

Міністерство освіти і науки України
Сумський державний університет
Навчально-науковий інститут бізнес-технологій «УАБС»
Кафедра економічної кібернетики

КВАЛІФІКАЦІЙНА МАГІСТЕРСЬКА РОБОТА

на тему «РОЗРОБКА ІНФОРМАЦІЙНОЇ СИСТЕМИ БАГ-ТРЕКІНГУ»

Виконав студент 2 курсу, групи ЕК.м-71а

Спеціальності 051 «Економіка»

(«Економічна кібернетика»)

Спичак Олександр Володимирович

Керівник: к.т.н, доцент

Гриценко Костянтин Григорович

ЗМІСТ

ВСТУП.....	3
1 ДОСЛІДЖЕННЯ СТАНУ АВТОМАТИЗАЦІЇ ПРОЦЕСУ ДОКУМЕНТУВАННЯ ЗНАЙДЕНИХ НЕВІДПОВІДНОСТЕЙ З ОЧІКУВАНИМ РЕЗУЛЬТАТОМ	5
1.1 Основні засади в розробці ІС баг-трекінгу.....	5
1.2 Структурні особливості системи.....	8
1.3. Існуючі інструменти розробки ІС	9
1.4 Багтрекери-аналоги.....	13
2.1. Опис ключових елементів системи.....	22
2.2. Розробка загальної структури ІС.....	26
3 РЕАЛІЗАЦІЯ ПРОТОТИПУ ІНФОРМАЦІЙНОЇ СИСТЕМИ БАГ-ТРЕКІНГУ .	31
3.1 Опис основних процедур та залучених модулів у розробці ІС.....	31
3.2. Опис послідовного алгоритму роботи багтрекера.....	32
3.3 Оцінка очікуваного ефекту від впровадження ІС баг-трекінгу	34
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ.....	42
ДОДАТКИ.....	46

ВСТУП

Ні для кого не секрет, що ІТ сфера стрімко розвивається по всьому світу в тому числі і Україна. Великий спектр компаній у цій сфері надають свої послуги по всьому світу. Одним з напрямків ІТ-послуг є саме забезпечення якості ПЗ.

Забезпечення якості (Quality Assurance) — це поняття, яке являє собою сукупність заходів, що охоплюють усі етапи розробки, випуску та експлуатації програмного забезпечення без виключення. Це діяльність (активності) на усіх етапах життєвого циклу ПЗ, яка здійснюється для забезпечення необхідного рівня якості продукту, що розробляється.

QA-інженер - Quality Assurance engineer (Quality Assurance дослівно означає «забезпечення якості») - це фахівець по тестування програмного забезпечення. Його діяльність спрямована на поліпшення процесу розробки ПЗ, запобігання дефектів і виявлення помилок в роботі продукту.

В наш час інформаційних технологій QA-інженерія є досить затребуваною, адже кожен програмний продукт потребує постійної перевірки якості на усіх етапах свого життя. Тестувальнику потрібно описувати всі знайдені, баги в спеціальній системі. Він детально описує проблему, присвоює пріоритет по її усуненню, описує шлях до проблеми і ще вказує безліч деталей, які допоможуть команді розробників виправити всі невідповідності. Для систематизації знайдених помилок програмного продукту використовують спеціальні інформаційні системи, які називають багтрекерами.

Багтрекер — це інформаційна системи, що розроблена для того, щоб тестувальники та програмісти могли відстежувати історію звітів про баги і взаємодіяти один з одним під час своєї роботи. Він може розглядатись як різновид системи відстеження проблем.

Компанія з тестування ПЗ неодмінно використовує багтрекер. Дана програма є незамінним інструментом перевірки ПЗ. Багтрекер є прикладною

програмою, яка розроблена спеціально з метою контролю та обліку всіх знайдених помилок і збоїв.

Система відстеження помилок сприяє економії часу всіх учасників процесу розробки. Багтрекер в простій і зрозумілій формі надає всі виявлені невідповідності та помилки.

Існує величезна кількість баг-трекерів. Всі вони переслідують одну мету, але відрізняються інтерфейсом і функціональністю. Вибір системи залежить від побажань і потреб тестувальника, а також від вимог самого замовника.

Наявність багтрекера є дуже важливим компонентом у розробці програмного забезпечення, вони широко застосовуються компаніями, що розробляють програмні продукти. Загалом, використання багтрекера є однією з «ознак хорошої команди програмістів» [1-11].

Об'єктом дослідження являється галузь QA-інженерії, а оскільки однією з основних цілей тестування є виявлення дефектів та надання інформації для прийняття рішень, то предметом дослідження є конкретний процес документування знайдених дефектів.

Завданням дипломної роботи є:

- Визначення основних засад у розробці ІС баг-трекінгу.
- Визначення структурних особливостей системи.
- Вивчення існуючих інструментів розробки ІС.
- Дослідження існуючих багтрекерів-аналогів.
- Формулювання ключових елементів системи.
- Розробка загальної структури ІС.
- Моделювання основних процедур за допомогою модулів конструктора ІС.
- Опис послідовного алгоритму роботи багтрекера.
- Оцінка очікуваного ефекту від впровадження ІС баг-трекінгу.

Відповідно метою є сама система з її функціоналом, яка допоможе систематизувати процеси тестування ПЗ.

1 ДОСЛІДЖЕННЯ СТАНУ АВТОМАТИЗАЦІЇ ПРОЦЕСУ ДОКУМЕНТУВАННЯ ЗНАЙДЕНИХ НЕВІДПОВІДНОСТЕЙ З ОЧІКУВАНИМ РЕЗУЛЬТАТОМ

1.1 Основні засади в розробці ІС баг-трекінгу

Для початку дослідження необхідно визначити основні поняття та засади в QA-інженерії та в розробці інформаційних систем в цілому.

Взагалі, інформаційна система – це організований набір елементів, що збирає, обробляє, передає, зберігає та надає дані. До складу ІС входять: люди, обладнання, процеси, процедури, дані та операції. Інформаційна система складається з наступних компонентів:

- структура системи;
- функції кожного елемента системи;
- вхід і вихід кожного елемента і системи в цілому;
- мета і обмеження системи та її окремих елементів.

Інформаційна система представляє не тільки функціонування об'єкта управління, а й впливає на нього через органи управління. Вона являється сукупністю інформаційних процесів для задоволення потреби в інформації різних рівнів прийняття рішень. Метою ІС є продукування інформації для використання управлінським апаратом. Система такого роду забезпечує нагромадження, передачу, збереження, оброблення та узагальнення інформації “знизу вгору”, а також конкретизує інформацію “зверху донизу”. ІС призначена для опису економічного об'єкта, його станів, взаємодії, що виражається через економічні показники. Вона покликана своєчасно подавати органам управління необхідну і достатню інформацію для прийняття рішень, якість яких забезпечує високоефективну діяльність об'єкта управління та його підрозділів.

До головних завдань належать:

- виявлення джерел інформації;
- збирання, реєстрація, обробка та видача інформації, що характеризує стан виробництва та управління;

– розподіл інформації між керівниками, підрозділами та виконавцями відповідно до їх участі в управлінні.

Найбільш важливими принципами побудови ефективних інформаційних систем є наступні.

Принцип інтеграції, що полягає в тому, що оброблювані дані, одного разу введені в систему, багаторазово використовуються для вирішення великої кількості завдань.

Принцип системності, що полягає в обробці даних в різних аспектах, щоб отримати інформацію, необхідну для прийняття рішень на всіх рівнях управління.

Принцип комплексності, що полягає в механізації і автоматизації процедур перетворення даних на всіх етапах функціонування інформаційної системи.

Інформаційні системи також класифікуються:

– за функціональним призначенням: виробничі, комерційні, фінансові, маркетингові та ін.;

– по об'єктах управління: інформаційні системи автоматизованого проектування, управління технологічними процесами, управління підприємством (офісом, фірмою, корпорацією, організацією) і т. п. ;

– за характером використання результатної інформації: інформаційно-пошукові, призначені для збору, зберігання і видачі інформації за запитом користувача; інформаційно-радіть, пропонують користувачеві певні рекомендації для прийняття рішень (системи підтримки прийняття рішень); інформаційно-керуючі, результатная інформація яких безпосередньо бере участь у формуванні керуючих впливів.

Розвиток комп'ютерної інформаційної технології нерозривно пов'язаний з розвитком інформаційних систем. Вони використовуються в економіці для автоматизованого (людиномашинного) розв'язування економічних задач. Для того, щоб розв'язати будь-яку задачу з допомогою комп'ютера необхідно спочатку створити інформаційне забезпечення (забезпечити розрахунки потрібними даними) і математичне забезпечення (створити математичну модель розв'язування задачі, за якою складається програма для ЕОМ). Спрощену схему автоматизованого

розв'язування економічної задачі (наприклад, розрахунок оптимальної виробничої програми) зображено на рисунку Б.1. Необхідна для розв'язування інформація може надходити безпосередньо (вхідна інформація) або через систему інформаційного забезпечення, яка може поповнюватися і за рахунок нової інформації. Визначальною особливістю інформаційної системи є те, що вона забезпечує користувачів інформацією з кількох організацій.

У системах обробки інформації головними її компонентами є дані та обчислення. Більшість інформаційних систем управління інформаційними ресурсами в організаціях містять і багато інших компонентів, таких як вимоги, запити, тригери і звіти. І всі вони, зокрема, містять великі описи свого власного змісту в тій чи іншій формі. Ці описи необхідні для інтерпретації і для коректного використання наданої інформації (коли в системі немає повного опису, то передбачається, що користувачі отримують його з іншого джерела).

Для головних компонентів інформації (даних і обчислень) важливе значення має така характеристика, як їх надмірність. Означення надмірності суттєво залежить від одиниці інформації. Коли одиниця вибрана, то надмірність — це просто дублювання однієї й тієї самої одиниці в системі. Важливим у виборі одиниці інформації є її розмір. Вибір занадто малої одиниці призводить до високого рівня незалежності блоків інформації, але водночас і до збільшення накладних витрат затрат на їх підтримку; у разі взяття великої одиниці неможливо виключити численне дублювання підблоків інформації [11-14].

Оскільки багтрекер – це інформаційна система обліку та документообігу, то відповідно він має всі риси такого роду системи.

Інформаційне забезпечення ІС обліку передбачає створення єдиного інформаційного фонду, систематизацію та уніфікацію показників і документів (багів) та розроблення засобів формалізованого опису даних тощо.

Інформаційне забезпечення є важливим елементом автоматизованих інформаційних систем обліку, який призначений для відображення інформації, що характеризує стан керованого об'єкта і є основою для прийняття управлінських рішень.

У процесі розробки інформаційного забезпечення варто визначити:

- склад інформації, що охоплює перелік інформаційних одиниць або сукупностей, достатніх для розв'язання комплексу задач;
- структуру інформації та перетворення її, тобто формування показників документів;
- характеристики руху інформації, тобто обсяг потоків, маршрути, терміни;
- характеристику якості інформації;
- способи перетворення інформації.

Програмне забезпечення та інформаційна технологія, зорієнтованою на кінцевого користувача, ведуться паралельно з організацією інформаційного забезпечення.

Основою інформаційного забезпечення ІС є інформаційна база (ІБ), що застосовується задля повного функціонування ІС. За складом та змістом вона повинна відповідати вимогам тих задач і проектувати ті системи, які розв'язуються на основі цієї ІБ. [15-16]

1.2 Структурні особливості системи

Структуру інформаційної системи складає сукупність окремих її частин – підсистем. Підсистемою є частина системи, що виділена за певною ознакою. Структура будь-якої інформаційної системи може бути представлена як сукупність підсистем, що забезпечують інформаційне, технічне, математичне, програмне, організаційне і правове забезпечення (рис. Б.2).

У цій ситуації інформаційне забезпечення визначається як сукупність єдиної системи класифікації та кодування повідомлень, уніфікованих систем документації, схем інформаційних потоків, які циркулюють в організації, а також методологія побудови баз даних.

Технічне забезпечення – комплекс технічних засобів, що призначені для роботи інформаційної системи, а також відповідна документація на ці засоби й технологічні процеси.

Математичне й програмне забезпечення - сукупність математичних методів, моделей, алгоритмів і програм для реалізації цілей і завдань інформаційної системи, а також нормального функціонування комплексу технічних засобів.

Організаційне забезпечення – сукупність методів і засобів, що регламентують взаємодію працівників з технічними засобами й між собою в процесі розробки й експлуатації інформаційної системи.

Правове забезпечення – сукупність правових норм, які визначають створення, юридичний статус і функціонування інформаційних систем, що регламентують порядок одержання, перетворення й використання відомостей. [17-18]

1.3. Існуючі інструменти розробки ІС

Існує достатньо широкий спектр інструментів розробки інформаційних систем. Оскільки, планується розробка інформаційної системи без особливого застосування мов програмування, то і розглядатимуться альтернативи, які являють собою алгоритмічно-комбіновані системи-конструктори з гнучким інтерфейсом та повністю або частково відкритим кодом для редагування.

WordPress — це система керування вмістом, що має відкритий кодом і, яка через свою простоту в установленні та використанні широко застосовується для створення веб-сайтів. Сфера використання є досить широкою, починаючи від блогів та закінчуючи складними за структурою веб-сайтами. Вбудована система тем і плагінів у поєднанні з вдалою архітектурою дозволяє конструювати на основі WordPress практично будь-які веб-проекти.

Написана система за допомогою мови програмування PHP з використанням бази даних MySQL. Початковий код поширюється на умовах ліцензії GNU General Public License [19].

Від інших систем управління вмістом сайту WordPress відрізняється достатньо простим інтерфейсом. Програмісту це можливо не гарантує всієї повноти зручностей, зате адміністратору веб-ресурсу значно полегшує роботу.

Публікації створюються моментально, сама платформа встановлюється не більше ніж за 5 хвилин.

Уже в другій версії CMS з'явився візуальний редактор. У наступних версіях додалися автозбереження (версія 2.1), можливість вказувати власний пароль при установці і задавати свої фонові зображення, підтримка користувацького меню (версія 3.0), а також формат мобільного застосування (версія 4.1.1.).

WordPress розповсюджується в мережі вільно і безкоштовно, важить кілька мегабайт і завантажується з офіційного сайту розробників. Там же, на сайті, можна знайти російськомовний Кодекс Wordpress, що містить актуальну корисну інформацію по роботі з системою для новачків і досвідчених розробників. Нижче наведено зображення адміністративної панелі CMS WordPress (рис. Б.3) [20].

Joomla! – це система керування вмістом (CMS), яка написана на мовах PHP і JavaScript і використовує в якості сховища бази даних СУБД MySQL або інші стандартні реляційні СУБД. Система є вільним програмним забезпеченням, поширюваним під ліцензією GNU GPL.

Система управління вмістом Joomla! є, по суті, відгалуженням широко відомої CMS Mambo, яка утворилася через те, що команда незалежних розробників відокремилася від проекту Mambo через незгоду в економічній політиці і 16 вересня 2005 року в світ вийшла перша версія Joomla!, що по своїй суті є перейменованою Mambo 4.5.2.3 і включає в себе виправлення знайдених на той момент помилок і вразливостей.

До літа 2008 року по числу щоденних завантажень Joomla! посіла друге місце після WordPress зі значним відривом від інших подібних систем.

CMS Joomla! включає в себе мінімальний набір інструментів при початковій установці, який доповнюється в міру необхідності. Це знижує захаращення адміністративної панелі непотрібними елементами, а також знижує навантаження на сервер і економить місце на хостингу.

Joomla! надає можливість відображати інтерфейс фронтальної та адміністративної частини будь-якою мовою. Каталог розширень містить широкий спектр мовних пакетів, які встановлюються засобами адміністрування. Доступні

пакети українською, російською, білоруською та ще деякими мовами пострадянського простору.

Структурно система являє собою ядро, до якого приєднуються компоненти. Компоненти можуть складатися по функціональному призначенню з модуля, плагіна і/або шаблону:

- модулі – це елементи, які відображаються на сторінці користувачів під виглядом окремих блоків.
- плагіни – це елементи, що відображаються на адміністративній панелі інструментів і надають додаткову функціональність розроблюваній системі.
- Шаблоном є елемент, який відповідає за уніфіковане оформлення всього сайту.

До складу ядра Joomla! входить:

- блок (функціональність визначається HTML-кодом всередині нього) і його різновиди з широкими можливостями керування з адміністративної панелі: банери (модулі для розміщення реклами), контакти (блок для розміщення контактної інформації та форми зворотного зв'язку), стрічка новин (RSS підписка);
- матеріал з мітками (ключові слова) і категоріями (папками), що слугують навігацією користувачів, редактором для зручного оформлення матеріалу.

Не входять до ядра вільні сторонні розширення:

- плагіни для резервного копіювання (backup) і відновлення, наприклад, Akeeba Backup.

Створення власного розширення здійснюється за допомогою PHP з використанням для взаємодії API Joomla! [21].

Wix – один з найавторитетніших конструкторів сайтів. Сервіс, який вже 12 років задає тенденції розвитку всієї ніші.

Wix найкраще підходить для створення різного роду сайтів-візиток — складних з дизайнерської і функціональної точок зору: портфоліо, бізнес-сайт, промо-сторінка або Лендінгем. Дуже вдалим виходять блоги - їх легко робити, зручно вести. eCommerce складова теж розвинена, особливо з урахуванням додатків і фірмових ноу-хау на зразок Wix Code.

Область застосування Wix можна охарактеризувати трьома факторами:

– Сайти з невеликою кількістю сторінок, оскільки дизайн кожної потрібно формувати вручну.

– Блоги, форуми, магазинні вітрини та інші типи сторінок, які динамічно розширюються ніяк не ускладнювати роботу над сайтом. Впливає обсяг тільки статичних сторінок на кшталт контактів, переваг, умов співпраці, історій компаній, виробництва та іншого.

– Необхідність глибокої і тонкої настройки дизайну, а також введення великої кількості дрібних функціональних елементів.

Також слід зазначити, що магазин додатків Wix містить багатий асортимент додаткового функціоналу (по суті, модулів, якщо брати за аналогією з іншими системами). Кілька категорій корисних додатків (всього понад 250 найменувань) допоможуть досягти практично будь-якої розумної мети.

Wix ідеально підходить для створення маленьких і середніх за обсягом сайтів зі складним дизайном і функціональністю. Цей конструктор може бути простим, поверхневим, якщо потрібно: взяли готовий шаблон, замінили демо-контент, поправили кольору, отримали сайт. З тим же успіхом його можна використовувати для конструювання як завгодно складних сторінок з купою деталей: анімаціями, ефектами, формами для збору і пакування інформації в бази даних, інтерактивними елементами, спливаючими вікнами авторизацій різних етапів допуску до інформації, всілякими опитуваннями, табами, колонками і іншим.

Сукупні можливості Wix складаються з функціоналу редактора, загальних налаштувань, панелі управління, встановлених сторонніх додатків і опцій, які можна впровадити на сайт самостійно за допомогою інструменту Wix Code.

Будь-який розділ редактора містить підказки по роботі з ним. При наведенні курсора на кружечок «і» відобразиться коротке пояснення і посилання на повний варіант FAQ по темі. Абсолютно всі опції і віджети забезпечені поясненнями, просто в деяких потрібно натиснути на знак «?» Для виклику спливаючого вікна з інструкцією. Також можна відразу піти в Центр Підтримки - бази знань Wix з

можливістю пошуку відповідей на запит. Пробігшись за категоріями бази знань очима, ви зможете отримати уявлення про можливості конструктора - назви розділів FAQ чітко перераховують функціонал.

Фірмовий магазин додатків, які можна додати на сайт. Дуже сильна частина Wix. Використання даного розділу в кілька разів розширює стоковий функціонал платформи. Тут міститься величезна кількість (на поточний момент 252) додатків різної спрямованості: форми, робота з соцмережами, галереї, додатки для магазину і маркетингу. Можна додати інтерактивний календар, наприклад, інструменти для Email-маркетингу, опитування, Dropbox, eCommerce-примочки, SEO-інструменти та інші корисності в допомогу стандартного функціоналу конструктора. Всі додатки відсортовані за категоріями. Більшість з них розроблені командою Wix, інші, як правило, є інтеграцією можливостей сторонніх сервісів. Приблизно половина додатків зовсім безкоштовні, а інші мають як безкоштовний варіант використання, так і платний з додатковими можливостями. [22]

1.4 Багтрекери-аналоги

Mantis Bug Tracker (MantisBT) — це безкоштовна система відстеження, у якої висхідний код є відкритим. Розповсюджується за ліцензією GNU General Public License 2. Система створена з метою задоволення загальних потреб відстеження запитів (англ. issue management system), управління проектами, та, найчастіше, для відстеження помилок в ПЗ. Інтерфейс користувача — веб-сайт.

Назва та логотип ІС являють собою англomовний каламбур. Mantis (укр. богомол) відомий, зокрема, тим, що відстежує та їсть інших комах (англ. bug). А слово «bug» також загально відоме як помилка в програмному забезпеченні.

Mantis – це напевно, це найпоширеніший представник систем стеження за багами. Він написаний на мові PHP. Його не можна назвати ідеальним багтрекером, однак він здатний вирішувати всі основні завдання, які від нього потрібні.

Тестувальник програмного забезпечення змушений працювати в даному багтрекер безпосередньо за допомогою браузера. Користувачі даного продукту постійно нарікають на проблеми з Unicode. Загалом, даний продукт ще вимагає особливої доопрацювання.

Jira. Багтрекер написаний на мові Java. Відображає хід виконання проектів, є зручні посилання, за допомогою яких можна контролювати звіти і поточні завдання. За допомогою даної системи можна створювати проекти через e-mail. Існує можливість імпорту звітів в Excel, а також можливість Wiki-форматування. Підтримує інтеграцію з Confluence. Даний багтрекер здатний працювати через захищене з'єднання із застосуванням SSL. Потенційних клієнтів, однак, може відлякувати вартість комерційної ліцензії.

JIRA це продукт, призначений для організації процесу контролю запитів і завдань, що має частину функціональності зазвичай властивої великим і дорогим системам управління проектами.

Ключовими поняттями в JIRA є проекти і завдання. Завдання створюються в проектах, для виконання завдань призначаються виконавці. Завдання можуть бути різного типу і мати підзадачі, завдання можуть бути пов'язаними з іншими завданнями. Статус завдань змінюється в процесі їх виконання.

Ви можете організувати контроль розробки проектів, роздавши завдання виконавцям, ви можете визначити свій власний метод руху завдань - від створення до виконання і контролю результатів, конфігурувати правила повідомлення про події всіх учасників процесу, управляти правами доступу користувачів і робити багато чого іншого.

JIRA приносить великий ефект будь-якої організації, діяльність якої можна інтерпретувати як виконання будь-яких проектів і завдань, що мають тематичні і тимчасові рамки.

Головна перевага цього продукту в його ні з чим незрівнянну здатності налаштувати під ваші потреби.

Наприклад, у фінансовій сфері, ви можете організувати процес оформлення кредиту, від заявки, до введення необхідних даних, до прийняття рішення і так далі.

У сфері державного управління - ви можете створити завдання, визначити терміни виконання, приєднати документи, організувати процес проходження завдання між співробітниками і проконтролювати результат.

Ключовими поняттями в JIRA є проекти і завдання. Проекти служать для групування завдань. Завдання створюються в проектах, для виконання завдань призначаються виконавці. Завдання можуть бути різного типу і мати підзадачі, завдання можуть бути пов'язаними з іншими завданнями. Статус завдань змінюється в процесі їх виконання.

У JIRA багато понять пов'язані з певними проектами. Проект пов'язаного з ним лідера проекту, URL сайту проекту, схему розсилки нотифікацій, схему контролю доступу - дуже гнучкий механізм контролю доступу користувача до завдань проекту.

Кожному проекту можна зіставити свою схему руху завдання (документообіг). Кожному проекту можна порівняти свій вигляд екрану - "зібрати" потрібне вікно з можливих компонентів.

Також, JIRA формує звіти по кожному проекту.

Завдання створюються в проектах. Завдання мають типи, наприклад: Завдання, Помилка, Нова ідея. Можна створювати і свої типи завдань. При описі кожного типу завдання є можливість управління набором полів.

JIRA дозволяє відшукувати завдання за всіма критеріями і по призначених для користувача полів, створювати фільтри, які можна зберегти і використовувати знову, а також зробити загальнодоступними і організувати автоматичну розсилку результатів роботи фільтрів членам робочої групи.

Для організації роботи з користувачами JIRA має групи користувачів і ролі. JIRA має систему контролю доступу користувачів до проектів, завдань і функцій

JIRA має систему контролю доступу користувачів до проектів, завдань і функцій, засновану на членство користувачів в групах і ролях.

Так, для кожного проекту, є можливість управління доступом кожної групи користувачів до кожної дії. Також, є можливість сформувати набір допусків в "роль".

Типовий найпростіший поділ ролей в JIRA включає в себе такі ролі:

- адміністратора
- Керівник проекту
- Співробітника що працює над проектом
- інші співробітники

Але можливості JIRA цим не обмежуються, можливе створення спеціальних ролей, наприклад таких, яким можливо тільки читання завдань одного типу але неможливо іншого.

Завдання JIRA в кожен момент часу мають певний статус. Можливі дії з завданнями, які мають той чи інший статус, визначається вбудованою системою управління рухом завдань. Тут наведена схема найпростішого опису руху завдання.

У JIRA є можливість створення такої складної схеми руху завдання, яка потрібна. Схема руху завдання може бути своя для кожного відділу, проекту, типу завдання.

Схема руху редагується вбудованим редактором. Редагуючи рух завдань, створюючи нові статуси завдань (події) і визначаючи можливі дії, можна організувати будь-яку роботу.

Рух завдання можна зробити залежним від умов, застосовувати логіку І / АБО, виконувати визначені дії на кожному етапі руху завдання.

Завдання JIRA в кожен момент часу мають певний статус. Користувачі інформуються по e-mail в разі будь-яких дій із завданнями, для цього служить налаштовується система нотифікації користувачів. Спільно з системою управління рухом завдання і налаштованим розсилаються фільтрами це дозволяє дуже ефективно інформувати всіх зацікавлених осіб про хід виконання завдання.

Bugzilla. Безкоштовний багтрекер, розроблений Mozilla Organization. Написаний на мові Perl. Свого роду еталон, з яким порівнюються всі інші системи стеження за багами з подібною функціональністю. Забезпечує автоматизацію

роботи з документацією, підтримує тісну інтеграцію з системою електронної пошти. Відсутня інтеграція з SVN. Користувачі скаржаться на огидний інтерфейс і не дуже якісний код. Останнє не дивно, так як багтрекер був випущений на ринок ще в 1998 році.

Ключовим поняттям системи є баг («Bug») – деяке завдання, запит, рекламація з приводу помилки в системі, або просто повідомлення, яке потребує зворотного зв'язку, і призначення системи.

Сутність Bug має набір атрибутів, робота з якими – редагування і запити – є основними сценаріями використання Bugzilla.

Основні структурні атрибути є такими:

- Product – основний атрибут, що задає структуру. Кожен «Product» складається з набору компонентів. Можна включити класифікацію продуктів – додатковий підрозділ продуктів на групи.

- Component – додаткова структурна класифікація. Залежно від обраного компонента, баг може мати різний набір прапорів.

Серед атрибутів життєвого циклу слід виділити такі:

Status – основний атрибут, який визначає поточний стан бага, тобто міру його активності – від самого «початкового» стану, коли він навіть не підтверджений, як баг, до благополучного завершення, коли баг виправлений / вирішене, що підтверджено Службою Якості. Набір станів залежить від конкретної інсталяції та настройки Bugzilla, однак, стандартний набір: UNCONFIRMED («Не підтверджено»);

- NEW («Новий») – тільки що зареєстрований або перевірений;

- ASSIGNED («Акцептований») – користувач, вказаний в «Assigned-to», підтвердив свою відповідальність за цей баг. Баг може бути «перенаправлення» іншій особі, і знову стати «NEW»;

- REOPENED – баг вже був одного разу вирішено, однак, рішення було або неправильними, або неостаточним.

Resolution - цей атрибут має сенс тільки для багів в станах «RESOLVED», «VERIFIED», «CLOSED». Також, набір «резолуцій» можна налаштовувати, але стандартний набір наступний:

- FIXED («Вирішено») – означає, що завдання виконано або баг виправлений.
- INVALID («Некоректно») – неправильна або некоректна постановка, яка не має сенсу.
- WONTFIX («Проблема є, але вирішувати її не будемо»).
- DUPLICATE («Дубль») – описана проблема вже реєструвалася в певному баге.
- WORKSFORME («А у мене працює...») – не вдалося відтворити описану проблему ні емуляцією сценарію, ні аналізом коду.
- MOVED – Проблема перенесена в іншу систему-«tracker».
- LATER – (Також можна використовувати) Завдання зафіксована, але її рішення відкладається на невизначений термін.

Trac – це безкоштовна і проста система управління проектами, але помітно спрощує наше життя при належному терпінні. Основою для Trac є SVN репозиторій. Тому якщо ви до сих пір не використовуєте в розробці Subversion, то швидше за все Trac буде для вас нецікавим. Основні функції включають:

- управління проектом (поділ проекту на етапи, контроль виконання, всі зміни по проекту заносяться на тимчасову шкалу);
- тікети (Стандартна функціональність - облік помилок, зауважень, побажань з можливістю фільтрації і занесення відповідно в milestone, roadmap);
- перегляд сховища (досить зручний модуль по перегляду Subversion репозиторію проекту. Дозволяє переглядати вихідний код з урахуванням ревізії, а також змін;
- управління користувачами;
- Wiki.

Варто відмітити, що для розширення функціональності є можливість підключення додаткових модулів, але вони не є безкоштовними.

Redmine – відкритий (безкоштовний) web-серверний додаток для управління проектами та завданнями (в тому числі для відстеження помилок). Поширюється згідно з ліцензією GNU General Public License.

Особливістю сертифікації системи менеджменту IRIS є те, що вимоги стандарту повинні не просто виконуватися, а повинні бути представлені свідоцтва управління процесами проектування і розробки на кожному етапі виконання проекту, при схваленні кожного етапу робіт перед переходом до наступного. Написаний на Ruby Rails. Підтримує близько 30 мов. Інтеграція з SVN, CVS, GIT, Mercurial і ін. Може підтримувати кілька проектів, на відміну від Trac. Існує пакетне редагування, засноване на AJAX. Можна встановлювати різні користувальницькі ролі в проектах. Підтримує настроювання процесів (workflow). З мінусів: обмежені можливості при роботі з файлами і документами, немає прав на окремі типи переходів настроюються процесів, неможливо управляти також правами доступу на рівні окремих полів завдання.

Користувачі є одним з центральних понять предметної області. Модель користувача є основою для ідентифікації і аутентифікації працює з системою персоналу і клієнтів, а також для авторизації їх в різних ролях, проектах і т. д.

Ролі користувачів визначаються гнучкою моделлю визначення прав доступу користувачів. Ролі включають в себе набір привілеїв, що дозволяють розмежовувати доступ до різних функцій системи.

Користувачам призначається роль в кожному проекті, в якому він бере участь, наприклад, «керівник», «розробник» або «спостерігач» та інші. Користувач може мати кілька ролей.

Кожен користувач системи може подивитися завдання, в яких він бере участь як замовник, спостерігач і виконавець. Система кольором виділяє завдання з різними по терміновості параметрами.

Проект є одним з основних понять в предметній області систем управління проектами. Завдяки цій сутності можливо організувати спільну роботу і планування декількох проектів одночасно з розмежуванням доступу різним користувачам. Проекти допускають ієрархічну вкладеність.

Завдання є центральним поняттям всієї системи, що описує якусь задачу, яку необхідно виконати. У кожного завдання в обов'язковому порядку є опис і автор, в обов'язковому порядку завдання прив'язана до трекера.

Для того, щоб додати нове завдання в проект, необхідно перейти на вкладку «Нова задача», вибрати трекер завдання і заповнити обов'язкові (*) і додаткові (в тому числі і призначені для користувача) поля завдання.

Трекери є основною класифікацією, за якою упорядковано завдання в проекті. По суті, в «Redmine» трекери представляють собою аналог підкласів класу «Завдання» та є основою для визначення і сортування різного роду завдань, дозволяючи призначати для кожного їх типу різні поля. Прикладами трекерів є «Проблема», «Поліпшення», «Проектування», «Сервіс» і т.д.

Кожне завдання має статус. Статуси представляють собою окрему сутність з можливістю визначення прав на призначення статусу для різних ролей (наприклад, статус «відхилений» може привласнити тільки керівник).

За відстеження змін параметрів завдань користувачами в системі RM відповідає вкладка: «Дія» та «Змінений параметр». Запис журналу відображає одну дію користувача по редагуванню параметрів завдання і/або додавання коментаря до неї. Тобто служить одночасно інструментом ведення історії завдання і інструментом ведення діалогу. Сутність «Змінений параметр» прив'язана до окремого запису журналу і призначена для зберігання старого і нового значення зміненого користувачем параметра.

Система підтримує облік витраченого часу завдяки сутності «Витрачений час», пов'язаної з користувачами і завданням. Сутність дозволяє зберігати витрачений час, вид діяльності користувача (розробка, проектування, підтримка) і короткий коментар до роботи. Ці дані можуть бути використані, наприклад, для аналізу внеску кожного учасника в проект або для оцінки фактичної трудомісткості і вартості розробки.

Хоча Redmine і є безкоштовним та все ж деякі з модулів доступні лише після оплати [23].

Отже, проаналізувавши всі вище перелічені багтрекер-системи слід визначити, що існує велике різноманіття ІС такого типу з широким функціоналом, більшість із них є умовно безкоштовними, адже для одержання повного функціоналу все ж необхідно платити. Також більшість таких систем не є «чистими» багтрекерами, вони призначені більше для ведення проектів в цілому, а документування дефектів є лише побічним функціоналом. З більшістю аналогів також виникає проблема освоєння інтерфейсу та логіки системи через перенавантаження різноманітністю функціоналу.

2 ПРОЕКТУВАННЯ ІНФОРМАЦІЙНОЇ СИСТЕМИ БАГ-ТРЕКІНГУ

2.1. Опис ключових елементів системи

Проаналізувавши всі вищеперелічені конструктори ІС було обрано саме конструктор сайтів Wix за його гнучкість у налаштуванні та широкий спектр надбудов та модулів.

Основним модулем системи звісно ж буде структурована БД, яка повинна буде містити в собі всю інформацію про користувачів, які зареєстровані в системі та інформацію про проекти і задокументовані дефекти для кожного з них

При розгляді бази даних важливим є виділення місця розробки баз даних в процесі проектування та реалізації інформаційних систем. Зазвичай, процес розробки інформаційної системи складається з чотирьох ключових етапів, які є регламентацією жорсткої послідовності створення інформаційної системи: аналіз, проектування, розробка та впровадження. В нашому випадку, завдяки інтерактивній пристосованості конструктора Wix база даних створює відповідні таблиці і поля автоматично під час наповнення сторінок сайту контентом та налаштування модулів для ефективного функціонування системи. Отже, таким чином ми економимо час на додаткове налаштування БД.

Наступним елементом системи є можливість реєстрації нових та авторизації вже існуючих користувачів – тобто структурована система користувачів. Кожен користувач матиме свій особистий кабінет, в якому можна буде коригувати особисті дані, а також швидко переміщатися по сторінках за допомогою інтерактивного меню[10].

Оскільки ІС повинна бути надійно захищена, адже проекти, які розробляються не повинні бути розголошеними, то буде впроваджено систему потрібного захисту даних.

Першим етапом безпеки є, безпосередньо, сама реєстрація в системі. Другим етапом є те що, зареєструватися в системі зможуть лише користувачі, яких підтвердить адміністратор сайту. Таким чином буде відсіяно можливість

випадкової або умисної реєстрації небажаних користувачів. І останній етап являє собою наявність паролю перед входом на сторінку проектів. Пароль зможе змінювати лише адміністратор сайту. Також слід зазначити, що можливість створювати нові записи також матимуть лише ті користувачі, яким відкрито доступ адміністратором (менеджером проекту).

Не менш важливим елементом системи являється загальний інтерфейс. Відомо, що якість процесу інтерактивної взаємодії користувача із системою (швидкість, зручність, низький рівень втоми) пов'язана з деякими психологічними характеристиками людини, а саме: короткострокова і середньострокова пам'ять, час реакції, якість сприйняття візуальної інформації, то при розробці інтерфейсу необхідно врахувати, що саме інтерфейс є однією з найважливіших частина ІС з точки зору її реклами з метою продажу і з точки зору безпосереднього користувача системи, який буде працювати з нею по декілька годин поспіль. Інтерфейс впливає на характер рішень, які приймає ОПР, він може прискорювати час прийняття рішення та покращувати або погіршувати їх якість;

Однією з основних властивостей інтерфейсу є його адаптованість, яка означає, що інтерфейс повинен бути:

- сумісним з потребами та можливостями користувачів;
- забезпечувати простоту переходу від виконання однієї функції до іншої;
- забезпечувати користувачів на високому рівні вказівками стосовно їх можливих дій, а також генерувати належний зворотний зв'язок на їх запити;
- надавати користувачам можливість відчувати себе повноправними керівниками ситуації при розв'язанні всіх типів задач, тобто, забезпечувати їх всією необхідною інформацією; користувач повинен бути впевненим, що він сам розв'язує поставлену задачу;
- забезпечувати користувача різними, взаємно доповнюючими формами представлення результатів в залежності від типу запиту або від характеру отриманого рішення;
- враховувати особливості користувачів різних рівнів;

Ще однією важливою властивістю є достатність інтерфейсу, яка означає:

- допустимі запити користувача мають бути чіткими і однозначними для користувачів кожного рівня, а також для прикладних задач будь-якого типу;
- реакція системи на всі типи запитів також повинна бути однозначною і зрозумілою і, по мірі можливості, простою.

Дружність інтерфейсу – це максимальна простота його використання і готовність задовольнити запити користувача в повній мірі при розв’язанні визначеного класу задач.

Гнучкість інтерфейсу – це можливість адаптування інтерфейсу до розв’язання конкретної задачі. Якщо задача, яка вирішується є досить складною, то інтерфейс повинен полегшувати формулювання запитів і видавати результати у такій формі, яка швидко і легко сприймається користувачем. Тобто сам інтерфейс повинен буди максимально простим навіть у тому випадку, коли розв’язується дуже складна та об’ємна задача. При цьому простота означає наступне:

- інтерфейс не повинен бути перевантажений деталями щодо представлення розв’язку поставленої задачі, адже користувач може не охопити всіх подробиць (і в цьому, як правило, немає потреби), тобто нічого зайвого, крім того, що необхідно для розуміння результату;
- він не повинен містити зайвих декоративних деталей та стильових оформлень, які відволікають від головної задачі;
- інтерфейс має бути консистентним, тобто, ґрунтуватись на використанні відомих, загальноприйнятих методів, стандартів і засобів представлення інформації;
- в ідеалі процес взаємодії користувача з системою повинен проходити максимально просто і не повинно виникати ніяких труднощів.

Перспективним напрямком створення інтерфейсів користувача вважаються адаптивні інтерфейси. Адаптивним інтерфейсом користувача (AI) називають сукупність програмних та технічних засобів, що дозволяють користувачу ефективно використовувати всі можливості, які надає система, та задаються конкретні налагодження для кожного користувача. Одним з головних критеріїв ефективності функціонування програмних продуктів є максимальне використання

для роботи з ними людських ресурсів. Це пов'язано перш за все з тим, що вартість людських ресурсів для роботи з програмним забезпеченням вже досягла вартості системних компонент, а в багатьох випадках досить суттєво їх перевищує.

Для того, щоб адаптивний інтерфейс міг бути коректно вбудованим в процес діалогу для користувача кожного рівня, він повинен враховувати апіорну інформацію про психофізичні, професійні, особисті характеристики користувача. Ця інформація здобувається шляхом попереднього початкового тестування користувача. Але цієї інформації для коректної роботи AI явно недостатньо. Додаткова інформація про користувача, який взаємодіє в даний момент з системою, повинна бути отримана з аналізу дій користувача безпосередньо в процесі роботи. Цей спосіб отримання знань про користувача і є основою динамічної адаптації.

Адаптивний інтерфейс повинен забезпечувати користувачу полегшений режим взаємодії. Адаптивна система повинна комбінувати в собі особливості адаптивних і адаптованих компонентів.

Можливість адаптації системи до рівня професіоналізму користувача здійснюється за рахунок здатності сприймати і виконувати запити на внутрішній формальній мові, що забезпечує більш швидкий доступ до інформації. Природно-мовний інтерфейс є посередником між людиною і базою даних. Він переводить запити, що надходять природною мовою, у формальне представлення, звертається з ним до бази даних, і представляє результат, використовуючи алгоритми і технології реалізовані на сайті, у вигляді, придатному для перегляду й аналізу (рис. В.1).

Використання ПМ-інтерфейсу найвигідніше у випадку масового користувача (наприклад, відвідувача сайту), а також як інтегровану з інформаційною системою організації можливість прийняття рішення, при якому доступ до даних є штатною можливістю інформаційної системи організації і, таким чином, є універсальним і, можливо, основним способом пошуку й одержання інформації [24-35].

2.2. Розробка загальної структури ІС

На початкових етапах створення ІС необхідно мати уявлення про те, як працює організація, яку ми збираємося автоматизувати. Для опису і візуального представлення необхідно побудувати модель. Саме для побудови функціональної моделі (або моделі процесів) і призначено програмний продукт BPwin.

Найбільш зручною мовою моделювання бізнес-процесів є IDEF0, запропонована більш 20 років тому Дугласом Россом.

Під моделлю в IDEF0 розуміють опис системи (текстове і графічне), яке має дати відповідь на деякі заздалегідь певні питання. Процес моделювання якої-небудь системи в IDEF0 починається з визначення контексту, тобто найбільш абстрактного рівня опису системи в цілому. У контекст входить визначення суб'єкта моделювання, цілі і точки зору на модель. Під суб'єктом розуміється сама система, при цьому необхідно точно встановити, що входить в систему, а що лежить за її межами, іншими словами, ми повинні визначити, що ми будемо в подальшому розглядати як компоненти системи, а що як зовнішній вплив. На визначення суб'єкта системи буде істотно впливати позиція, з якої розглядається система, і мета моделювання - питання, на які побудована модель повинна дати відповідь. IDEF0-модель передбачає наявність чітко сформульованої мети, єдиного суб'єкта моделювання і однієї точки зору [36 с. 23].

Візуальне представлення ІС баг-трекінгу наведено на рисунку В.2.

Як видно на рисунку, основною інформацією яка буде надходити в систему є знайдені неспівпадіння з очікуваним результатом. Всі неспівпадіння будуть вноситися на основі інформації про проект QA-інженерами і назначатися на виконання розробникам. Все це буде відбуватися під контролем менеджера проекту. Ну і в результаті ми отримаємо задокументований баг з усіма необхідними критеріями.

Далі проведемо декомпозицію моделі за допомогою методології DFD.

Діаграми потоків даних (DFD) являються основним засобом моделювання функціональних вимог майбутнього ПЗ. З їх допомогою ці вимоги розбиваються на

функціональні компоненти (процеси) і представляються у вигляді мережі, зв'язаної потоками даних. Основною метою таких засобів є демонстрація кожного процесу, який перетворює свої вхідні дані у вихідні, а також виявляє відносини між цими процесами.

При використанні DFD моделі систему представляють як ієрархію діаграм потоків даних, що описують процес перетворення інформації з моменту введення в систему до видачі користувачеві і цей процес є асинхронним. На кожному наступному рівні ієрархії відбувається уточнення процесів, поки черговий процес не буде визнаний елементарним.

Моделі потоків даних були запропоновані спочатку Е. Йорданом (1975), а потім Ч. Гейном і Т. Сарсоном (1979). На таких моделях засновані класичні методології структурного аналізу і проектування програмного забезпечення. Така ж модель використовується в методології структурного аналізу і проектування SSADM (Structured Systems Analysis and Design Method) прийнятою у Великобританії як національний стандарт розробки ІС.

Основою моделі є поняття зовнішньої сутності, процесу, сховища (накопичувача) даних і потоку даних.

Зовнішня сутність – це матеріальний об'єкт або фізична особа, що виступають в якості приймачів інформації або її джерел, наприклад, замовники, персонал, постачальники, клієнти банк і тому подібне.

Процес – перетворення вхідних потоків даних у вихідні відповідно до певного алгоритму. Кожен процес в системі має свого виконавця, який здійснює дане перетворення, і свій особистий номер, який пов'язаний з виконавцем. Так само як у разі функціональних діаграм фізично перетворення може здійснюватися комп'ютерами, уручну або спеціальними пристроями. На верхніх рівнях ієрархії, доки процеси ще не є визначеними, замість поняття «процес» використовують поняття «система» і «підсистема», які позначають систему в цілому або її функціонально закінчену частину відповідно.

Сховище даних – абстрактний пристрій який призначено для зберігання інформації. Деталізувати тип пристрою і способи розміщення, вилучення і

зберігання для такого пристрою не потрібно. Фізично це може бути база даних, файл, таблиця в оперативній пам'яті, картотека на папері і тому подібне.

Потік даних — процес передачі інформації від джерела до приймача. Фізично процес передачі інформації може відбуватися вручну за участю пристроїв або людей зовні проектованої системи або по кабелях під управлінням програми або програмної системи.

Таким чином, діаграма демонструє як потоки даних, породжені деякими зовнішніми сутностями, трансформуються відповідними процесами (або підсистемами), зберігаються накопичувачами даних і передаються іншій зовнішній суті — приймачам інформації. В результаті ми отримуємо мережеву модель зберігання/обробки інформації [37-39].

На рисунку В.3 наведено декомпозицію моделі за допомогою DFD.

Як видно з рисунку знайдено неспівпадіння тестувальник спочатку вводить в відповідну форму з урахуванням інформації про проект (специфікацій). З форми введені дані потрапляють до БД знайдених багів. З БД всі скомпоновані дані виводяться на сайт за запитом будь-кого з користувачів (тестувальника, розробника чи менеджера проекту). І відповідно ці скомпоновані дані являють собою задокументовані баги, які ми отримуємо на виході. Також слід зазначити, що існує можливість доповнення знайдених невідповідностей після виправлень розробника, адже не завжди баги виправляються з першого разу.

Наступним кроком є розробка макету сайту. Звісно, конструктор сайтів Wix пропонує широкий спектр шаблонів, але все одно нам потрібна певна індивідуальність в настройці інтерфейсу, про який вже було описано нижче. Слід зазначити, що завдяки адаптивності конструктора нам не потрібно турбуватися про вигляд сайту на різних екранах. Та все ж братимемо за основу макету розширення 1366x768 – стандартне розширення офісних ноутбуків.

Отже, першою сторінкою є «Головна» (рис. В.4) На цій сторінці буде мінімальна кількість елементів за для забезпечення простоти, зручності та вишуканості інтерфейсу. Також слід врахувати те, що система буде закритою, тобто, основний масив інформації буде прихованим з міркувань конфіденційності.

Отже, на цій сторінці будуть присутні стандартні елементи сайту: шапка з логотипом, та пунктами меню загального доступу та підвал з контактною інформацією. На місці контенту буде розташовано назва та лозунг компанії і кнопка «Вхід» для авторизації та реєстрації користувачів у впливаючому вікні, макет якого наведено в рисунку В.5.

Як видно з рисунку в вікні буде присутня, безпосередньо, сама форма входу, тобто поля електронної адреси та паролю і кнопка вхід. Також планується можливість швидкої реєстрації через соціальну мережу Facebook та через сервіс Google. Якщо ж це новий користувач, який не має аккаунтів в вище перелічених сервісах, то для нього передбачено гіперпосилання на форму авторизації. Як зазвичай, в верхньому правому куті впливаючого вікна розташована іконка закриття «Х».

Що стосується впливаючого вікна «Реєстрація» то макет його буде схожим до макету вікна «Вхід». Різниця буде лише у назві кнопки, поля будуть ті ж самі, адже решта інформації буде вводиться вже після дозволу адміністратора на реєстрацію нового користувача в особистому кабінеті.

Наступним є макет сторінки «Особистий кабінет», який зображено на рисунку В.6. На сторінці буде відображатися форма для редагування особистих даних та меню навігації, для пришвидшення переходу між сторінками. Цей блок буде відображатися на всіх сторінках, які будуть присутні у меню цього блоку. Розташовуватися він буде у лівому сайдбарі. Планується, що там будуть присутні такі пункти як «Заведені баги», «Мої коментарі», «Особистий кабінет», «Сповідання» та «Настройки». Усі ці сторінки матимуть однакову структуру макету. Буде змінюватися лише контентна частина сторінки. Так, наприклад, на сторінці Сповідань, будуть відображатися сповідання про зміни в заведених багах, а на сторінці «Мої баги» будуть відображатися всі раніше задокументовані дефекти. В «Настойках» же будуть міститися налаштування сповідань та інших функцій ІС.

Слід також зазначити, що після авторизації користувача в шапці сайту з'явиться невеличке меню швидкого доступу та сповідань з зображенням

користувача. Таким чином буде налагоджено ефективне використання елементів сторінки

Наступною буде сторінка «Проекти» (рис. В.7). На сторінці «Проекти» буде відображатися список вже існуючих проектів з ключовою інформацією про них та можливістю їх відкриття. Також буде присутнє поле пошуку по вже задокументованим дефектам і кнопка додавання нового проекту, яка буде відображатися лише для користувачів з правами менеджера проекту/адміністратора. Додавання ж проекту буде відбуватися у впливаючому вікні, яке буде з'являтися з правого боку сторінки.

Аналогічно буде виглядати і сторінка «Баги». Відмінність полягатиме в тому, що замість списку проектів буде відображатися список задокументованих багів і функціонал кнопки «Додати» буде направлено на додавання нового бага.

Додавання багів буде відбуватися на сторінці «Баги», вигляд якої наведено на рисунку В.8.

Основним компонентом на цій сторінці буде форма яка буде складатися з полів «Загальний опис бага» (Summary) і «Кроки відтворення». Також буде присутній функціонал додавання мультимедійних файлів (наприклад скріншотів) і відповідно кнопка створення багу.

Сторінка вже створеного багу буде відрізнятися лише тим, що форма буде вже заповнена. Буде присутня можливість її редагування/видалення. Також внизу під формою буде присутній блок коментарів, який слугуватиме полем для комунікацій команді з приводу проблеми.

Отже, саме такою буде структура ІС баг-трекінгу. Звісно будуть додані ще сторінки інформаційного характеру такі як: «Про багтрекер», «Контакти» та «Колеги», але їх наповнення буде приведено до загального вигляду з використанням всіх наведених вище стилів, оскільки ці сторінки являються другорядними в самій ІС.

3 РЕАЛІЗАЦІЯ ПРОТОТИПУ ІНФОРМАЦІЙНОЇ СИСТЕМИ БАГ-ТРЕКІНГУ

3.1 Опис основних процедур та залучених модулів у розробці ІС

Перш ніж розпочати створення ІС необхідно пройти реєстрацію на сайті Wix.com. Після реєстрації сервіс пропонує обрати шаблон за певними тематиками. Мною було обрано шаблон з розділу «Бізнес» під назвою «Персональний коуч», оскільки саме цей шаблон якнайкраще підходить під вищеописані макети.

Вже безпосередньо в редакторі сайту було налаштовано сторінку «Головна». На сторінку було додано весь необхідний контент. До кнопки «Вхід» було прив'язано модуль авторизації користувачів. Модуль називається «Користувач». В ньому присутній функціонал налаштування вікна авторизації/реєстрації. Саме тут була ввімкнена функція підтвердження нових зареєстрованих користувачів адміністратором сайту (рис. Г.1). Суть функціоналу полягає у тому, що адміністратор отримує лист-сповіщення на заздалегідь указану адресу, в якому міститься вказана при реєстрації електронна адреса і кнопка «Розглянути запит» (рис. Г.2) [40-41].

Модуль «Користувачі» також автоматично генерує сторінку «Особистий кабінет» та прилягаючі до неї сторінки (наприклад «Налаштування»). Таким чином було утворено навігаційне меню в особистому кабінеті. Також варто зазначити, що цей модуль дає можливість створення іконки з випадаючим списком швидкої навігації користувача, про яку було вказано вище в описі макету.

Після створення всіх необхідних сторінок, деякі з них були обмежені в доступі, який відкривається тільки авторизованим користувачам, таке обмеження було встановлено в настройках відповідних сторінок (рис. Г.3). До сторінки «Проекти» також було додано додаткову міру безпеки – пароль, який необхідно вводити перед кожним переходом на сторінку (рис. Г.4). Це звісно буде забирати час працівників, але все ж зможе захистити ІС від небажаних несанкціонованих доступів.

Весь функціонал створення проектів та багів, їх редагування та коментування було впроваджено за допомогою надбудови під назвою «Wix Forum». Звісно, було виконано певне коригування надбудови під потреби нашої ІС, адже основне призначення цього модулю є саме форуми. Модуль було додано з магазину Wix.

В модулі були налаштовані права доступу лише зареєстрованим користувача, встановлені всі стилі, які відповідають загальному оформленню всіх сторінок, та виконана настройка всіх полів, які необхідні для створення проекту та багу.

Після додавання всіх необхідних модулів та форм було налаштовано систему сповіщень, яка регулюється у модулі «Користувач». Було вирішено додати сповіщення про кожен новостворений баг для всіх користувачів, адже для коректної розробки проекту необхідно щоб всі учасники знали основні помилки та неточності програми чи сайту, що розробляється. Також система сповіщень допоможе уникнути дублікатів – багів з однаковою описаною проблемою. Слід зазначити, що сповіщення також були налаштовані і на додавання коментарів, щоб користувачі своєчасно отримували інформацію про зміни, які відбулися з багом.

Отже, саме за допомогою вищеперелічених модулів була налаштована загальна система баг-трекінгу. Що стосується бази даних, то у нашому випадку всі поля в ній генеруються автоматично, та заповнюються по мірі надходження інформації. Вивід інформації ж відбувається за допомогою модулів, які посилають запит до БД і виводять візуально сприятливий результат.

3.2. Опис послідовного алгоритму роботи багтрекера.

Користувач починає свій шлях з основної сторінки, де вводить свої дані для авторизації і після підтвердження адміністратором авторизується в системі. Скріншот головної сторінки та вікна авторизації наведено на рисунках Г.5-Г.6. З рисунків видно, що сторінка і вікно повністю відповідає вказаному вище макету.

Після логіну користувач перенаправляється до сторінки «Особистий кабінет» (рис. Г.7). На сторінці у користувача є можливість редагування особистих даних. Ця форма розташована в контентній частині сторінки. Також зліва в сайдбарі розташовано меню швидкої навігації з наступними пунктами: «Заведені баги» (з указаною кількістю біля пункту), «Мої коментарі», «Особистий кабінет», «Сповідання» та «Налаштування»

Якщо користувач перейде через меню швидкої навігації на сторінку «Заведені баги» то він отримає представлення всіх заведених багів ним багів (рис Г.8) Через цю сторінку також можна перейти до безпосередньо редагування багу.

Якщо ж користувач перейде на сторінку «Мої коментарі» то там аналогічно до попередньої сторінки будуть відображатися всі коментарі користувача, та його колег, які залишили коментар для багів які створив користувач (рис Г.9)

На сторінці «Сповідання» відображаються всі сповідання які отримав користувач (рис. Г.10). На цю сторінку також можна перейти натиснувши іконку «Дзвоник» у шапці сайту. Слід також зазначити, що біля дзвоника відображається кількість нових сповідань при їх наявності (рис Г.11)

При переході на сторінку «Налаштування» користувач побачить перед собою форму налаштувань системи сповідань (рис Г.12)

Налаштування на сторінці реалізовані за допомогою перемикачів за принципом «так/ні».

Однією з ключових сторінок є сторінка «Проекти». Перед першим переходом на неї користувач буде перенаправлений на форму вводу пароля, який задається адміністратором за для забезпечення більшої безпечності системи (рис Г.13)

Після коректного вводу пароля користувач потрапляє безпосередньо на сторінку «Проекти» (рис. Г.14). Сторінка оформлена згідно макету. Та має поле для пошуку задокументованих багів, кнопку управління проектами, де адміністратор може додати, видалити проект або ж підкоригувати його назву (рис Г.15).

Якщо відкрити один зі створених проектів, користувач побачить перед собою список задокументованих багів з короткою інформацією про кожен з них (рис. Г.16). На сторінці також відображається поле пошуку багів і поряд з ним було впроваджено функціонал фільтрації задокументованих дефектів.

Після натискання кнопки «Створити пост» користувач перенаправляється на сторінку створення нового багу (рис. Г.17) На сторінці присутня форма, в яку користувач вводить короткий опис багу, кроки відтворення і прикріплює медіафайл підтвердження. Після натискання кнопки «Опублікувати» баг додається до списку вже заведених багів, всі користувачі системи отримують сповіщення про його створення. Сторінка задокументованого багу зображена на рисунку Г.18.

На цій сторінці користувач може додати коментарі (рис Г.19), про які буде сповіщено автора багу. Слід також зазначити, що існує можливість вимкнення сповіщень для конкретних багів (рис. Г.20)

Якщо ж користувач захоче перейти на сторінки інформативної наповненості (наприклад «Контакти»), то всі вони знаходяться у шапці сайту.

Отже, саме таким чином здійснюється основний комплекс задач ІС баг-трекінгу.

3.3 Оцінка очікуваного ефекту від впровадження ІС баг-трекінгу

Впровадження інформаційних технологій завжди пов'язано з капітальними вкладеннями як на розробку проекту, так і на придбання техніки чи виконання підготовчих робіт і підготовку кадрів. Тому перед впровадженням повинно бути здійснене економічне обґрунтування доцільності впровадження інформаційної системи. Тобто, що повинна бути визначена ефективність використання інформаційної системи.

Ефективність ІС визначається ступенем їх позитивного впливу на підприємство, що управляється або процес в результаті використання засобів обчислювальної техніки, програмного забезпечення, зміни інформаційних потоків та організаційної структури управління.

Під ефективністю автоматизованого перетворення економічної інформації мається на увазі саме доцільність використання засобів комп'ютерної та організаційної техніки при формуванні, передачі та обробці даних.

Виділять фактичну та розрахункову ефективність. Фактичну визначають за результатами впровадження техноробочого проекту, а розрахункову – на стадії проектування автоматизації інформаційних робіт, тобто розробки техноробочого проекту.

Економічна ефективність від впровадження комп'ютерної та організаційної техніки поділяється на два види: на пряму та непряму.

Під прямою економічною ефективністю мають на увазі економію матеріально-трудових ресурсів і грошових коштів, що отримана в результаті скорочення чисельності управлінського персоналу, фонду заробітної плати, витрачання основних та допоміжних матеріалів внаслідок автоматизації конкретних видів планово-облікових і аналітичних робіт.

Показники непрямої ефективності демонструють якісні результати: отримання нових відомостей для управління, загальне підвищення рівня планування, обліку і аналізу господарської діяльності. Непряма ефективність, зазвичай, є основною частиною ефекту, особливо в невиробничій сфері управління.

Основними джерелами ефективності є:

- Пришвидшення процесу розробки програми чи сайту шляхом автоматизації документообігу
- Покращення комунікацій в команді розробки
- Удосконалення способу оновлення актуальної інформації по дефектах на проекті
- Економія на впровадженні інформаційної системи баг-трекінгу
- Скорочення витрат на оплату робочої сили

Якщо коротко описати перше джерело, то слід сказати, що пришвидшення процесу розробки на пряму залежить від якості прилеглого ПЗ, яким користуються працівники під час розробки. ІС баг-трекінгу є однією з частин цього оточення. В будь-якому випадку, присутність системи відстеження помилок є саме позитивним

явищем, адже досить часто замість багтрекера, компанії, які займаються розробкою невеликої кількості проектів використовують замість багтрекінгових систем

Покращення комунікацій в команді буде обумовлене тим, що за допомогою багтрекера буде налагоджено чіткий комунікаційний зв'язок, між колегами, який буде стосуватися конкретної проблеми (бага). Таким чином буде усунена можливість виникнення дискomuнікацій та надання помилкової інформації.

Під удосконаленням способу оновлення актуальної інформації по дефектах мається на увазі система сповіщень, яка забезпечить постійний контроль над оновленням стану проблеми.

Економія на впровадження ІС проявляється у тому, що для компаній з невеликим обсягом проектів платити за використання дорогих багтрекерів або тасктрекерів просто немає сенсу. Доцільніше використовувати безкоштовний багтрекер, яким і являється наша ІС. Адже гроші будуть витрачені лише на одноразову розробку, та на придбання хостингу, який можна або придбати прямо на сайті Wix або ж перенести всі дані на свій домен на іншому хостингу. На рисунку Д.1 зображені можливі тарифні плани Wix.

Як бачимо, найдешевший тариф коштує 4\$ в місяць, в той час як найдешевший тариф в Jira коштує 7\$ в місяць, але слід зауважити, що в Jira існує обмеження на кількість користувачів і якщо їх кількість буде менше 10, то ціна сягне аж до 10\$ за місяць (рис. Д.2).

Як бачимо логічніше витратити кошти на власну розробку зі своїм персональним дизайном та функціоналом, ніж постійно переплачувати за сторонній продукт.

Скорочення витрат на оплату робочої сили напряму залежить від скорочення часу на розробку, адже зазвичай в ІТ-компаніях використовують погодинну оплату праці, а отже швидкість прямопропорційно впливає на вартість робіт.

Для оцінки ефективності ІС можна розрахувати наступні показники:

а) суму капітальних витрат, які включають витрати на проектування і впровадження проекту, придбання необхідного устаткування і його установку,

програмування комплексу задач та його налагодження. Відповідно капітальні витрати (К) необхідно розраховувати за формулою 3.1:

$$C = V_{\text{проект}} + V_{\text{програма}} + V_{\text{н}} + V_{\text{впр}} \quad (3.1)$$

$V_{\text{проект}}$ – витрати на проектування комплексу задач;

$V_{\text{прогр}}$ – витрати на програмування комплексу задач;

$V_{\text{н}}$ – витрати на програмування в період налагодження ПЗ комплексу задач;

$V_{\text{впр}}$ – витрати на впровадження;

У нашому випадку витрати на проектування будуть дорівнювати 200\$, витрати на програмування – 250\$; витрати на програмування в період налагодження комплексу задач будуть дорівнювати 120\$ і, відповідно, витрати на впровадження будуть дорівнювати 48\$ (по 4\$ на місяць протягом року згідно з тарифним планом Wix). Отже, сума капітальних витрат $C=618\$$

б) суму річної економії, яка розраховується як різниця між річною вартісною оцінкою результатів використання COEI об'єктом управління і приведеними до даного року витратами. При автоматизації задач за умови отримання приблизно однакових кінцевих результатів, які отримано за базовим варіантом, значення економії визначається за формулою 3.2:

$$S = B_{\text{б}} - B_{\text{п}} \quad (3.2)$$

де $B_{\text{б}}$ – приведені до одного року витрати на обробку інформації при базовому варіанті організації обробки;

$B_{\text{п}}$ – приведені до одного року витрати на обробку інформації при впроваджуваному (пропонованому) варіанті організації системи обробки.

Отже, підрахувавши витрати на тарифні плани в Wix та Jira отримаємо результат $S=120-48=72\$$. Таким чином бачимо, що сума річної економії становить аж 72\$ – що є суттєвим показником економії.

Тобто зробивши підрахунки, можна визначити, що лише за допомогою економії на тарифному плані повний цикл розробки ІС окупиться за 8,5 років. З одного боку здається, що це значний період для такої невеликої суми. Та не слід забувати, що ми також отримуємо економію на оплаті праці, що значно пришвидшить повернення коштів, які були витрачені на розробку та впровадження [42-45].

ВИСНОВКИ

Отже, інформаційною системою є організований набір елементів, що збирає, обробляє, передає, зберігає та надає дані. Інформаційна система складається з людей, обладнання, процесів, процедур, даних та операцій. Кожна інформаційна система включає в себе наступні компоненти:

- структура системи;
- функції кожного елемента системи;
- вхід і вихід кожного елемента і системи в цілому;
- мета і обмеження системи та її окремих елементів.

Забезпечення якості (Quality Assurance) являє собою сукупність заходів, охоплюючих абсолютно усі етапи розробки, випуску та експлуатації програмного забезпечення. Це активності на усіх етапах життєвого циклу ПЗ, які вживаються для забезпечення необхідного рівня якості випускаемого продукту.

Багтрекером є інформаційна системи, розроблена, щоб допомогти тестувальникам та програмістам відстежувати історію звітів про баги та взаємодіяти один з одним під час своєї роботи. Він може розглядатись як різновид системи відстеження проблем.

Оскільки, об'єктом дослідження являється галузь QA-інженерії, а однією з основних цілей тестування є виявлення дефектів та надання інформації для прийняття рішень, то предметом дослідження був конкретний процес документування знайдених дефектів. На основі цього було виконано основне завдання дипломної роботи, а саме, розробка ІС баг-трекінгу та досягнуто мети роботи. Відповідно метою є сама система з її функціоналом, яка допоможе систематизувати процеси тестування ПЗ.

Було визначено структуру інформаційної системи, визначено всі її особливості і специфіки у інтерфейсі та функціоналі в цілому. На цій основі, було розроблено макети, та описано всі основні вимоги, які необхідні для створення ІС баг-трекінгу.

Після цього було виконано низку операцій по вибору та настройці модулів безпосередньо у конструкторі сайтів Wix, який було обрано як найбільш підходящий для розробки ІС такого типу. Wix ідеально підходить для створення маленьких і середніх за обсягом сайтів зі складним дизайном і функціональністю. Цей конструктор може бути простим, поверхневим, якщо потрібно: взяли готовий шаблон, замінили демо-контент, поправили кольору, отримали сайт. З тим же успіхом його можна використовувати для конструювання як завгодно складних сторінок з купою деталей: анімаціями, ефектами, формами для збору і пакування інформації в бази даних, інтерактивними елементами, спливаючими вікнами авторизацій різних етапів допуску до інформації, всілякими опитуваннями, табами, колонками і іншим.

Тобто, було виконано настройку модулів відповідно до попередньо визначених вимог та розроблених макетів.

Основні модулі, які використовувалися для розробки є:

- модуль «Користувачі» – за допомогою якого було здійснено настройку авторизації, особистого кабінету, здійснено розробку структури доступів до даних користувачами та налаштовано зв'язки між ролями;

- модуль «Wix Forum» – через який було здійснено настройку документування безпосереднього створення проектів та документування знайдених невідповідностей з очікуваним результатом (багів). Також за допомогою цього модулю було здійснено настройку сповіщень користувачів системи.

Було визначено також ефективність від впровадження ІС баг-трекінгу, яка показала, що ефект буде лише позитивним.

Тобто зробивши підрахунки, було визначено, що лише за допомогою економії на тарифному плані повний цикл розробки ІС окупиться за 8,5 років. З одного боку здається, що це значний період для такої невеликої суми. Та не слід забувати, що ми також отримуємо економію на оплаті праці, що значно пришвидшить повернення коштів, які були витрачені на розробку та впровадження. Тобто також буде досягнуто підвищення продуктивності праці робітників, за

рахунок простоти і неперевантаження зайвим функціоналом. Також таке впровадження допоможе систематизувати життєвий цикл бага в цілому.

Отже, після виконання всіх видів робіт та проходження переддипломної практики можна стверджувати, що кінцевий результат роботи повністю відповідає поставленим задачам, а саме:

- Визначено основні засади у розробці ІС баг-трекінгу.
- Визначено структурні особливості системи.
- Вивчено існуючі інструменти розробки ІС.
- Досліджено існуючі багтрекери-аналоги.
- Сформульовано ключові елементів системи.
- Розроблено загальну структуру ІС.
- Змоделювано основні процедури за допомогою модулів конструктора ІС.
- Описано послідовний алгоритм роботи багтрекера.
- Здійснено оцінку очікуваного ефекту від впровадження ІС баг-трекінгу

Тобто було створено повноцінний прототип, який наповнено всім необхідним функціоналом для документування знайдених невідповідностей з очікуваним результатом у програмах та сайтах.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Луценко Є.С. Багтрекер, як інструмент контролю за процесом тестування// Збірник наукових праць студентів спеціальностей «Інформаційні управляючі системи і технології», «Комп'ютерний еколого-економічний моніторинг» / редкол.: В.С. Пономаренко [ті ін.]. Харків: ХНЕУ, 2011. 308 с.
2. Калбертсон Р., К. Браун, Г. Кобб, Швидке тестування. Вільямс, 2012. 374 с.
3. Louise Tamres, Introducing Software Testing. USA, 2012
4. Judy McKay, Managing the Test People: A Guide to Practical Technical Management. USA, 2008
5. Frank P. Ginac, Customer Oriented Software Quality Assurance. USA, 2012.
6. Michael S. Deutsch, Software quality engineering. USA, 2008.
7. Н. Kan, Metrics and Models in Software Quality Engineering. USA, 1995.
8. Philip B. Crosby, Quality Is Free. USA, 1979.
9. John W. Horch, Practical guide to software quality management. USA, 2011
10. Криспін Л., Грегорі Д., Гнучке тестування: практичне керівництво для тестувальників ПО і гнучких команд. Вільямс, 2010. 464 с.
11. Інформаційні системи та їх роль в управлінні економікою. Одеса, 2017.
URL: <https://www.uzhnu.edu.ua/uk/infocentre/get/6742>
12. Луценко Є.С. Сучасні підходи до розробки програмного забезпечення // Збірник наукових праць студентів спеціальностей «Інформаційні управляючі системи і технології», «Комп'ютерний еколого- економічний моніторинг». Харків ХНЕУ, 2010. 342 с.
13. Ситник В. Основи інформаційних систем. Науковий посібник. Київ, 2015
14. Cheng Hsu. Information Systems. The Connection of People and Resources for Innovation. USA New York, January 2013
15. Терещенко Л.О., Матієнко-Зубенко І. І. Інформаційні системи і технології в обліку. Навч. посіб. — К.: КНЕУ, 2004. 187 с.

16. Сініцин С. В., Налютин Н. Ю., Верифікація програмного Забезпечення. Біном, 2008. – 368 с.
17. K.Rubin. Developing object-oriented software // IBM Object-Oriented. Technology Center, USA, 2014. 22-31 с.
18. Буйницька О.П. Інформаційні технології та технічні засоби навчання. Навч. посіб. – К.: Центр учбової літератури, 2012. 240 с.
19. Офіційний сайт Wordpress. URL: <https://uk.wikipedia.org/wiki/WordPress>
20. Опис CMS Wordpress та оцінки користувачів. URL: <http://wordpress.cmsmagazine.ru>
21. Дэн Рамел. Joomla! для профессионалов = Advanced Joomla!. — М.: «Вильямс», 2014. — 448 с
22. Огляд та відгуки про конструктор сайтів Wix. URL: <https://uguide.ru/konstruktor-sajtov-wix-obzor-otzyvy-primery-sajtov>
23. Багтрекери, як вибрати кращий – Інтернет-стаття. URL: <http://ru.qatestlab.com/knowledge-center/qa-testing-materials/bug-trackers-what-to-choose/>
24. Vikrant Bhateja, Bao Le Nguyen, Nhu Gia Nguyen, Suresh Chandra Satapathy, Dac-Nhuong Le. Information Systems Design and Intelligent Applications. Proceedings of Fourth International Conference. India, 2017
25. Jay M. Gould Input/Output Databases. Uses in Business and Government. Chicago, April 2018
26. Кумскова І. О. База даних. Навчальний посібник. Москва, 2016
27. Бідюк П. І., Гожий О. П., Коршевніюк Л. О. Комп'ютерні системи підтримки прийняття рішень. Київ, 2010. 89-103 с.
28. Larry E. Wood, User Interface Design Bridging the Gap from User Requirements to Design. USA, May 2018
29. Brad A. Myers, Languages for Developing User Interfaces. USA, 2011
30. Бірман І. М. Інтерфейс користувача. Електронний посібник. Росія, Москва. 2013
31. Jon Duckett, Web Design with HTML, CSS, JavaScript and JQuery Set. USA, 2017.

32. David Love, Tkinter GUI Programming by Example
33. Learn to create modern GUIs using Tkinter by building real-world projects in Python. USA, April 2018.
34. Chris Aquino and Todd Gandee, Front-End Web Development: The Big Nerd Ranch Guide. USA, 2016.
35. Andreas Schwarz, Dave Thomas, David Heinemeier Hansson, James Duncan Davidson, Justin Gehtland, Leon Breedt, and Mike Clark, Agile Web Development with Rails. USA, 2010.
36. Створення моделі процесів у BPwin (IDEF0). Росія, Москва, 2014. URL: <http://www.interface.ru/public/caseall/caseall3.htm>
37. August-Wilhelm Scheer, Henrik von Scheel, and Mark von Rosing. The Complete Business Process Handbook. 1st Edition. Body of Knowledge from Process Modeling to BPM, Volume 1. USA, December 6 2014.
38. Matthias Weidlich, Mathias Weske. Business Process Modeling Notation: Second International Workshop, BPMN 2010, Potsdam. Germany, October 13-14, 2010 Proceedings
39. Інтернет стаття: Діаграми потоків даних – Студопедія. URL: <https://studopedia.info/1-113597.html>
40. David Le Blanc, John Viega, and Michael Howard, 24 Deadly Sins of Software Security: Programming Flaws and How to Fix Them, USA, 2009.
41. Geetha Manjunath and Sitaram. Moving to the Cloud: Developing Apps in the New World of Cloud Computing. Canada, December 1, 2011.
42. Capers Jones, Olivier Bonsignour, The Economics of Software Quality. USA, July 14 2011.
43. Ільїн В. В., Управління ефективністю впровадження інформаційних систем. Росія, Москва, 2014
44. Філімонова Є. В., Інформаційні технології в професійній діяльності. Росія, Москва. 2016.
45. Черніков Б. В., Лексикологічний синтез документів в комплексах інформаційних систем. Росія Санкт-Петербург, 2016.

ДОДАТКИ

Додаток А
(обов'язковий)
Анотація магістерської роботи

SUMMARY

Spychak O. V. Development of the information system of bug-tracking. – Qualification master's work. Sumy State University, Sumy, 2018

The work analyzes the state of automation of the documentation of found mismatches (bugs) and formulates the requirements for the information system of bug-tracking. The design of an automated information system was carried out. The main purpose of this study is to develop a prototype of the information system of bug-tracking.

Key words: documentation automation, information system, bug-tracking, software product, enterprise.

АНОТАЦІЯ

Спичак О. В. Розробка інформаційної системи баг-трекінгу. – Кваліфікаційна магістерська робота. Сумський державний університет, Суми, 2018 р.

У роботі досліджено стан автоматизації процесів документування знайдених невідповідностей (багів) та сформовано вимоги до інформаційної системи баг-трекінгу. Проведено проектування автоматизованої інформаційної системи. Основною метою цього дослідження є розробка прототипу інформаційної системи баг-трекінгу.

Ключові слова: автоматизація документування, інформаційна система, багтрекінг, програмний продукт, підприємство.

Додаток Б
(довідковий)

Схематичні зображення принципів роботи ІС та приклад панелі адміністрування

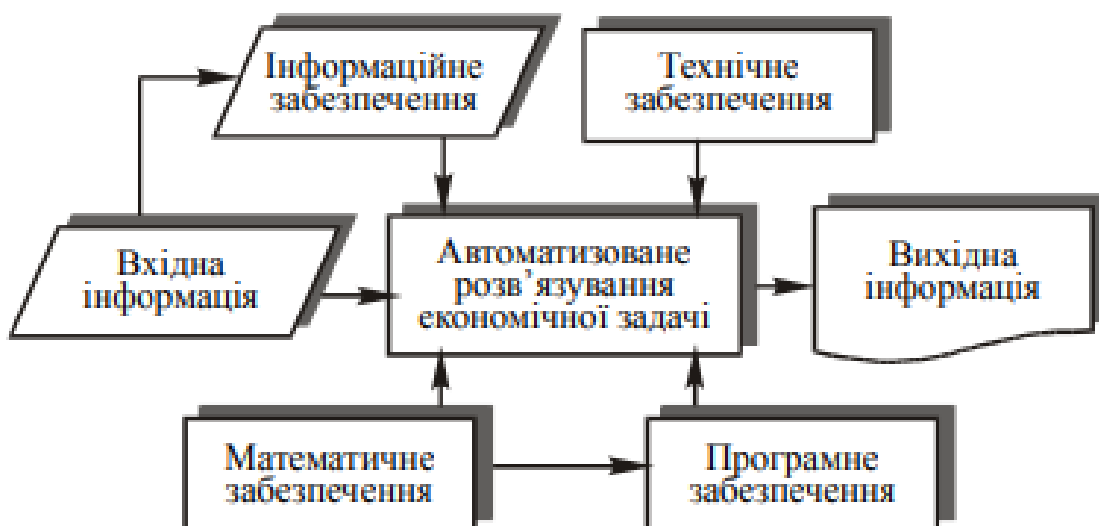


Рисунок Б.1 - Схема автоматизованого розв'язування економічних задач



Рисунок Б.2 – Основні підсистеми ІС

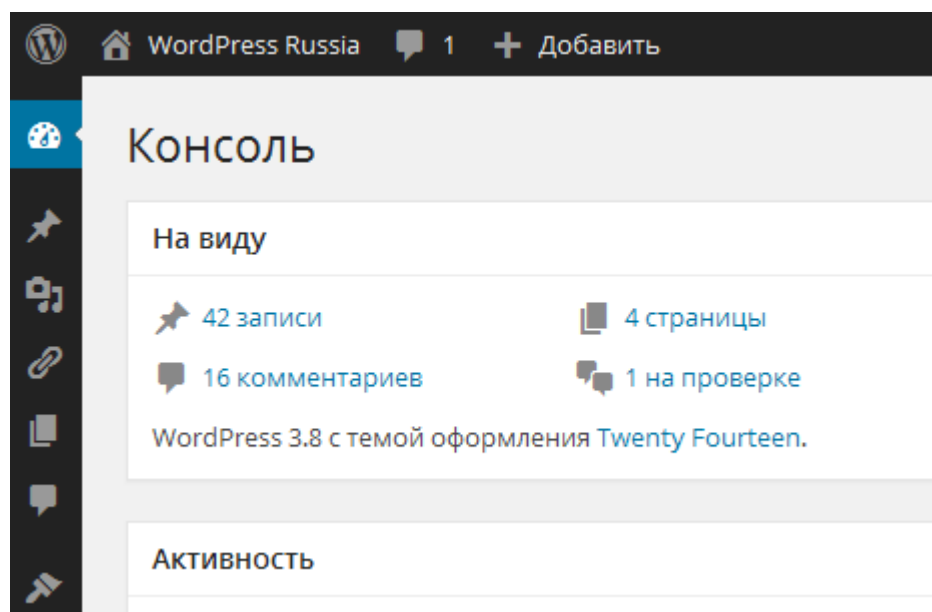


Рисунок Б.3 – Адміністративна панель CMS WordPress

Додаток В
(Довідковий)

Структура там макети розроблюваної інформаційної системи

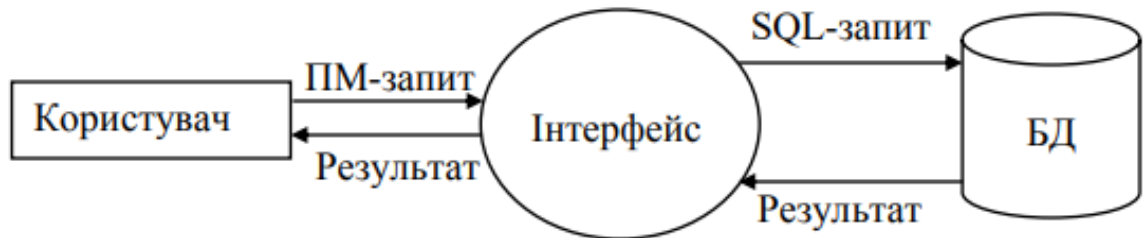


Рисунок В.1. – . Загальна схема потоків даних ПМ-інтерфейсу

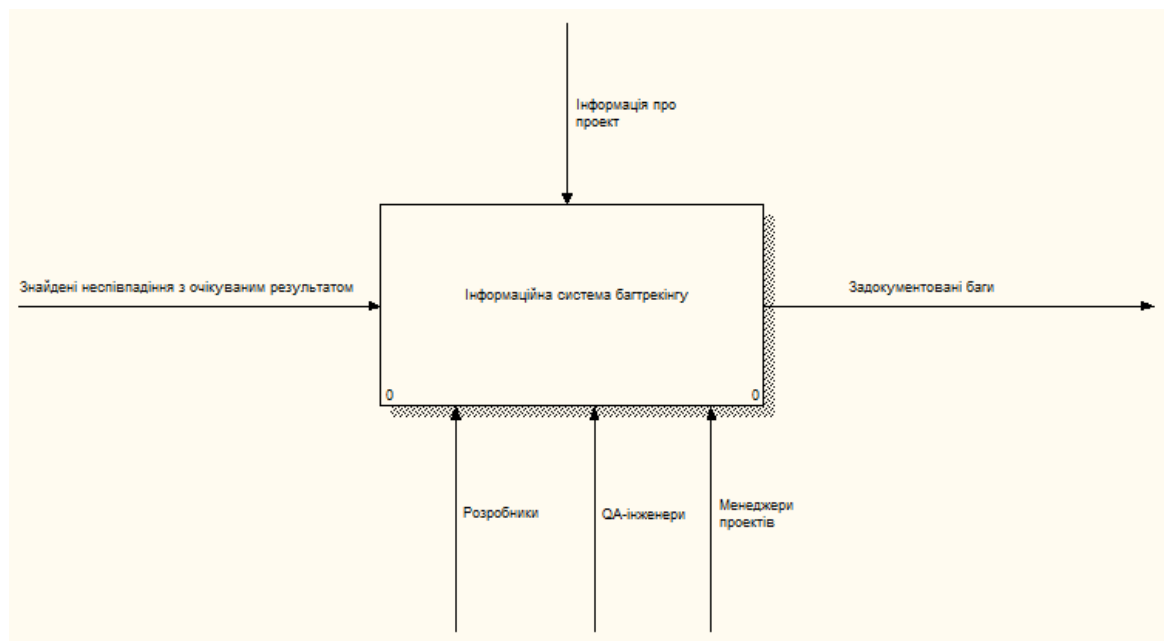


Рисунок В.2 – Представлення ІС баг-трекінгу за допомогою IDEF0

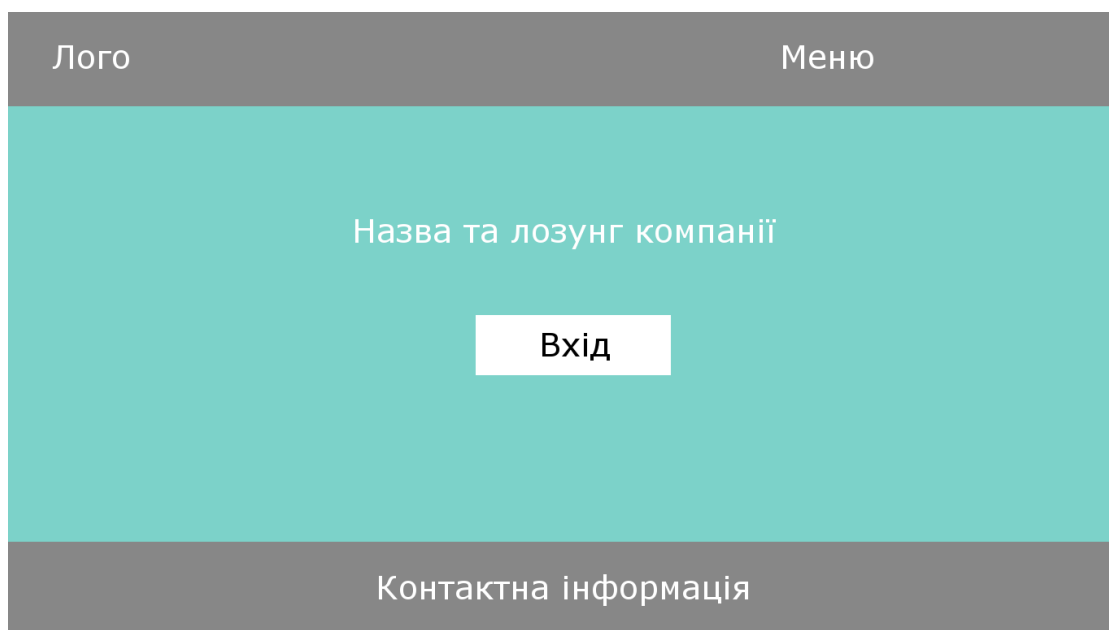


Рисунок В.4 – Макет сторінки «Головна»

X

Вхід

Гіперпосилання на форму реєстрації

Блок авторизації через
Facebook або Google

Електронна адреса

Пароль

Вхід

Рисунок В.5 – Макет впливаючого вікна «Вхід»

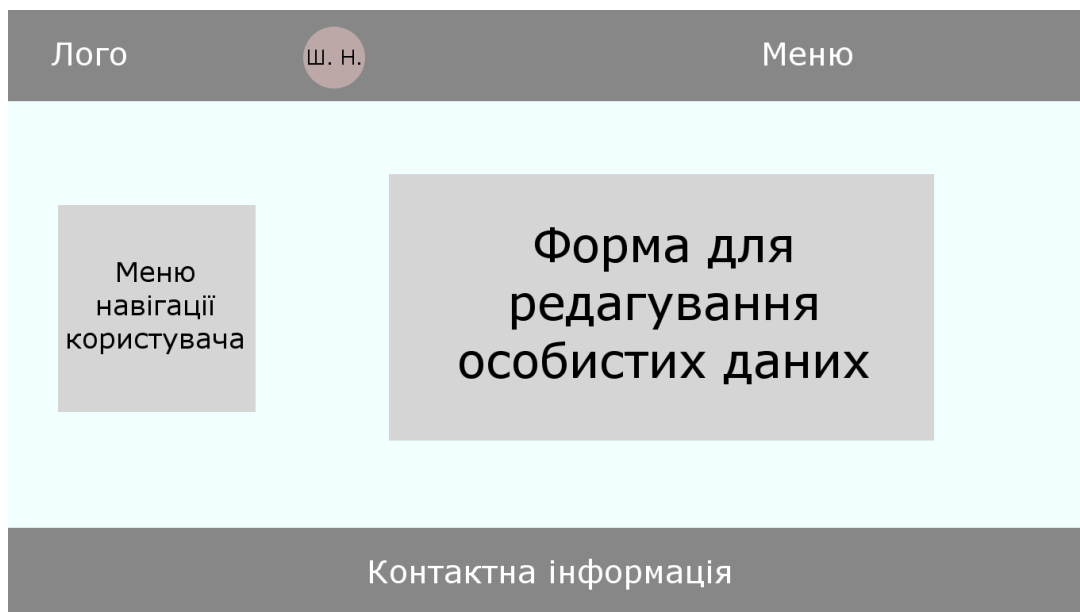


Рисунок В.6 – Макет сторінки «Особистий кабінет»

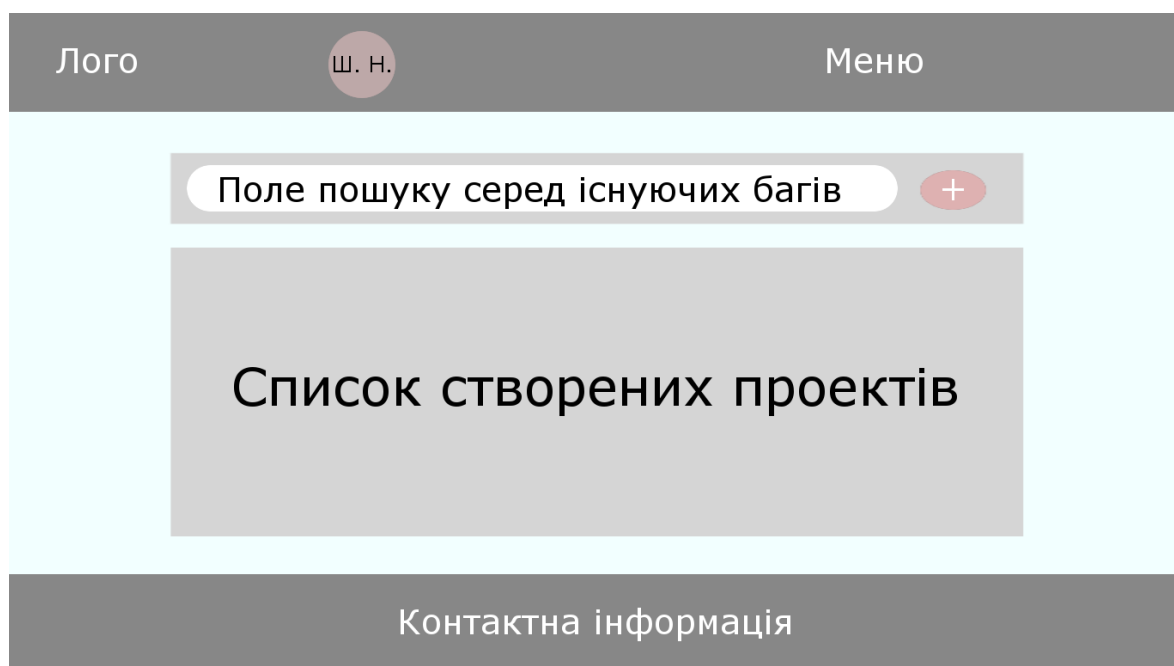


Рисунок В.7 – Макет сторінки «Проекти»

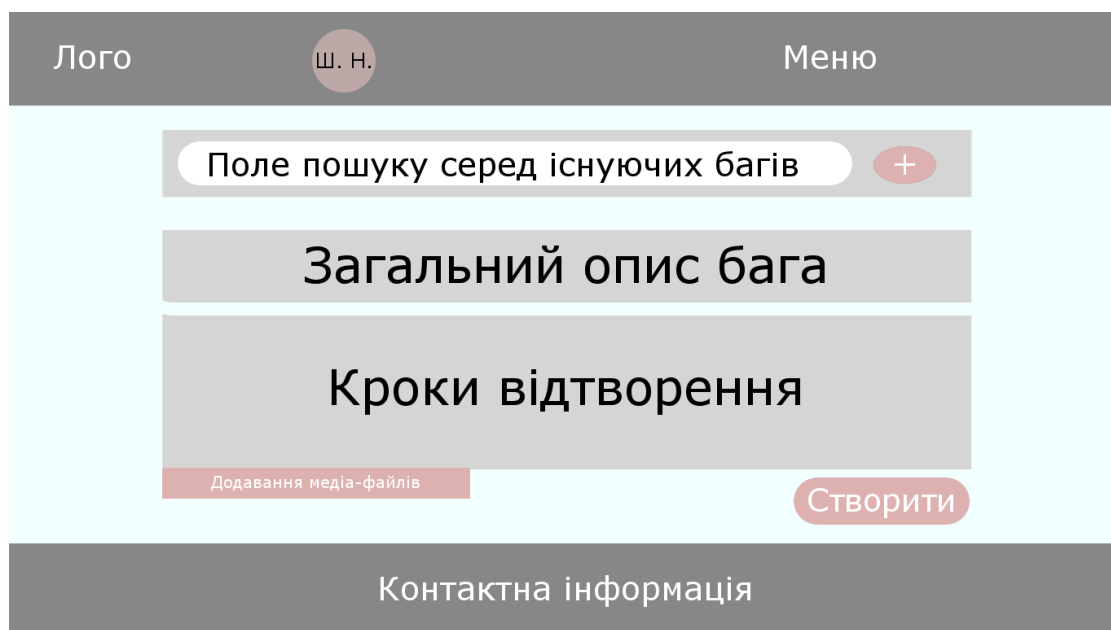


Рисунок В.8 – Макет сторінки «Створити баг»

Додаток Г (Довідковий)

Демонстрація настройки модулів системи та їх функціонування

Настройка регистрации посетителей

Общее Соцсети Правила

Работает только для пользовательского раздела.

Кого регистрировать?

Всех автоматически

Только одобренных мною

На первом месте:

Регистрация новых посетителей

Вход зарегистрированных пользователей

Сохранить

Рисунок Г.1 – Вікно «Настройка реєстрації користувачів»

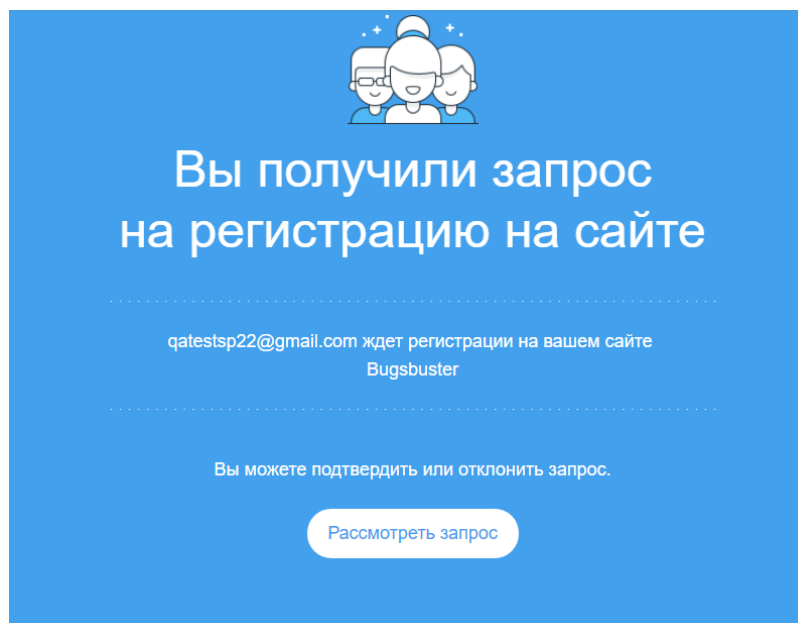


Рисунок Г.2 – Лист-сповіщення про реєстрацію

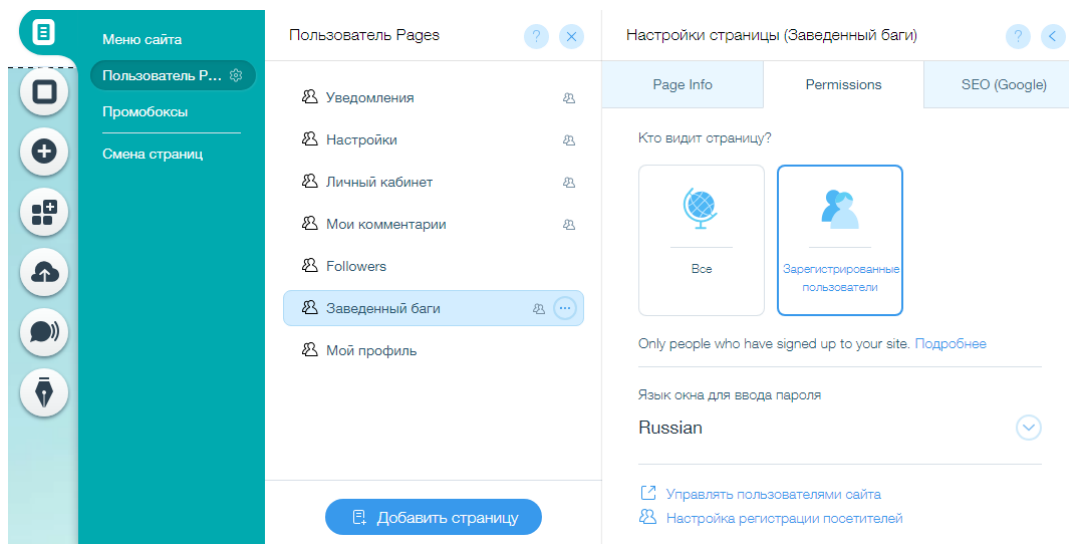


Рисунок Г.3 – Вікно налаштування обмежень у доступі до сторінки

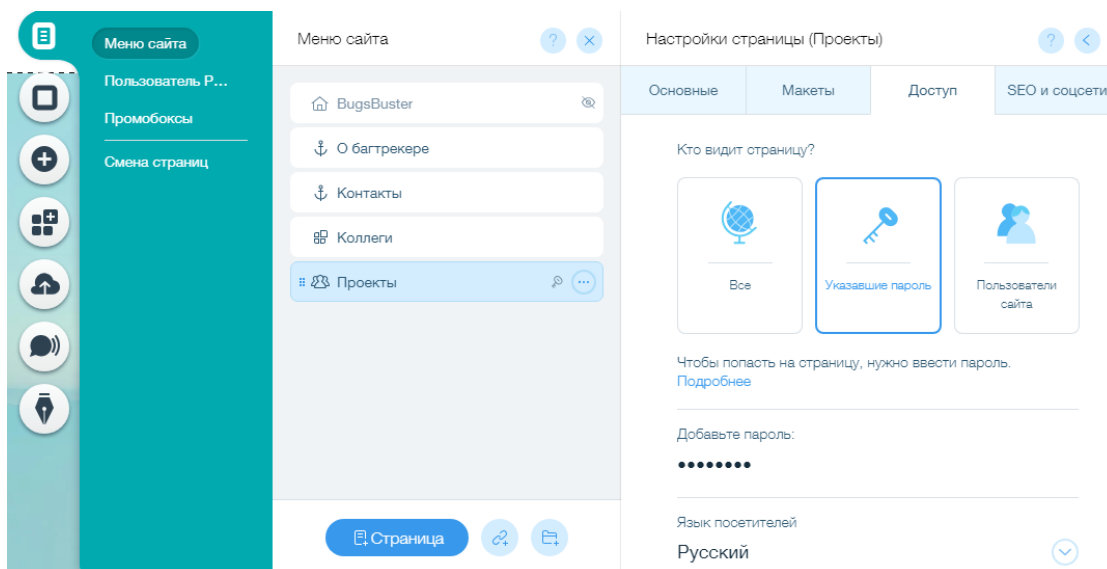


Рисунок Г.4 – Додавання паролю до сторінки «Проекты»

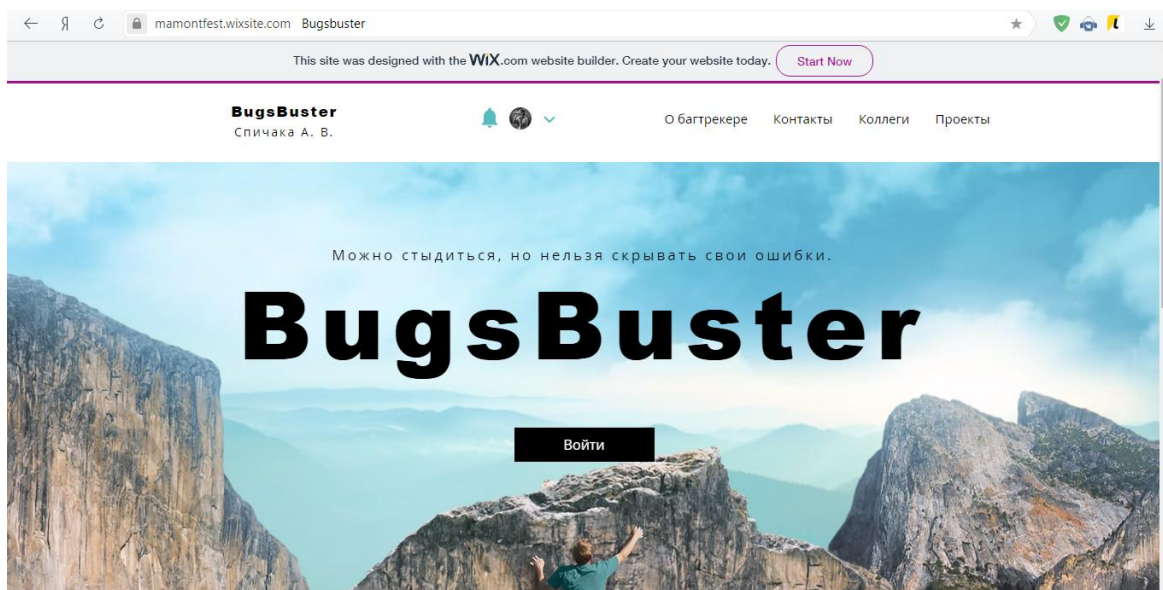


Рисунок Г.5 – Скріншот сторінки «Головна»

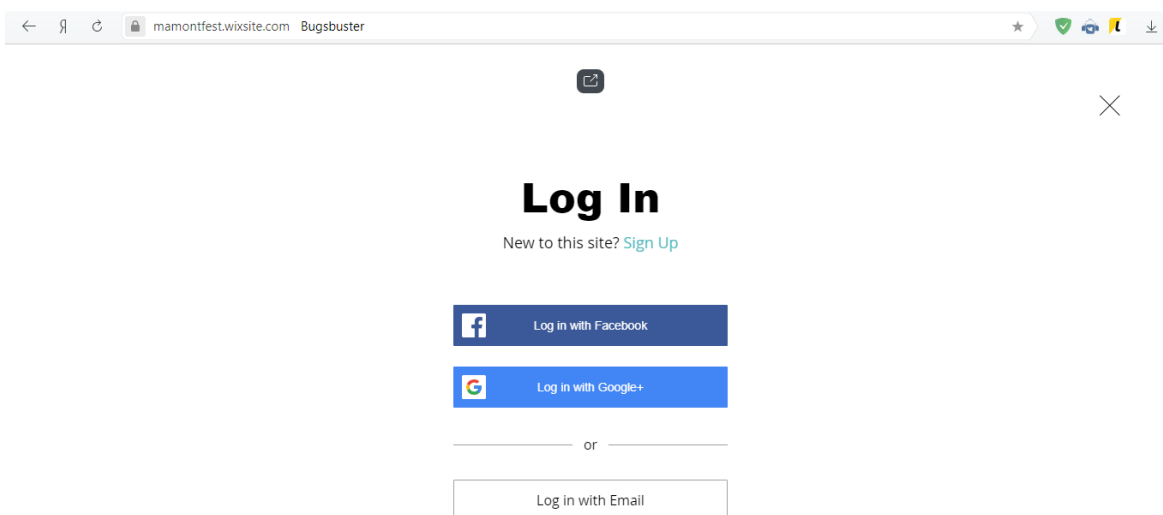


Рисунок Г.6 – Скріншот вікна авторизації

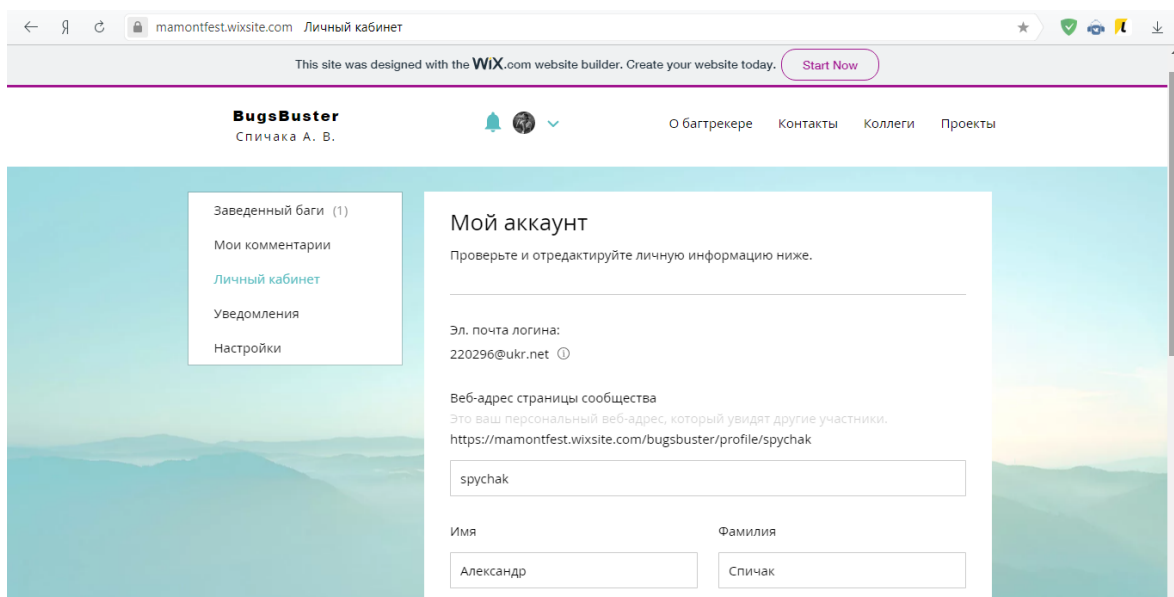


Рисунок Г.7 – Скріншот сторінки «Особистий кабінет»

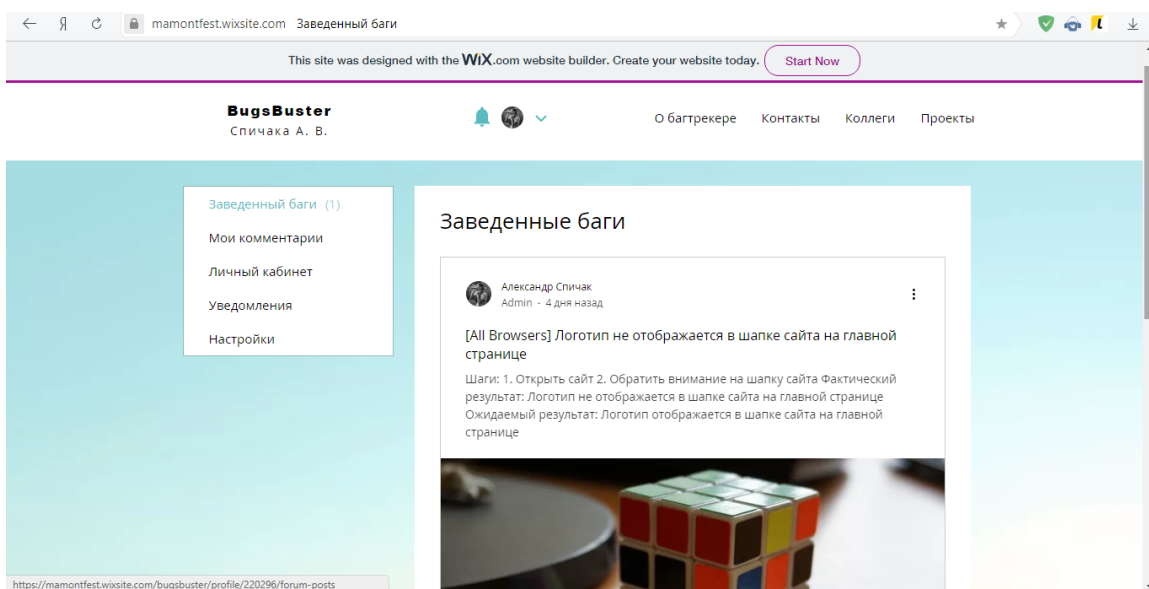


Рисунок Г.8 – Скріншот сторінки «Заведені баги»

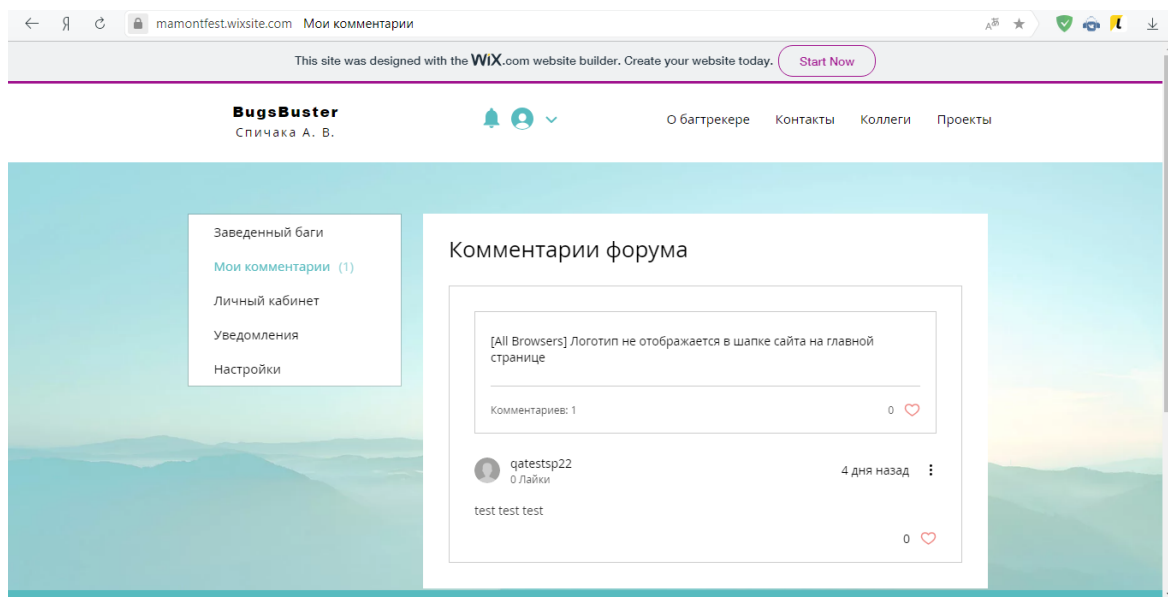


Рисунок Г.9 – Скріншот сторінки «Мої коментарі»

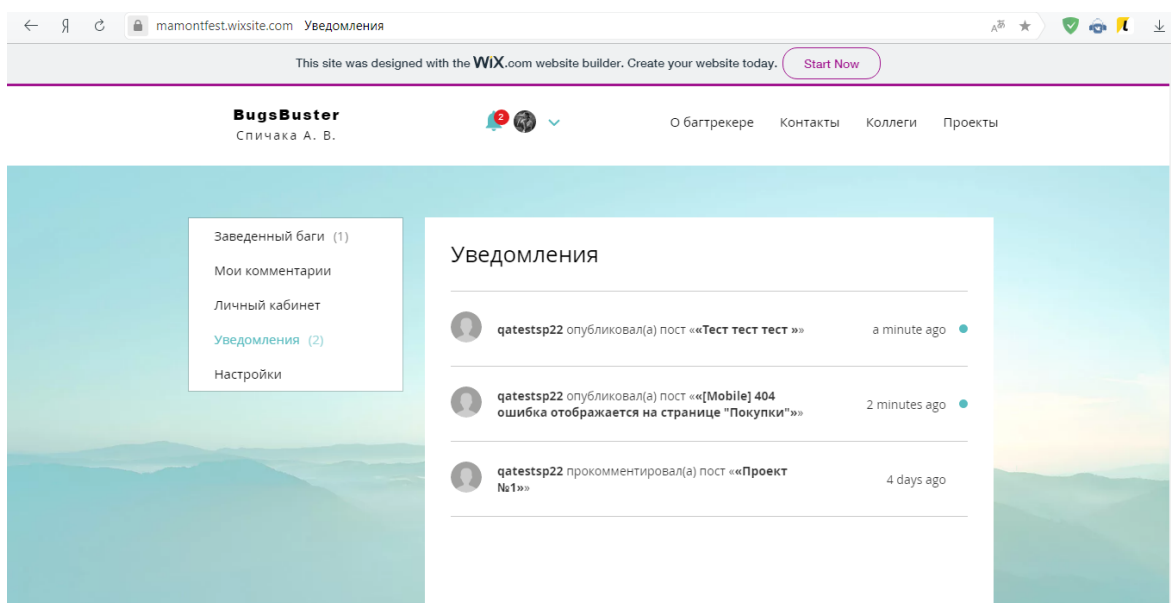


Рисунок Г.10 – Скріншот сторінки «Сповіднення»

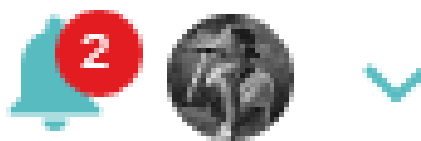


Рисунок Г.11 – Скріншот іконки сповіщень «Дзвоник»

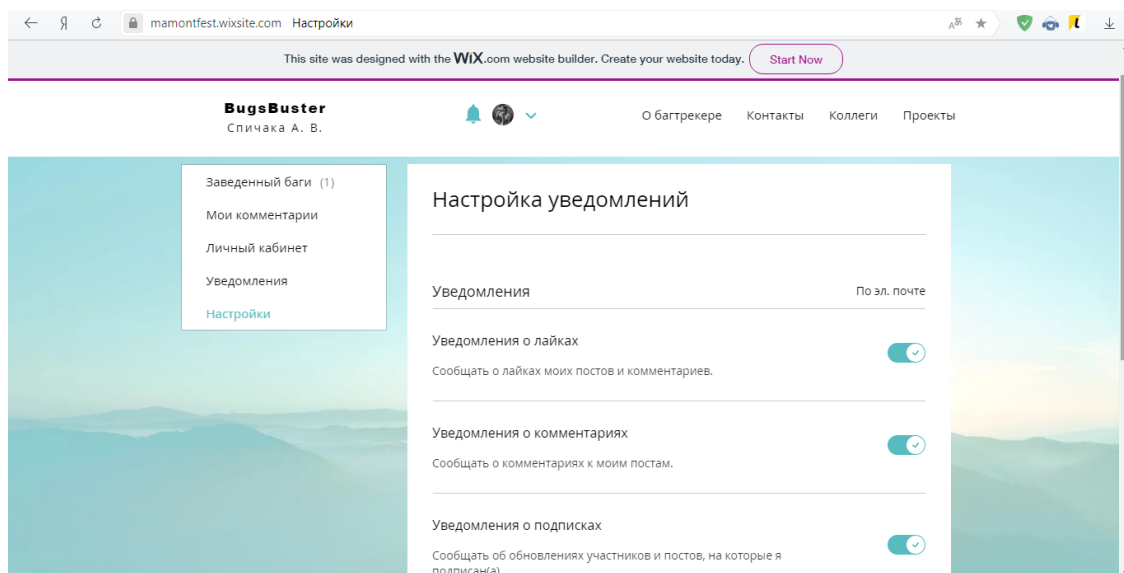


Рисунок Г.12 – Скріншот сторінки «Настройки»

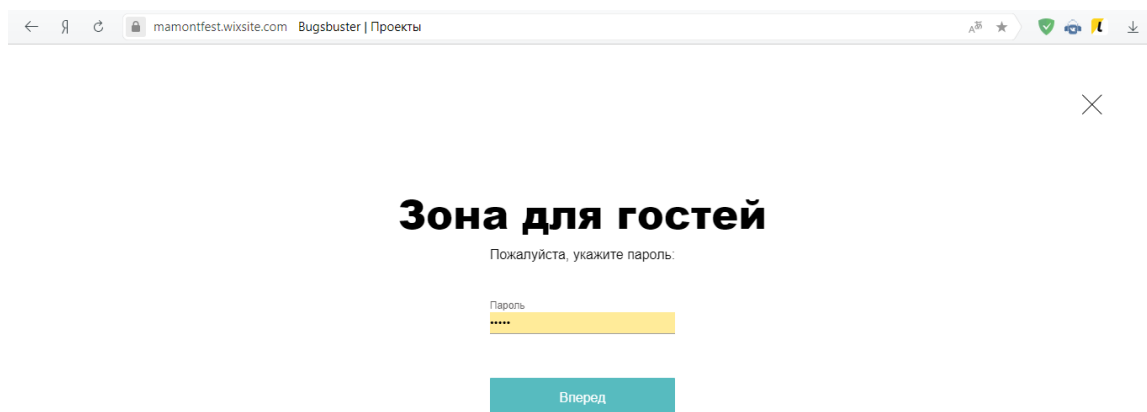


Рисунок Г.13 – Скріншот вікна вводу пароля для сторінки «Проекты»

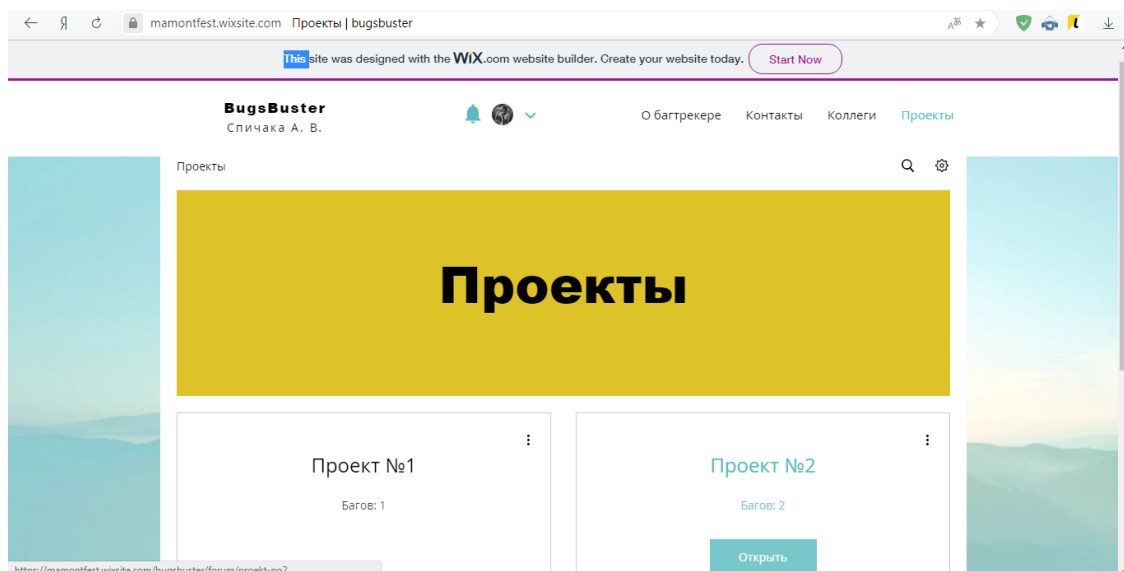


Рисунок Г.14 – Скріншот сторінки «Проекты»

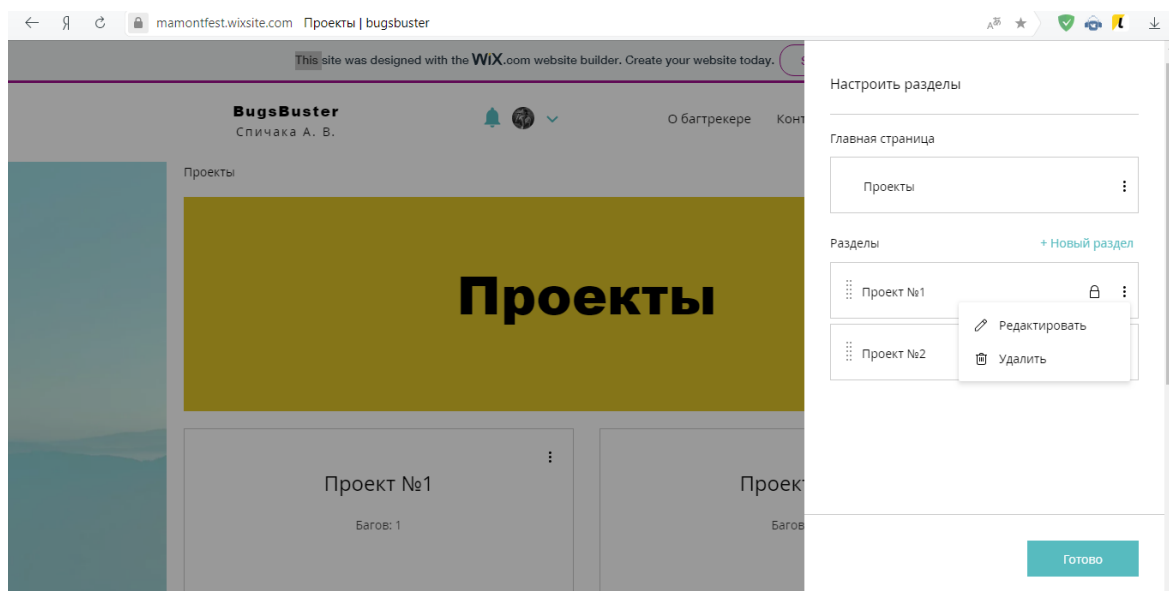


Рисунок Г.15 – Вікно управління проектами

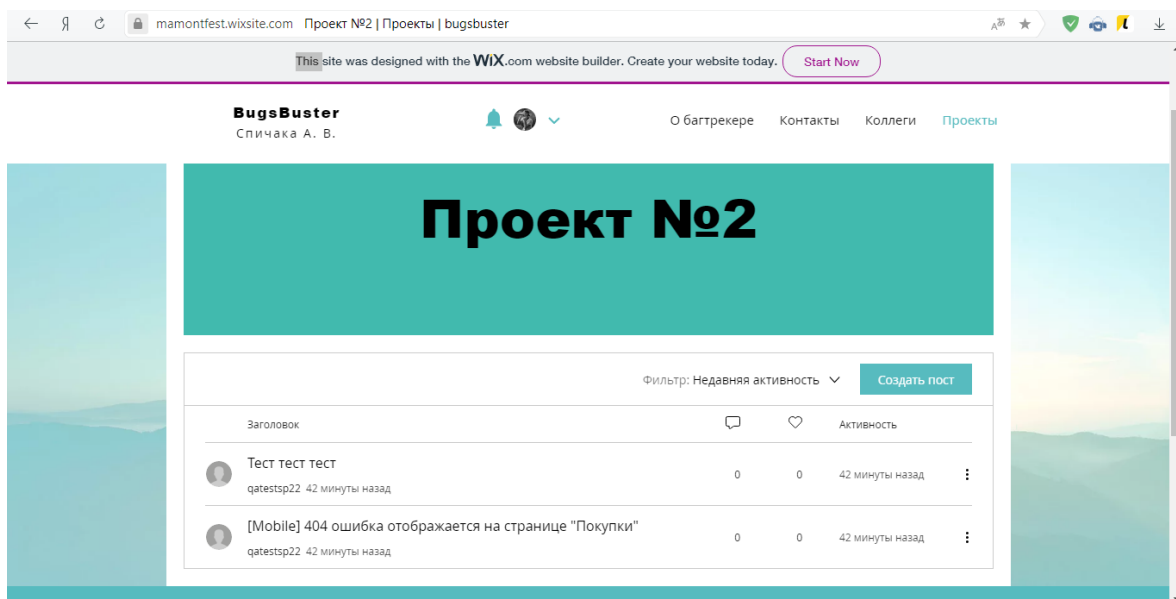


Рисунок Г.16 – Скріншот сторінки проекту

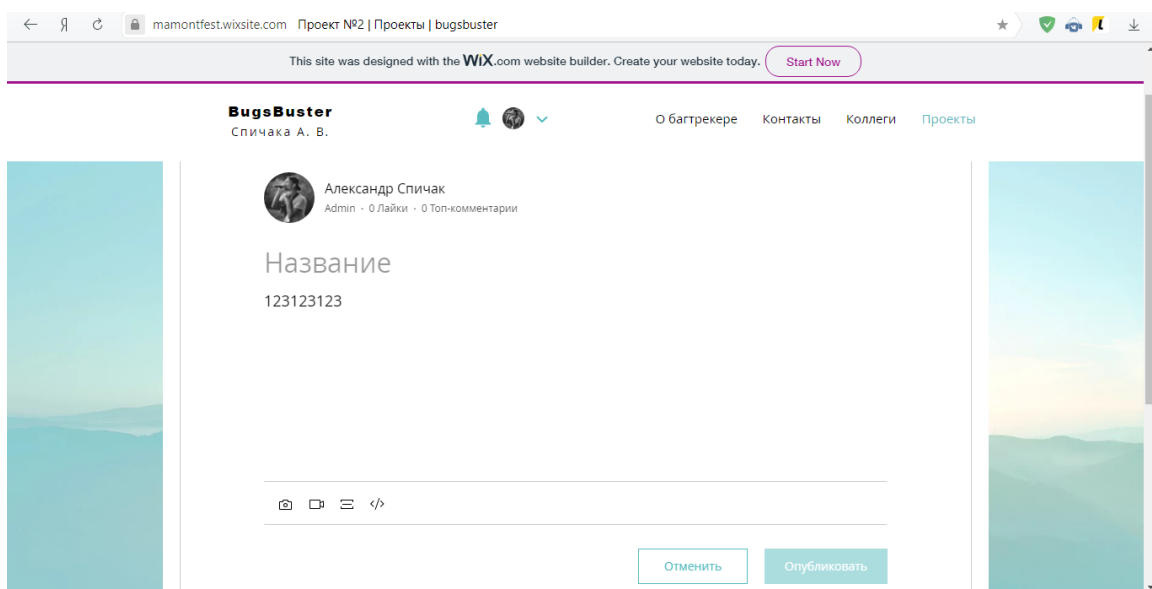


Рисунок Г.17 – Скріншот сторінки створення нового багу

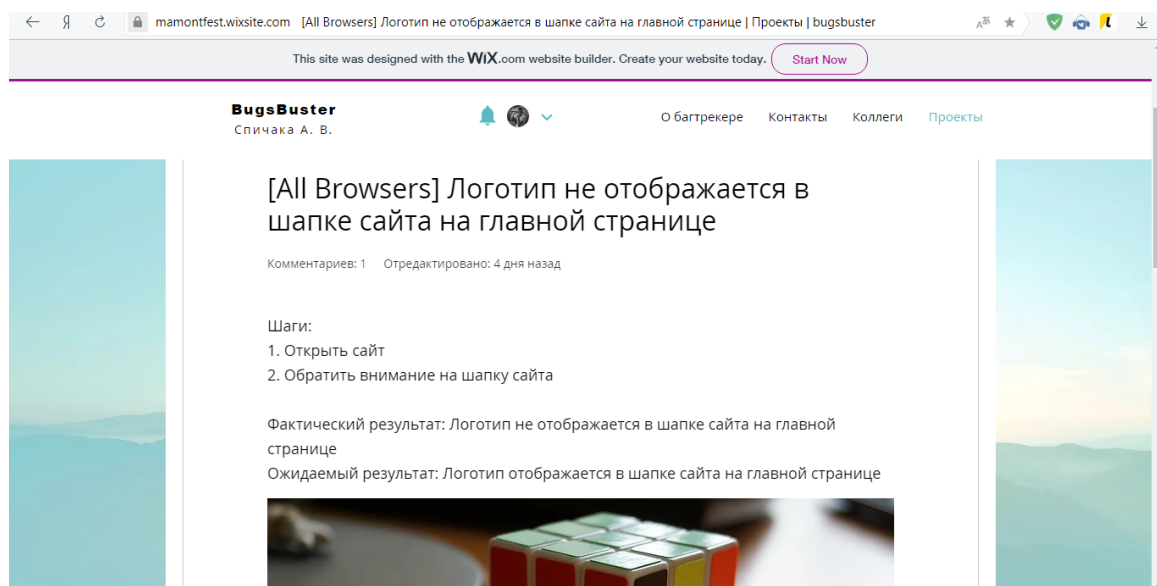


Рисунок Г.18 – Скріншот сторінки задокументованого багу

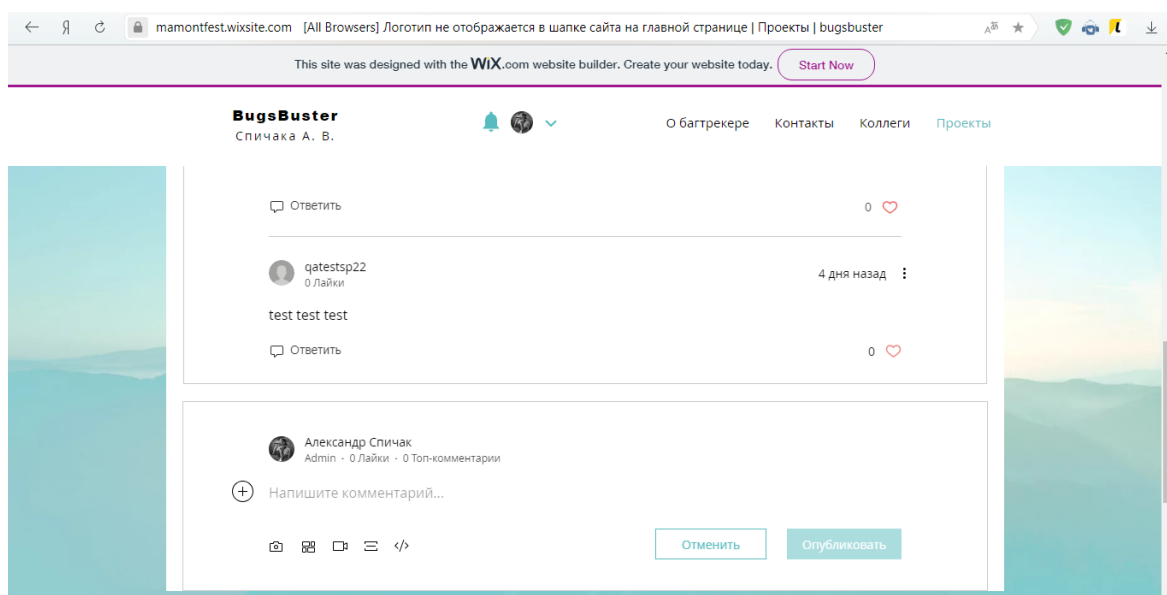


Рисунок Г.19 – Блок додавання коментарів на сторінці задокументованого бага

главной странице



Перестать следить

Ответить

Рисунок Г.20 – Кнопка вимкнення сповіщень про зміни на сторінці конкретного багу

Додаток Д (Довідковий)

Тарифні плани розробленого багтрекеру та багтрекеру-аналогу

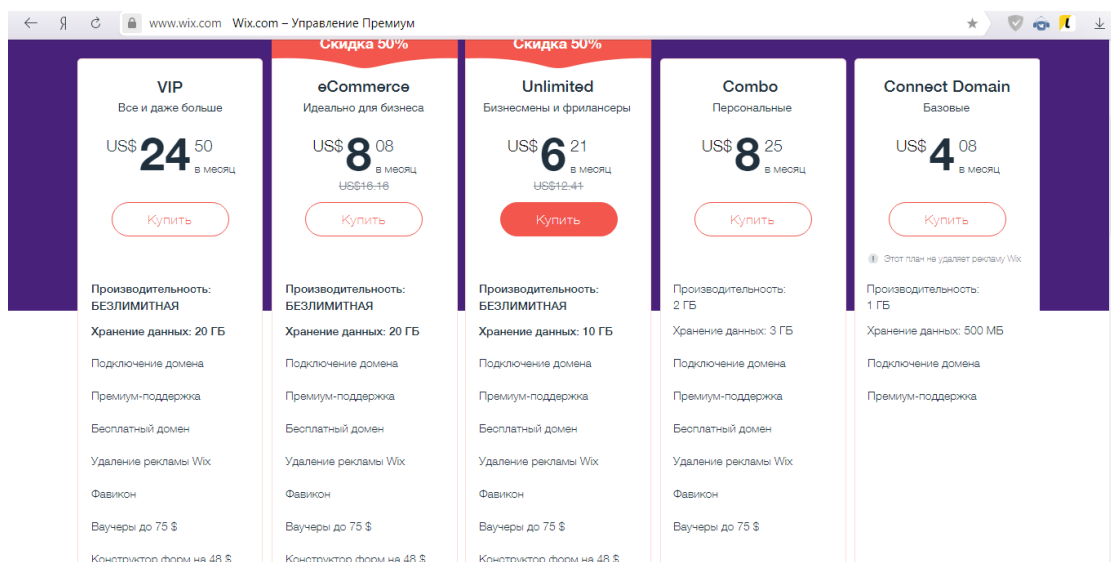


Рисунок Д.1 – Тарифні плани на сайті Wix

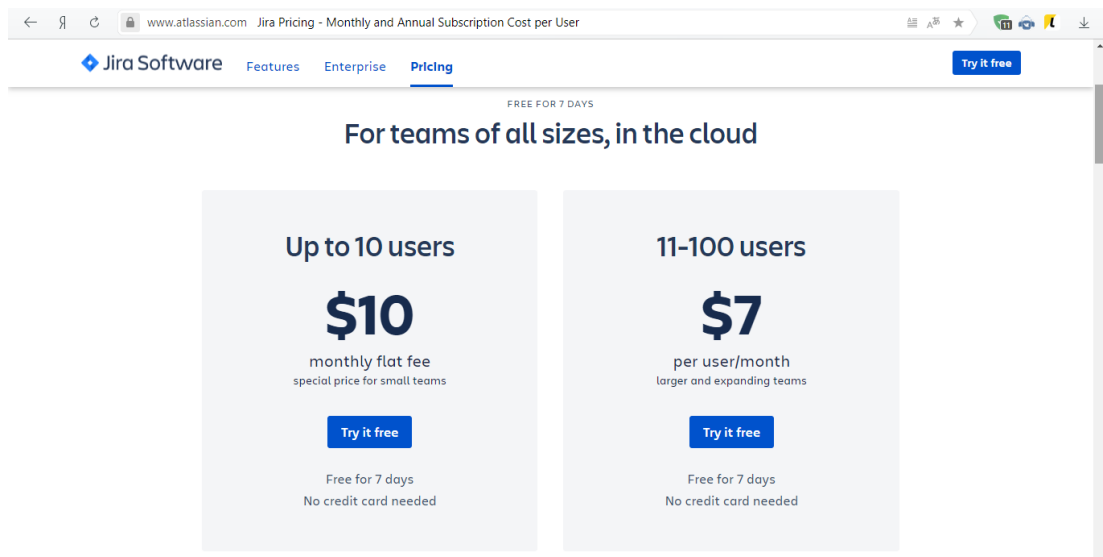


Рисунок Д.2 – Тарифні плани на сайті Jira