

به نام خدا

فاز دوم پروژه

مهندسی نرم افزار

دکتر فرید فیضی - ترم ۱۴۰۳ دانشگاه گیلان



The background is a light blue-grey gradient with various abstract geometric elements. There are several overlapping circles and semi-circles in shades of blue and grey. A central white rectangular box with a thin black border contains the text. To the left of the box, there are some small black dots and a larger dark grey circle. To the right, there are more dots and a vertical line. The overall style is modern and minimalist.

اعضای گروه :

ثنا منصوری

فرحان باقری

محمد رضا اصفهانی

حسین یداللهی

مهدی نظام دوست

تحلیل کیفیت کد پروژه humhub با SonarQub

فهرست

صفحه	بخش
4	مقدمه
4	بررسی دقیق شاخص های کلیدی
9	تحلیل خطاها
11	راهکار ها
12	جمع بندی

در پروژه‌های نرم‌افزاری بزرگ مانند humhub با بیش از ۲۵۰ هزار خط کد، حفظ کیفیت نرم‌افزار یک چالش مهم است. کیفیت پایین کد نه تنها باعث افزایش خطاهای اجرایی و باگ‌ها می‌شود، بلکه هزینه‌های نگهداری و توسعه را نیز به شدت افزایش می‌دهد و می‌تواند امنیت سیستم را به مخاطره بیندازد.

ابزار SonarQube با آنالیز استاتیک کد و شناسایی مشکلات رایج، کمک می‌کند توسعه‌دهندگان به نقاط ضعف توجه کنند و به بهبود مستمر کیفیت کد بپردازند. در این گزارش، تمام جنبه‌های مهم کیفیت کد پروژه humhub بررسی و تحلیل خواهد شد.

2. بررسی دقیق شاخص‌های کلیدی

Software Quality ?	
Security	3
Reliability	412
Maintainability	4.8k
Severity ?	
Blocker	14
High	2.1k
Medium	2k
Low	891
Info	73

2.1. امنیت (Security)

- تعریف: امنیت به میزان مقاومت کد در برابر حملات و آسیب‌پذیری‌ها اشاره دارد.
- وضعیت پروژه: در پروژه humhub تعداد ۳ مسئله امنیتی باز گزارش شده است که شامل ۱۴ مورد بحرانی (Blocker) می‌شود.
- مفهوم **Blocker**: این‌ها بدترین نوع مشکلات امنیتی هستند که اگر برطرف نشوند، می‌توانند باعث نفوذ هکرها، سرقت داده‌ها یا کنترل کامل سیستم شوند.
- نمونه آسیب‌پذیری‌ها: ممکن است شامل XSS، Injection، CSRF یا مشکلات مدیریت دسترسی باشند.
- راهکار:

- اجرای بررسی امنیتی جامع (Security Audit) روی کد
- استفاده از ابزارهای تخصصی (SAST (Static Application Security Testing)
- آموزش تیم توسعه در زمینه نوشتن کد امن
- پیاده‌سازی تست نفوذ و رفع آسیب‌پذیری‌ها در سریع‌ترین زمان ممکن

2.2. قابلیت اطمینان (Reliability)

- تعریف: قابلیت اطمینان نشان‌دهنده توانایی سیستم برای اجرا بدون خطا و رفتار پیش‌بینی‌شده است.
 - مشکلات پروژه: ۴۱۲ مشکل باز در این حوزه شناسایی شده است.
 - علل رایج:
- استفاده نادرست از متغیرها
 - مدیریت ناقص استثنایا (Exception Handling)

- دسترسی به متغیرهای مقداردهی نشده
 - خطاهای منطقی
 - پیامدها: باعث کرش برنامه، رفتار غیرمنتظره و کاهش رضایت کاربران می‌شود.
 - راهکار:
 - افزودن تست‌های خودکار (Unit, Integration)
 - اصلاح کدها و پوشش کامل مسیرهای کد
 - بهبود مدیریت خطا و کنترل استثناءها
 - اجرای فرآیندهای Code Review دقیق
-

2.3. قابلیت نگهداری (Maintainability)

- تعریف: قابلیت نگهداری به سهولت تغییر و توسعه کد در طول زمان اشاره دارد.
- نتایج: حدود ۴,۸ هزار ایراد باز در این حوزه و وضعیت کلی "A" یعنی نسبتاً خوب.
- نکته مهم: وضعیت "A" به معنی کامل نبودن کد نیست، بلکه نسبت به معیارهای SonarQube وضعیت مناسبی است.
- مشکلات رایج:
 - پیچیدگی بیش از حد توابع و کلاس‌ها
 - کد تکراری (duplications)
 - مستندسازی ناقص
 - نام‌گذاری نامناسب

- راهکار:

- بازنویسی کدهای پیچیده و جدا کردن به ماژول‌های کوچکتر
 - حذف کدهای تکراری و استفاده از الگوهای طراحی
 - مستندسازی دقیق‌تر
 - ایجاد استانداردهای کدنویسی در تیم
-

2.4. پوشش تست (Coverage)

- تعریف: درصد کدی که توسط تست‌های خودکار پوشش داده می‌شود.
- وضعیت پروژه: صفر درصد! یعنی هیچ تست واحد یا خودکاری ندارید.
- پیامد:

- افزایش احتمال ورود باگ‌های مخفی
- دشواری در اطمینان از صحت تغییرات
- افت کیفیت نرم‌افزار در نسخه‌های بعدی

- راهکار:

- تعریف چارچوب تست و استراتژی تست جامع
- استفاده از فریمورک‌های تست PHP مانند PHPUnit، JavaScript مانند Jest و غیره
- آموزش تیم توسعه در نوشتن تست
- پیاده‌سازی CI/CD برای اجرای خودکار تست‌ها

2.5. تکرار کد (Duplications)

- تعریف: بخش‌هایی از کد که بدون تغییر یا با تغییرات جزئی در چند مکان تکرار شده‌اند.
- آمار: ۷,۲ درصد کد تکراری روی ۳۶۴ هزار خط.
- مشکل:
 - افزایش حجم کد
 - افزایش ریسک ناسازگاری هنگام تغییر
 - سخت شدن نگهداری
- راهکار:
 - شناسایی بلوک‌های تکراری با ابزارهای تحلیل کد
 - ایجاد توابع یا کلاس‌های مشترک
 - به کارگیری الگوهای طراحی مثل Strategy، Singleton، Factory
 - Refactoring دوره‌ای

3. تحلیل خطاها و پیام‌های مهم گزارش شده

humhub / main

Overview Issues Security Hotspots Measures Code Activity Project Settings Project Information

5,115 issues

Gruntfile.js

Unexpected var, use let or const instead.

Unexpected var, use let or const instead.

Unexpected var, use let or const instead.

100 of 5,115 shown Show More

Group parts of the regex together to make the intended operator precedence explicit.

Alternatives in regular expressions should be grouped when used with anchors [javascript:S5850](#)

Line affected: L109 • Effort: 10min • Introduced: 6 days ago

Open Not assigned regex

Where is the issue? Why is this an issue? Activity More info

humhub > /Gruntfile.js

Open in IDE See all issues in this file

```
104 farhan...
105         },
106         uglify: {
107             compress: {
```

humhub / main

Overview Issues Security Hotspots Measures Code Activity Project Settings Project Information

5,115 issues

Unexpected var, use let or const instead.

Unexpected var, use let or const instead.

Unexpected var, use let or const instead.

Group parts of the regex together to make the intended operator precedence explicit.

100 of 5,115 shown Show More

Replace "or" with "||".

"&&" and "||" should be used [php:S2010](#)

Line affected: L23 • Effort: 5min • Introduced: 6 days ago

Open Not assigned suspicious

Where is the issue? Why is this an issue? Activity More info

```
24 defined('YII_ENV_TEST') or define('YII_ENV_TEST', true);
25
26
27 require(__DIR__ . '/protected/vendor/autoload.php');
```

Replace "or" with "||".

Remove the parentheses from this "require" call.

3.1. پیام‌های "Unexpected var, use let or const instead"

- نشان‌دهنده استفاده از `var` در JavaScript است که 6ES منسوخ کرده است.
- چرا مهم است؟
 - `var` دامنه بلوک ندارد و ممکن است باعث خطاهای نامشخص شود
 - `let` و `const` به دامنه بلوک محدود شده و خوانایی کد را افزایش می‌دهند
- اصلاح: تمام متغیرها را با `let` (برای متغیرهای تغییرپذیر) یا `const` (برای ثابت‌ها) جایگزین کنید.

3.2. پیام "'|' Replace 'or' with"

- نشان‌دهنده استفاده نادرست از عملگر منطقی `or` است.
- در PHP و JavaScript باید از `||` برای OR منطقی استفاده شود.
- اصلاح این موضوع باعث جلوگیری از خطاهای منطقی می‌شود.

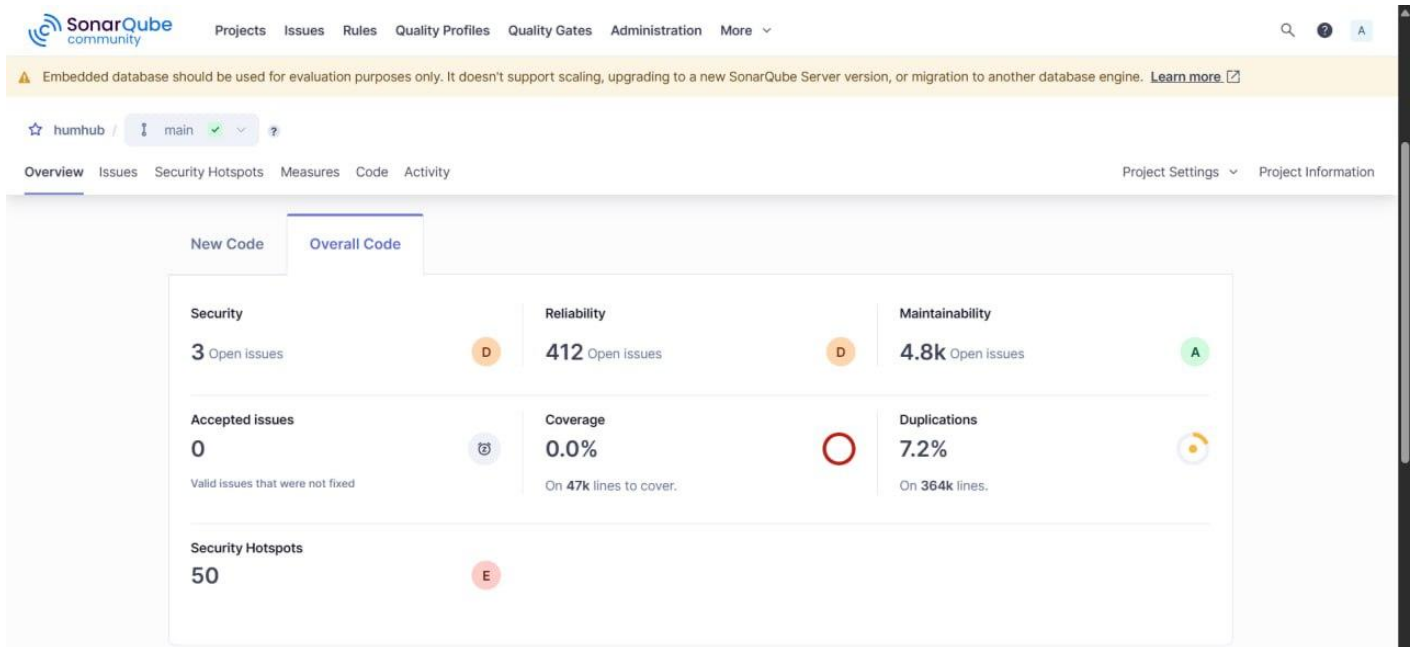
3.3. گروه‌بندی درست در عبارات منظم (Regex)

- هشدارها بیانگر این است که باید بخش‌های مختلف regex با پرانتز گروه‌بندی شود تا اولویت عملگرها واضح باشد.
- عدم رعایت این نکته می‌تواند باعث بروز خطاهای ناخواسته در تطبیق الگوها شود.

3.4. اصلاحات فراخوانی تابع `require`

- توصیه به حذف پرانتزهای غیرضروری در برخی فراخوانی‌ها.

- رعایت این موارد کمک به خوانایی و استاندارد بودن کد می‌کند.



4. راهکارها و پیشنهادات عملیاتی

4.1. طراحی برنامه بهبود کیفیت

- تشکیل تیم ویژه کیفیت با مسئولیت اختصاصی رفع خطاها
- اولویت‌بندی مسائل بر اساس شدت (Security Blockers اولویت اول)
- استفاده از ابزارهای اتوماتیک برای رفع برخی ایرادات (Linting, Formatter)
- بازنگری کامل معماری کد در بخش‌های پر خطا و تکراری

4.2. پیاده‌سازی تست‌های خودکار

- تعریف اهداف پوشش تست (حداقل ۷۰٪ خطوط بحرانی)
- استفاده از Mocking و Dependency Injection برای تست آسان‌تر
- راه‌اندازی CI/CD برای اجرای تست‌ها در هر تغییر کد

4.3. آموزش و مستندسازی

- برگزاری کارگاه‌های آموزشی برای تیم توسعه در زمینه امنیت و نگهداری کد
- مستندسازی کامل کد و فرآیندها برای تسهیل نگهداری و توسعه
- تدوین استانداردهای کدنویسی و مرور دوره‌ای

4.4. مدیریت تکنولوژی و به‌روزرسانی

- بررسی به‌روزرسانی کتابخانه‌ها و فریم‌ورک‌ها
- استفاده از نسخه‌های پایدار و امن
- حذف کدهای منسوخ و ناکارآمد

5. جمع‌بندی

خلاصه اینکه پروژه humhub با وجود حجم بالای کد و کاربری گسترده، نیازمند توجه ویژه به موارد امنیتی، پوشش تست و بهبود قابلیت اطمینان است. هرگونه غفلت در این زمینه‌ها می‌تواند منجر به بروز مشکلات جدی در عملکرد و امنیت سیستم شود.

در کنار آن، حفظ و ارتقای قابلیت نگهداری و کاهش کدهای تکراری از عوامل کلیدی برای تسریع توسعه و کاهش هزینه‌هاست.

My FavoritesAll

Filters

Quality Gate

Passed

1

Failed

0

Security

A ≥ 0 info issues

0

B ≥ 1 low issue

0

C ≥ 1 medium issue

0

D ≥ 1 high issue

1

E ≥ 1 blocker issue

0

Create Project

Search for projects...

Perspective

Overall Status

Sort by

Name

1 project(s)

☆ humhub PUBLIC

Passed

Last analysis: 27 minutes ago · 251k Lines of Code · PHP, CSS, ...

D 3

D 412

A 4.8k

E 0.0%

0.0%

7.2%

Security

Reliability

Maintainability

Hotspots Reviewed

Coverage

Duplications

1 of 1 shown