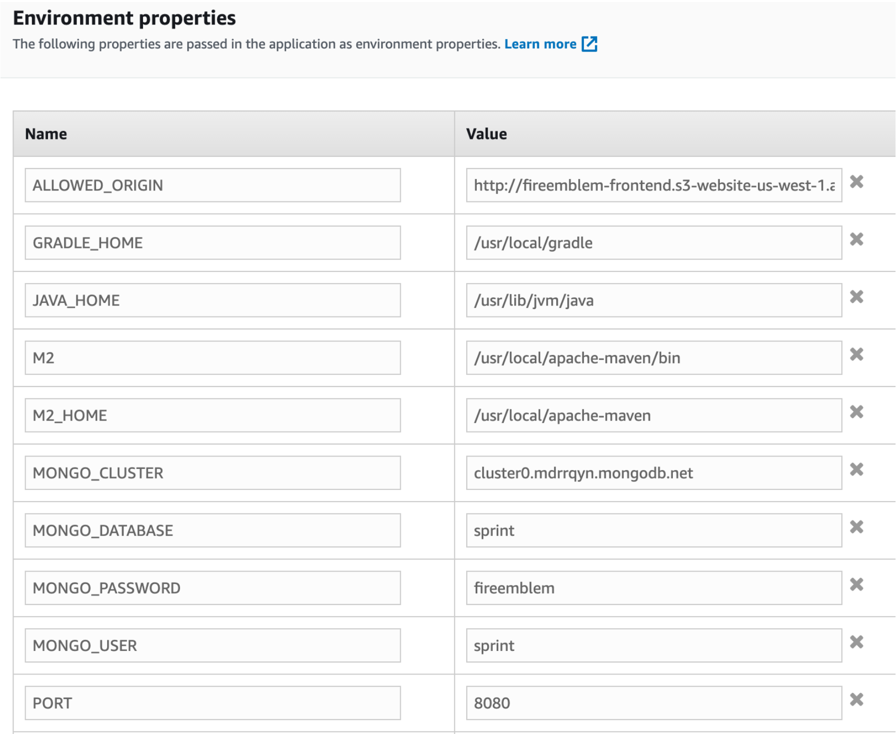
3. By implementing CORS configuration and SecurityConfig, access to the API from the frontend is controlled and restricted to only permitted origins.



```java
import org.springframework.beans.factory.annotation.Value;
import org.springframework.context.annotation.Bean;
import org.springframework.context.annotation.Configuration;
import org.springframework.web.servlet.config.annotation.CorsRegistry;
import org.springframework.web.servlet.config.annotation.WebMvcConfigurer;

import org.springframework.security.crypto.bcrypt.BCryptPasswordEncoder;

xinyisherryz
@Configuration
public class CORSConfig {

  1 usage
  @Value("${env.ALLOWED_ORIGIN}")
  private String allowedOrigin;

  xinyisherryz
  @Bean
  public WebMvcConfigurer corsConfigurer() {
    return addCorsMappings(registry) → {
        registry.addMapping( pathPattern: "/**")
            .allowedOrigins(allowedOrigin)
            .allowedMethods("GET", "POST", "PUT", "DELETE", "HEAD", "OPTIONS")
            .allowedHeaders("*")
            .allowCredentials(true);
    };
  }

  xinyisherryz
  @Bean
  public BCryptPasswordEncoder bCryptPasswordEncoder() {
    return new BCryptPasswordEncoder();
  }
```



```java
package cs5500.fireemblem.sprint;

import org.springframework.context.annotation.Bean;
import org.springframework.context.annotation.Configuration;
import org.springframework.security.config.annotation.web.builders.HttpSecurity;
import org.springframework.security.config.annotation.web.configuration.EnableWebSecurity;
import org.springframework.security.config.annotation.web.configurers.AbstractHttpConfigurer
import org.springframework.security.web.SecurityFilterChain;

xinyisherryz
@Configuration
@EnableWebSecurity

public class SecurityConfig {

  xinyisherryz
  @Bean
  public SecurityFilterChain filterChain(final HttpSecurity http) throws Exception {
    return http
        .httpBasic(AbstractHttpConfigurer::disable)
        .csrf(AbstractHttpConfigurer::disable)
        .build();
  }
}
```

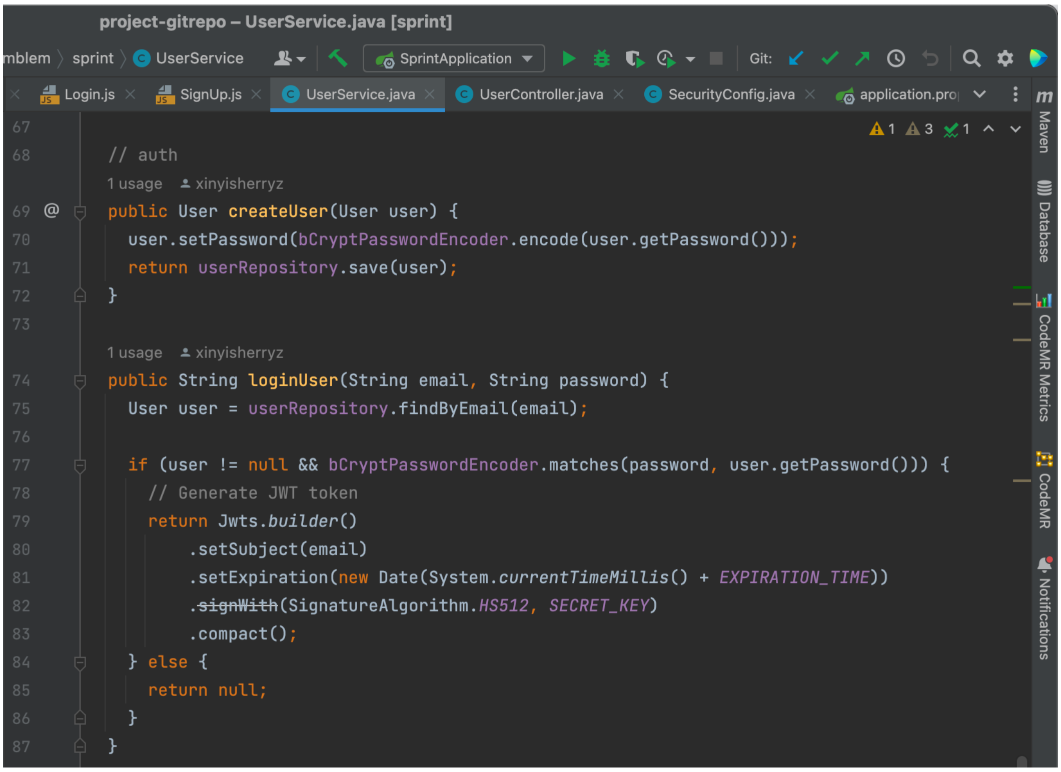


Environment properties

The following properties are passed in the application as environment properties. Learn more

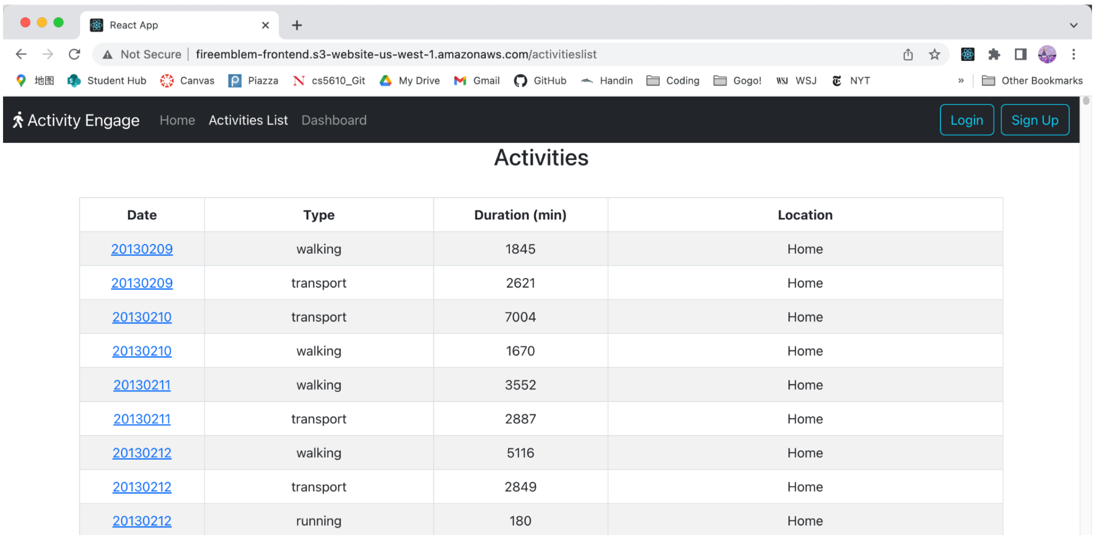
Name	Value
ALLOWED_ORIGIN	http://fireemblem-frontend.s3-website-us-west-1.a
GRADLE_HOME	/usr/local/gradle
JAVA_HOME	/usr/lib/jvm/java
M2	/usr/local/apache-maven/bin
M2_HOME	/usr/local/apache-maven
MONGO_CLUSTER	cluster0.mdrrqyn.mongodb.net
MONGO_DATABASE	sprint
MONGO_PASSWORD	fireemblem
MONGO_USER	sprint
PORT	8080

4. User authentication is implemented through the use of JWT and BcryptPasswordEncoder, ensuring secure account management and login validation for enhanced system security.



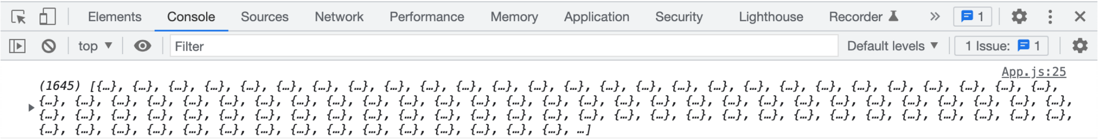
```java
    // auth
    public User createUser(User user) {
      user.setPassword(bCryptPasswordEncoder.encode(user.getPassword()));
      return userRepository.save(user);
    }

    public String loginUser(String email, String password) {
      User user = userRepository.findByEmail(email);

      if (user != null && bCryptPasswordEncoder.matches(password, user.getPassword())) {
        // Generate JWT token
        return Jwts.builder()
            .setSubject(email)
            .setExpiration(new Date(System.currentTimeMillis() + EXPIRATION_TIME))
            .signWith(SignatureAlgorithm.HS512, SECRET_KEY)
            .compact();
      } else {
        return null;
      }
    }
```

5. End-to-end tests are conducted to simulate user interactions with the system, ensuring that the entire application operates as expected and provides a seamless user experience.



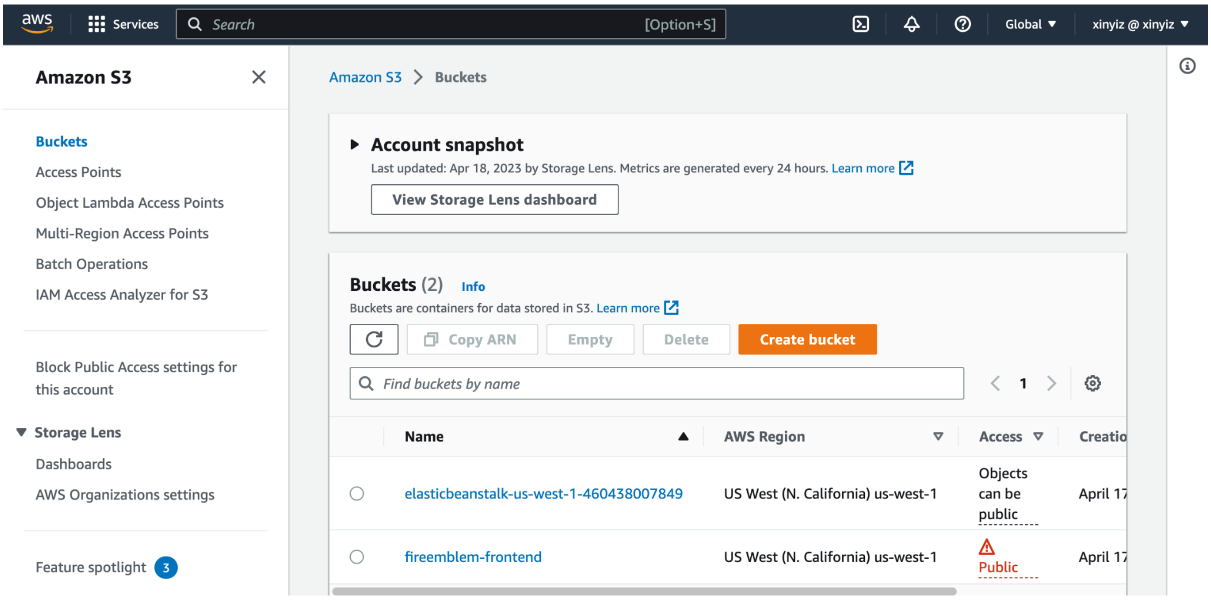
Activities

Date	Type	Duration (min)	Location
20130209	walking	1845	Home
20130209	transport	2621	Home
20130210	transport	7004	Home
20130210	walking	1670	Home
20130211	walking	3552	Home
20130211	transport	2887	Home
20130212	walking	5116	Home
20130212	transport	2849	Home
20130212	running	180	Home



```
(1645) [{…}, {…}, {…}, {…}, {…}, {…}, {…}, {…}, {…}, {…}, {…}, {…}, {…}, {…}, {…}, {…}, {…}, {…}, {…}, {…}, {…}, {…}, {…}, {…}, {…}, {…}, {…},
{…}, {…}, {…}, {…}, {…}, {…}, {…}, {…}, {…}, {…}, {…}, {…}, {…}, {…}, {…}, {…}, {…}, {…}, {…}, {…}, {…}, {…}, {…}, {…}, {…}, {…}, {…}, {…}, {…},
{…}, {…}, {…}, {…}, {…}, {…}, {…}, {…}, {…}, {…}, {…}, {…}, {…}, {…}, {…}, {…}, {…}, {…}, {…}, {…}, {…}, {…}, {…}, {…}, {…}, {…}, {…}, {…}, {…},
{…}, {…}, {…}, {…}, {…}, {…}, {…}, {…}, {…}, {…}, {…}, {…}, {…}, {…}, {…}, …]
```

6. The frontend React application is hosted on an Amazon S3 bucket, while the backend is deployed using AWS Elastic Beanstalk for a reliable and scalable cloud infrastructure.



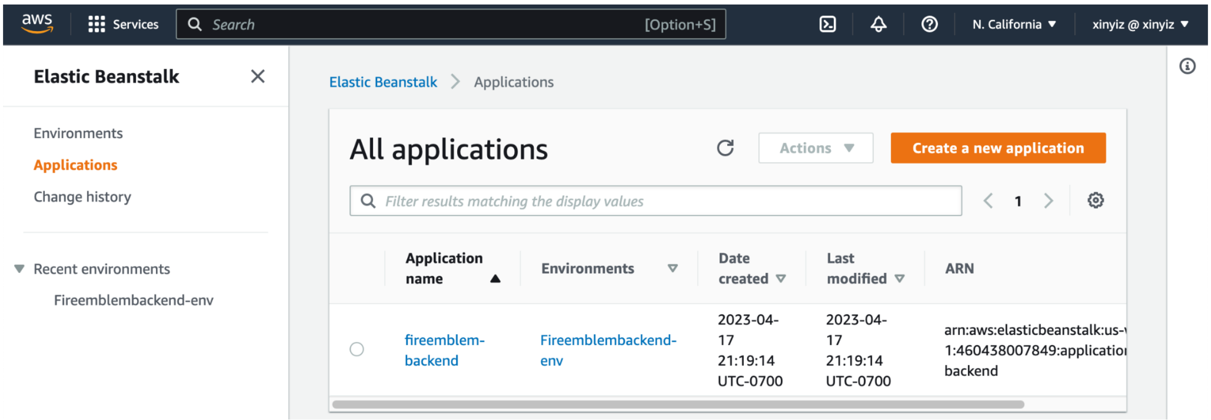
Account snapshot

Last updated: Apr 18, 2023 by Storage Lens. Metrics are generated every 24 hours. Learn more

Buckets (2) Info

Buckets are containers for data stored in S3. Learn more

	Name	AWS Region	Access	Creatio
○	elasticbeanstalk-us-west-1-460438007849	US West (N. California) us-west-1	Objects can be public	April 17
○	fireemblem-frontend	US West (N. California) us-west-1	⚠ Public	April 17

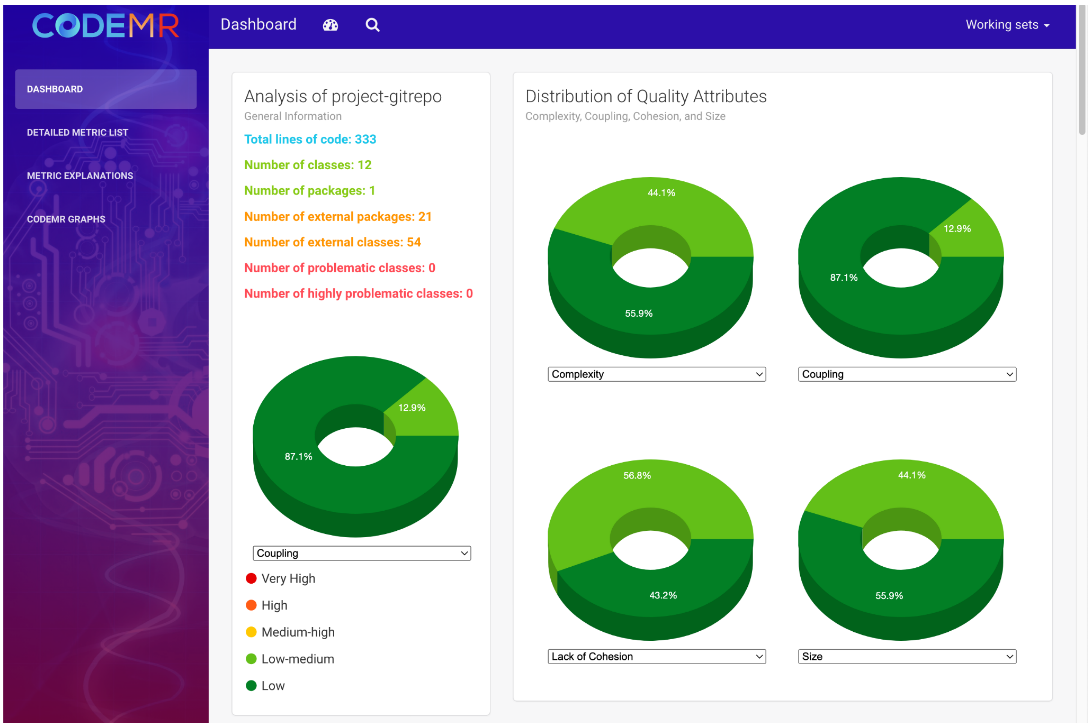


	Application name	Environments	Date created	Last modified	ARN
○	fireemblem-backend	Fireemblembackend-env	2023-04-17 21:19:14 UTC-0700	2023-04-17 21:19:14 UTC-0700	arn:aws:elasticbeanstalk:us-1:460438007849:applicatio backend

* Code Quality

We have tested our code quality using CodeMR again and confirmed that the codebase has positive and robust attributes.

1. Coupling (87.1% Low, 12.9% Low-Medium): The modules exhibit good encapsulation, with a low degree of dependency on each other.
2. Complexity (55.9% Low, 44.1% Low-Medium): The codebase has a low to moderate level of complexity,which suggests that it is convenient for developers to comprehend and make changes.
3. Lack of Cohesion (43.2% Low, 56.8% Low-Medium): The codebase has a high level of cohesion, with relevant functions confined to the same modules or classes.
4. Size (49% Low, 51% Low-Medium): The codebase has a manageable size that is convenient for developers to work with and maintain.



Analysis of project-gitrepo

General Information

Total lines of code: 333

Number of classes: 12

Number of packages: 1

Number of external packages: 21

Number of external classes: 54

Number of problematic classes: 0

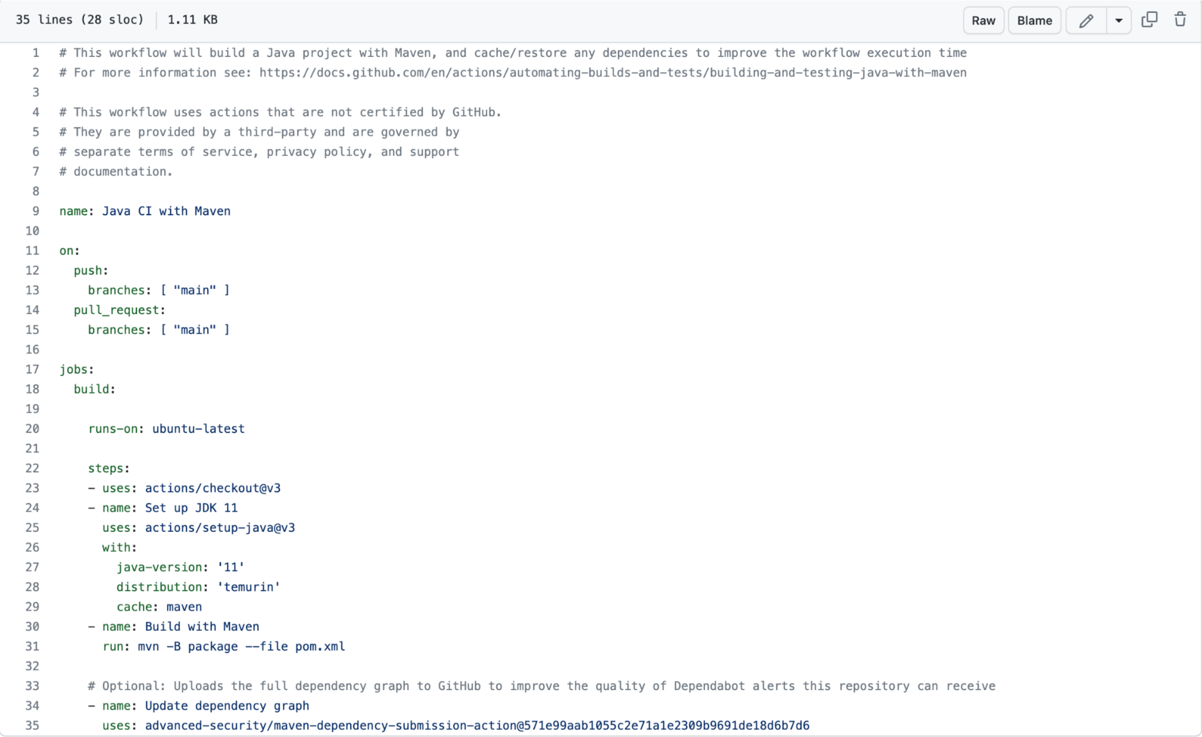
Number of highly problematic classes: 0

Distribution of Quality Attributes

Complexity, Coupling, Cohesion, and Size

* CI pipeline on Github

We have created a ".github/workflows" directory in our GitHub repository and added a new YAML file that contains our CI pipeline configuration for a Java project with Maven. We saved and committed the YAML file to our repository. Then, we can edit the YAML file to define tasks, such as trigger events (like push or pull requests), runtime environment, dependencies, and scripts to run.



35 lines (28 sloc) | 1.11 KB

```yaml
# This workflow will build a Java project with Maven, and cache/restore any dependencies to improve the workflow execution time
# For more information see: https://docs.github.com/en/actions/automating-builds-and-tests/building-and-testing-java-with-maven

# This workflow uses actions that are not certified by GitHub.
# They are provided by a third-party and are governed by
# separate terms of service, privacy policy, and support
# documentation.

name: Java CI with Maven

on:
  push:
    branches: [ "main" ]
  pull_request:
    branches: [ "main" ]

jobs:
  build:

    runs-on: ubuntu-latest

    steps:
    - uses: actions/checkout@v3
    - name: Set up JDK 11
      uses: actions/setup-java@v3
      with:
        java-version: '11'
        distribution: 'temurin'
        cache: maven
    - name: Build with Maven
      run: mvn -B package --file pom.xml

    # Optional: Uploads the full dependency graph to GitHub to improve the quality of Dependabot alerts this repository can receive
    - name: Update dependency graph
      uses: advanced-security/maven-dependency-submission-action@571e99aab1055c2e71a1e2309b9691de18d6b7d6
```