

КРИТЕРІЇ ТЕСТУВАННЯ САЙТІВ

Висоцька Н., студентка гр. МКІн-171
 Науковий керівник: **Казимир В.В.**, д.т.н., професор
Чернігівський національний технологічний університет

Тестування сайту - це останній і обов'язковий етап технічної розробки сайту. Він грає найважливішу роль в процесі створення ресурсу, так як саме від якості тестування залежить подальше життя ресурсу. Для організації тестування веб-сайту передбачена спеціально розроблена методика, згідно з якою і здійснюється перевірка.

На даний момент, ми все частіше стикаємося з проблемою, коли клієнти не розуміють які етапи перевірки повинен пройти їх продукт перед тим як запустити його у світ. Зараз з'являється велика кількість сайтів, які при запуску на проходять звичайний тест реєстрації великої кількості користувачів, такі проблеми мають вирішуватися на етапі тестування. В силу своєї необізнаності та бажання заощадити кошти, люди дозволяють собі обійти цей етап стороною. Пропоную розібратися чому тестування сайтів є навид'ємною частиною при їх створенні.

Розпочнемо з аналізу всіх критеріїв тестування сайтів:

1) Починається все з підготовчих робіт - тестувальник вивчає отриману документацію (аналізує функціонал по тех. Завданням, вивчає кінцеві макети сайту і становить план тесту для подальшого тестування)

2) Функціональне тестування - найбільш тривалий етап перевірки ресурсу. Суть цього процесу полягає в перевірці всього описаного функціоналу:

- Перевірки роботи всіх обов'язкових функцій сайту;
- Тестування працездатності призначених для користувача форм на сайті (наприклад, зворотний зв'язок, додавання коментаря в блог);
- Перевірки роботи пошуку (включаючи релевантність результатів);
- Перевірки гіперпосилань, пошук неробочих посилань;
- Перевірки подгрузки файлів на сервер;
- Перевірки працездатності лічильників, встановлених на сторінках сайту;
- Перегляд на відповідність вмісту сторінок сайту вихідного контенту, наданого замовником.

3) Тестування Верстки - при перевірці верстки насамперед тестувальник перевіряє розташування елементів, відповідність їх позицій наданим макетів, а так само перевіряє оптимізацію зображень і графіки.

4) Usability тестування - проводиться для оцінки зручності продукту у використанні, заснований на залученні користувачів в якості тестувальників і аналіз отриманих результатів.

5) Тестування безпеки - На даній стадії тестування фахівець перевіряє - чи немає у користувачів доступу до службових / закритим сторінкам а так само проводить перевірку захисту всіх критично важливих сторінок (наприклад, розділу адміністрування сайту) від зовнішнього впливу.

6) Тестування продуктивності сайту - проводиться з метою визначення швидкодії сайту або його частини під певним навантаженням.

Протягом усього етапу тестування, фахівець створює і доповнює звіт про виявлені ним помилки. Даний звіт передається учасникам проекту, після чого керівник проекту визначає відповідального за виправлення кожної з помилок. Далі визначається графік виправлення помилок, після чого проводиться повторне тестування з метою контролю якості виправлення помилок, а так само відсутності нових. Дана процедура повторюється поки сайт не буде відповідати специфікаціям технічного завдання. Саме тому тестування - такий довгий процес.

По завершенню тестування проект готовий до розміщення на сервері і повноцінній роботі, ефективно і стабільно виконуючи покладені на нього бізнес-функції.

Проаналізувавши критерії тестування сайтів, можна прийти до висновку, що саме тестування є гарантом спокійного сну як для замовника, так і для команди розробників веб-сайту.

Список використаних джерел

1. «Тестирование программного обеспечения. Базовый курс», Св'ятослав Куліков
2. «Ключевые процессы тестирования. Планирование, подготовка, проведение, совершенствование», Рекс Блэк
3. «Быстрое тестирование», Роберт Калбертсон, Крис Браун, Гэри Кобб

РЕАЛІЗАЦІЯ ІНФОРМАЦІЙНИХ СИСТЕМ ЗА ДОПОМОГОЮ FIREBASE

Мельниченко Р.Ю., студ. гр. МКІн-171
 Науковий керівник: **Бивойно П.Г.**, к.т.н., доц.
Чернігівський національний технологічний університет

Інформаційні системи все глибше проникають в наше життя. Учора віддалене керування різноманітними системами, чи то роботами-маніпуляторами, чи то побутовими приладами було фантастикою, а сьогодні це звичне явище. Інформаційні системи теж повинні йти в ногу з часом, та вдосконалюватися. Деякі досить легко піддаються масштабуванню, інші розширювати майже неможливо, через що потрібно створювати все з нуля. Інструментарії розробників теж не стоять на місці, а постійно оновлюються та розвиваються. Спочатку дані зберігали у файли, зараз – у бази даних (БД), яких є декілька типів, на початку програма виконувалася тільки в одному потоці, пізніше

– розробник повинен був забезпечити створення окремого потоку під кожне звернення користувача та синхронізувати отримані дані, зараз дані функції виконують фреймворки, а розробнику залишається лише реалізувати логіку системи. Проте, зазвичай, над системою працює не один розробник, а ціла команда, оскільки сучасна інформаційна система включає в себе різноманітні функції, в тому числі і синхронізація, та реалізовується для декількох платформ. Для великих систем, де реалізовується декілька складних логік її використання, це має рацію, оскільки обробка великої кількості даних потребує значної обчислювальної потужності машин, ресурси яких використовує дана система. Оскільки, зазвичай обчислювальна потужність клієнтів недостатня, цілком природньо, що обробку даних переносять на більш потужні машини системи – сервери. Але для середніх та малих, або спрощених, систем, серверна частина виконує роль прошарку між клієнтським додатком та базою даних. І таких систем, наразі, більшість, у яких серверну частину можна було б спростити чи, взагалі, видалити. Саме для подібних систем було створено firebase.

Firebase – це платформа для розробки web- та mobile-застосунків, яка надає базу даних та бекенд як службу, в режимі реального часу. Це означає, що кожен клієнт може додати себе, як наглядча, та відслідковувати зміни даних у таблицях, або їх частинах (при певних умов), на які цей клієнт підписаний. Водночас, при зміні даних, клієнт моментально отримує сповіщення про зміну даних та оновлені дані. Дана особливість дозволяє використовувати БД «напрямку від клієнта», забезпечує максимальну швидкодню, оскільки всі основні обчислення бере на себе сервер БД. Також забезпечується безпека передачі даних, оскільки API та сервер firebase використовують шифрування, можливість отримати доступ до акаунтів у соціальних мережах користувачів, за їх згоди, та отримати звітлія інформацію (так звана «авторизація в один клік»), налаштування відправки сервером сповіщень клієнтам (push сповіщення), можливість додати до БД сховище для файлів тощо.

Інтегрувати firebase до проекту може frontend розробник, оскільки для більшості з існуючих клієнтських платформ створені та відкриті SDK для роботи з firebase. Також існують розширення для інтеграції з firebase апаратних компонентів, наприклад Raspberry PI. До того ж існують SDK для серверних рішень, наприклад node.js, які можуть бути використані як механізм для розширення функціоналу системи. Також існує REST API, за допомогою якого можна вносити та отримувати дані в БД, керування відбувається і за допомогою HTTPS запитів, і за допомогою сокетів. Платформа firebase відмінно підходить для таких систем, як чати/меседжери, клієнти соціальних мереж, збір та аналіз статистики, новини, погода, розумний дім/авто, аналітика, бронювання квитків/номерів, покупки товарів/замовлення послуг, різноманітні довідники, записники тощо. Firebase є NoSQL базою даних і, як наслідок, більш простіше піддається до масштабування системи, на відміну від SQL баз, причому як при збільшенні оброблювальних даних, так і при збільшенні кількості користувачів системи.

На даний момент, а особливо після закриття parse.com, firebase набирає все більшу популярність у розробників. На сьогоднішній день ведеться активна робота з переходу на firebase при використанні авторизації через соціальні мережі, зокрема google вже перейшов. Також, відомий сервіс аналітики додатків fabtrics вже інтегрував до себе функціонал firebase.

Отже, можна сказати, що платформ, подібних до firebase – БД у режимі реального часу, було дуже мало або взагалі не існувало, і саме firebase зробило революційний крок до спрощення реалізації інформаційних систем та пришвидшення їх функціонування. Загалом, для бізнесу та різноманітних освітніх та наукових задач, при грамотному використанні firebase, з'являється можливість більш швидше запустити систему в обіг та зменшити витрати на її розробку.

Список використаних джерел

1. Firebase Realtime Database [Електронний ресурс]. – 2018. – Режим доступу до ресурсу: <https://firebase.google.com/docs/database/>.
2. Introduction to Firebase [Електронний ресурс]. – 2017. – Режим доступу до ресурсу: <https://hackernoon.com/introduction-to-firebase-218a23186cd7>.
3. Firebase [Електронний ресурс]. – 2017. – Режим доступу до ресурсу: <https://github.com/firebase/>.
4. Firebase - Wikipedia [Електронний ресурс]. – 2017. – Режим доступу до ресурсу: <https://uk.wikipedia.org/wiki/Firebase>.
5. Firebase Crashlytics graduates from beta [Електронний ресурс]. – 2018. – Режим доступу до ресурсу: <https://firebase.googleblog.com/2018/03/firebase-crashlytics-graduates-from-beta.html>.

УДК 004.42

ВЕБ-ДОДАТОК ДЛЯ ОПЕРАТИВНОГО КОНТРОЛЮ УСПІШНОСТІ УЧНІВ У ШКОЛІ

Подорога В.С., студентка ФЕІТ, **Дорош М.С.**, д.т.н., доц.
Чернігівський національний технологічний університет

Вступ. На початку XXI століття людство вступило в нову стадію свого розвитку - інформаційну еру, яка характеризується виникненням нових систем інформаційних технологій і, як наслідок, нових трансформацій освіти.

Як відомо, освіта є самостійною системою і одночасно стратегічним ресурсом держави. Тому розвинені країни досить активно розробляють та впроваджують інформаційні технології в систему освіти. Отже, мета проекту - створити веб-додаток для оперативного контролю успішності учнів, який сприятиме об'єднанню всіх педагогів, учнів та батьків України в єдину спільноту, модернізації навчального процесу та впровадження сучасних комп'ютерних технологій у школах.