

گزارش نهایی پروژه درس مهندسی نرم افزار

شایان شکر فروش
کیانوش عباسی
محمدرضا کارگر
امیررضا مزینی
علی ملکی

۱ گزارش Retrospective

کارهای خوبی که کردیم:

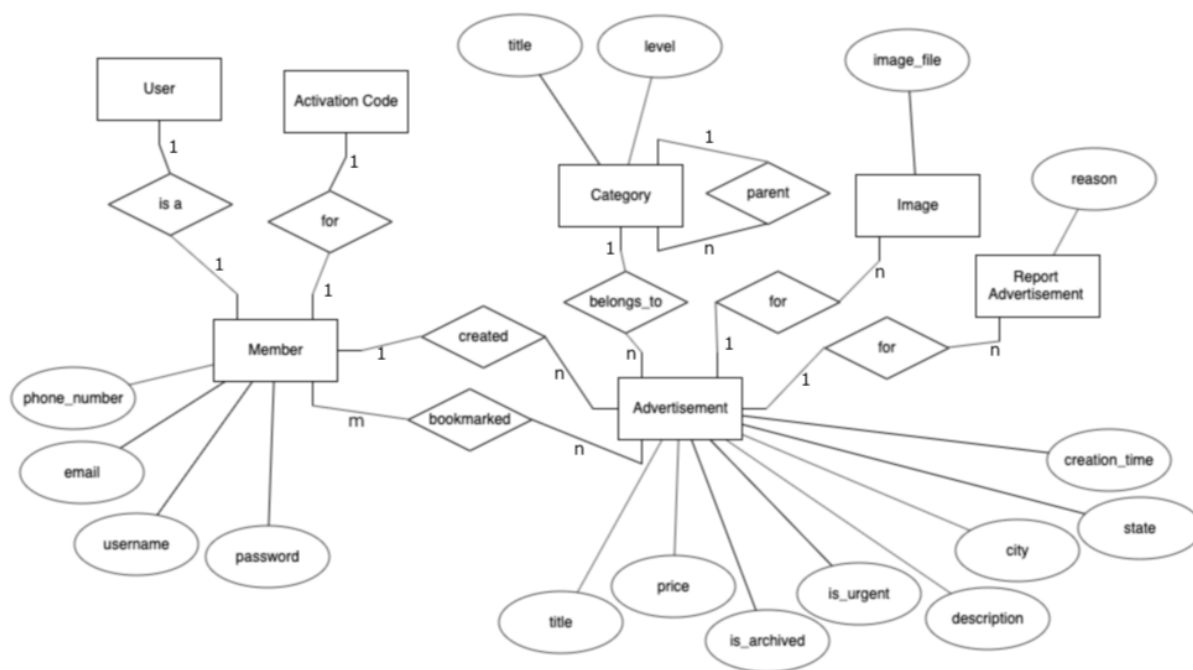
سعی کردیم به طور مرتب usecase ها و تسک های مرتبط به آن ها را در ابتدا و همچنین طول پروژه تعریف کنیم و توضیحات مربوط به آن ها را نیز در کنار هر تسک قرار دهیم تا در صورت رجوع مجدد آن را راحت تر به یاد بیاوریم. هر فردی سعی داشت بر حسب توانایی های خودش تسک هایی که تعریف شده بود را بردارد. مثلا بعضی اعضا که به بخش front-end تسلط بیشتری داشتند بیشتر تسک هایی که انجام دادند مربوط به آن بخش بود. اعضا سعی می کردند در صورت برخورد به یک مسئله چالشی با شخصی که شاید اطلاعات بیشتری داشت صحبت کنند و یا با کل تیم در میان می گذاشتند تا با همکاری هم مسئله را رفع کنند. از همان ابتدا سعی کردیم که کارفرما را وارد بازی کنیم تا تمام موضوعاتی که شاید گنگ بودند را برطرف کنیم. تلاش تیم بر این بود که در پیاده سازی back-end و front-end تا جای ممکن از ماژول هایی استفاده شود که قبلا کارایی آنها تست شده بودند. از همین رو به موضوع reuse توجه کردیم. به قول معروف سعی نکردیم دوباره چرخ را اختراع کنیم. سعی شد که continuous delivery رعایت شود به گونه ای که هر feature ای زده می شد به گیت اضافه می شد و به صورت manual تست می شد که آیا با بقیه سیستم درست کار می کند و یا side-effect ای وجود دارد یا نه.

کارهای بدی که کردیم:

در میان پروژه به علت مشغله زیادی که برای اعضای تیم به وجود آمد وقفه ای به وجود آمد که ما را از برنامه زمانی که مدنظر داشتیم عقب انداخت. بهتر بود که همراه با اتمام هر بخش unit test برای آن بنویسیم به جای آنکه به صورت دستی وقت زیادی بگذاریم و تست کنیم. در تیم سعی می کردیم پس از اتمام هر sprint درباره آن صحبت کنیم اما مستندسازی از نتیجه گیری ها و feedback ها انجام نمی دادیم که شاید اگر انجام می دادیم آسان تر مشکلاتمان را رفع می کردیم.

۲ طراحی پایگاه داده

نمودار ER مربوط به طراحی پایگاه داده در تصویر زیر مشاهده می‌شود:



شکل ۱: نمودار ER پایگاه داده

۳ بررسی کیفیت کد

برای بررسی کیفیت کد از ابزار SonarQube استفاده شد. گزارش حاصل از بررسی پروژه توسط این ابزار، در نشانی زیر موجود است:
https://sonarcloud.io/dashboard?id=AliMaleky7997_sonar_import

تصویر یکی از نمودارهای حاصل از این بررسی در ادامه آورده می‌شود.



شکل ۲: نمودار حاصل از بررسی کیفیت کد

۴ تست

برای تست کردن کدهای زده شده، از کتابخانه‌های موجود در Django و همچنین ابزار Selenium استفاده شد. همان‌طور که در ادامه مشاهده می‌شود، Line Coverage تست‌های طراحی شده ۹۰٪ است.

83% files, 90% lines covered	
Element	Statistics, %
.idea	
ads	83% files, 92% lines covered
divar	66% files, 100% lines covered
media	
src	
static	
templates	
users	87% files, 86% lines covered
venv	
.coverage	
db.sqlite3	
manage.py	41% lines covered
requirements.txt	

شکل ۳: بررسی coverage کل پروژه

83% files, 92% lines covered in 'ads'	
Element	Statistics, %
migrations	100% files, 100% lines covered
templates	
test	100% files, 93% lines covered
__init__.py	
admin.py	100% lines covered
apps.py	not covered
forms.py	100% lines covered
models.py	87% lines covered
tests.py	not covered
urls.py	100% lines covered
views.py	90% lines covered

شکل ۴: بررسی coverage اپلیکیشن ads

87% files, 86% lines covered in 'users'	
Element	Statistics, %
migrations	100% files, 100% lines covered
templates	
__init__.py	
admin.py	100% lines covered
apps.py	not covered
forms.py	60% lines covered
models.py	80% lines covered
tests.py	100% lines covered
urls.py	100% lines covered
views.py	88% lines covered

شکل ۵: بررسی coverage اپلیکیشن users