

**Кострикiна Д.А.**

Національний технічний університет України

«Київський політехнічний інститут імені Ігоря Сікорського»

## АНАЛІЗ СУЧАСНОГО РІВНЯ ДОСТУПНОСТІ ВЕБ-ДОДАТКІВ

*Мобільні пристрої стали основним методом спілкування та пошуку інформації для людей у сучасному світі. Для значного відсотка людей, які живуть з обмеженими можливостями, важливо, щоб ці пристрої та програмне забезпечення, були доступними у використанні для всіх. Хоча існують стандарти для забезпечення доступності в інтернеті, для вітчизняних мобільних додатків досить рідко застосовуються будь-які з них. На сьогодні розробники поступаються доступністю задля збільшення функціональності додатків, які у своїй більшості орієнтовані на користувачів, що не мають фізичних обмежень. Такий компроміс не повинен існувати, але в частіше за все він викликаний нестачею ресурсів та програмних інструментів, які доступні розробникам. Для людей з обмеженими можливостями, такими як проблеми зі слухом, зором, руховими навичками і когнітивними процесами, використання смартфонів може бути складним завданням. Слідуючи рекомендаціям щодо забезпечення доступності і розробці додатків з урахуванням цих груп користувачів, додатки можна модифікувати, щоб зробити їх більш доступними. Проте, хоча мобільні додатки в основному охоплюються тими ж стандартами доступності, що і веб-додатки, форм-фактор і методи взаємодії, використовувані мобільними пристроями, створюють нові проблеми, які розробники повинні вирішувати під час розробки.*

*У цій статті буде детально описано поточний стан сфери доступності мобільних додатків та ресурси, які допомагають розробникам та дизайнерам реалізовувати доступні веб-додатки, проаналізовано основні недоліки існуючих рішень та приведено етапи й вимоги до розроблюваної системи тестування доступності. Також буде описано проблему розповсюдження таких інструментів та впровадження таких інструментів розробки на сучасні мобільні додатки. Загалом стаття повинна розкрити необхідність створення програмного забезпечення для тестування доступності з відкритим кодом.*

**Ключові слова:** веб-додатки, доступність, тестування, алгоритм, користувацький інтерфейс.

**Постановка проблеми.** Насамперед люди з обмеженими можливостями обмежені у свої взаємодії з комп'ютерами та мобільними пристроями в цілому. Для того, щоб забезпечити безперешкодне використання пристрої та програмного забезпечення людям з різними вадами, усі додатки мають бути «доступними». Доступність програмного забезпечення передбачає реалізацію конкретних функцій, які допомагають людям у використанні технологій, що часто призводить до поліпшення зручності використання для всіх користувачів, у тому числі людей з обмеженими можливостями. Відповідно до статистичними даними Благодійного фонду допомоги інвалідам України за 2013 рік, 2,8 мільйона українців (6,1% населення України) мають певний тип інвалідності, за оцінками, у 1,3 мільйона (%) діагностували важку інвалідність [1]. Google, використовуючи дані та CDC Ukraine [2], виявив, що Україні є країною з найменшою кількістю людей з обмеженими можливостями в Європі (3,3% мають проблеми із зором та 0,7% людей з проблемами зі слухом), але не зважаючи на це Україна має одні з найменших показників «без бар'єрного середовища»: мега-

поліси мало забезпечені спеціальними світлофорами для людей з порушеннями зору чи / та слуху, не всі підземні переходи мають ескалатори для інвалідів, що вже казати про мобільні пристрої. З огляду на поширеність мобільних технологій та смартфонів у сучасному суспільстві, можна зробити висновок, що доступність цих пристроїв має першорядне значення.

**Аналіз останніх досліджень і публікацій.** Розробники дотримуються Рекомендації щодо забезпечення доступності веб-контенту (WCAG) 2.0 від Консорціуму Всесвітнього павутиння (W3C), щоб забезпечити широкий спектр доступних функцій в межах свого програмного забезпечення [3]. Цей стандарт робить реалізацію доступності в додатках чіткішою й більш цілеспрямованою, присвоюючи рівні доступності та критерії успіху кожній рекомендації, яку він надає. Рекомендації WCAG 2.0 були прийняті урядами та організаціями по всьому світу, щоб допомогти створити доступні веб-сайти. Однак, оскільки ці вказівки були створені для інтернету, вони недостатньо охоплюють усі аспекти мобільних додатків та інших веб-ресурсів. Робочі групи з W3C склали карту

WCAG 2.0 до мобільної екосистеми та запропоували нові рекомендації щодо сенсорних екранів, але не створили згуртованих настанов, таких як WCAG 2.0 для мобільних пристроїв. Таким чином, не існує узгодженого стандарту для створення власних мобільних додатків, доступних для інтернету.

Хоча мобільна доступність надзвичайно важлива, інструментів і середовищ тестування для розробки доступних додатків по суті не вистачає. Підручники і керівництва фрагментовані, бібліотеки для доступної розробки складні у використанні, а тестування доступності в реальних умовах практично відсутня. Оскільки бар'єр для входу занадто високий, розробники часто уникають завдання щодо забезпечення доступності додатків. Час і знання, необхідні для створення доступного додатку, є стримуючим фактором для більшості розробників додатків. Більш того, багато інфраструктури тестування спеціальних можливостей пов'язані з проблемами статичної доступності (проблемами зі статичними знітками екранів додатків, такими як колірні контрасти і мітки програм читання з екрану), а також з проблемами динамічної доступності (проблемами з динамічним поведінкою додатку, такими як навігація) і відповідь користувача звертаються рідше.

Було проведено мало досліджень по розробці інструментів тестування доступності, за винятком формалізації різних вимог до дизайну мобільного додатка, які повинні бути доступні для користувачів з різними обмеженнями [4, с. 384–389]. Що стосується бібліотек і структур, які розробники використовують для тестування доступних додатків, то функціональність низькорівневого і простого тестування забезпечується операційною системою Android, а деякі інструменти, такі як Google Accessibility Scanner, забезпечують тестування проблем статичної доступності. Однак ці інструменти складні у використанні, часто не універсальні для обробки певних типів додатків і не виявляють багато проблем доступності, які існують у додатках.

**Постановка завдання.** Розробити середу для тестування динамічної та статичної доступності на мобільних пристроях в цілому, яка визначала б зміни в контексті виключно за допомогою інформації про користувальницький інтерфейс, а також матиме автоматичну навігацію по додатку практично без участі розробника. Середина матиме алгоритм автоматичної генерації і скорочення простору станів призначеного для користувача інтерфейсу в контексті конкретного аналізу та

процес тестування проблем динамічної доступності, таких як клавіатурна навігація і спонтанні зміни в контексті програми, шляхом створення кінцевого автомата, що представляє додаток.

**Виклад основного матеріалу дослідження.** Коли компанія або фізична особа створює додаток, який використовуватимуть споживачі на мобільних пристроях, зазвичай існують два шляхи: вони можуть створити веб-сайт, який працюватиме на мобільних інтерфейсах, або вони можуть створити власний мобільний додаток. Мобільний додаток – це додаток, розроблений для певної мобільної платформи, найчастіше за допомогою інструментів, наданих засновниками цієї платформи. Наприклад, акт створення нативної програми для Android досягається за допомогою використання Android Studio [5]. Створення власних мобільних додатків важливо тим, що засоби розробки, що надаються потрібною мобільною платформою, зазвичай більше підходять для використання на цьому пристрої, ніж інструменти, доступні під час, скажімо, розробки веб-додатку, побудованого для телефону з використанням такої сервісу, як PhoneGap [6]. Оскільки до програми додаються більш складні функції, включаючи функції доступності, стає важче реалізувати програмне забезпечення для мобільної платформи, використовуючи лише веб-технології та універсальні платформи мобільного розвитку. Тому мобільний розвиток, порівняно з мобільним Інтернетом або гібридним розвитком, є переважачим у виборі розробки [7]. Компанії, що розробляють мобільні операційні системи, такі як Apple, Google та Microsoft, взяли під контроль власну мобільну доступність, побудувавши власні функції доступу до своїх мобільних операційних систем [8]. Незважаючи на те, що вони відрізняються на кожній платформі, ці вбудовані функції доступності в кінцевому підсумку сприяли створенню екосистеми мобільних додатків в цілому до більш інклюзивного середовища для людей з обмеженими можливостями. Кожна з цих операційних систем пропонує API та інструменти для реалізації доступних функцій

Одним із ключових кроків розробки доступного мобільного додатку є пошук стандарту для оцінки доступності вашої програми. Стандарт або настанова доступності допомагає розробнику, пояснюючи, як зробити певні компоненти доступними для користувача інвалідної групи. Сьогодні існує багато стандартів та загальних посібників для створення доступного програмного забезпечення.

Перший, найвідоміший із них, це WCAG 2.0. Ініціатива W3C щодо доступу до Інтернету (WAI) є провідною силою щодо поліпшення доступності всесвітньої мережі Інтернет для людей з обмеженими можливостями. Різні робочі групи, що складаються з провідних фахівців галузі та дослідників, створюють керівні принципи, технічні звіти, навчальні матеріали та інші документи, пов'язані з різними аспектами доступності в Інтернеті, наприклад, Правилами доступності веб-вмісту (WCAG 2.0). WCAG 2.0 – це документ, спрямований на надання рекомендацій та порад щодо впровадження доступного веб-контенту, будь то зображення, текст, відео та звук. Він має 12 керівних принципів, розбитих на 4 ключові (сприйнятливий, функціонуючий, зрозумілий та надійний), кожен з яких відрізняється рівнем критеріїв успіху (А, АА та ААА). Загалом цей документ стверджує, що чим доступнішими розробник робитиме кожен компонент програми, тим вище рівень критеріїв успіху [9].

Крім WCAG 2.0, існують й інші стандарти та вказівки, які містять поради та правила для створення доступних веб-сайтів, програмного забезпечення та програм. Наприклад, у документах із рекомендаціями щодо доступності агента користувача (UAAG) пояснюється, як зробити доступними агенти користувачів. Користувацькі агенти включають веб-переглядачі, розширення браузера, медіаплеєри, читачі та інші додатки, включаючи нативні мобільні додатки. Інший стандарт, ISO 9241, частина 171, говорить про доступність програмного забезпечення, використовуючи WCAG 2.0 в якості основи, але вивчаємо набагато більше деталей для не-веб-контенту. Європейські стандарти доступності, EN 301-549, містять вказівки щодо доступності, які є окремими для веб-документів. Незважаючи на те, що вона позичається значно у WCAG 2.0, EN 301-549 ефективно вимагає, щоб мобільні операційні системи мали API доступу. Він також заохочує розробників додатків підтримувати цей API, інакше розробник повинен відповідати довгим списком інших спеціальних функцій доступності. Крім того, приватні компанії, такі як IBM та BBC [10], мають власні стандарти доступу до програмного забезпечення, що не належить до Інтернету.

Через проблеми, пов'язані з адаптацією існуючих стандартів до рідного простору мобільної доступності, були сформовані робочі групи та доручено створити нові стандарти, які змінюють або адаптують існуючі вказівки до цього нового ландшафту. Наприклад, Робоча група W3C з

Настанов щодо доступності (AGWG), створена в грудні 2016 року, готувала узагальнену та розширену версію WCAG 2.0. За інформацією AGWG, «мета проекту» полягає в тому, щоб провести істотну редакцію стандарту WCAG та в рамках оновлення включити вимоги до користувацьких агентів та інструментів для створення авторських прав за необхідності.

Також Альянс з якості додатків (AQuA) розробив набір критеріїв тестування доступності для власних мобільних додатків для кожної мобільної операційної системи. Їх вказівки призначені для використання у перевірці доступності програми для користувачів із порушеннями в одній або декількох категоріях зору, сприйняття кольорів, слуху, мови, спритності та пізнання.

Мобільні додатки програмуються на різних мовах і вкрай залежать від мобільних рамок, що надаються їх хост-операційними системами. Дві основні мобільні операційні системи, iOS та Android, надають API доступності, які дозволяють розробникам отримувати доступ до вбудованих функцій доступності.

Існує три типи підтримки доступності, яку може надати власний розробник мобільних пристроїв у своєму додатку:

1) Основна підтримка доступності Екранні лупи, більший текст, відтінки сірого та підписи медіа.

2) Розширена підтримка доступності Це спеціальні функції, які в основному стосуються екранної навігації та презентації вмісту, як-то зчитувачі екрана та корекція кольорів.

3) Допоміжне обладнання. Сюди входять такі пристрої, як оновлені дисплеї, комутатори, ТТУ та слухові пристрої.

Кожна операційна система містить основні функції загальної системної доступності, які можна налаштувати користувачем, при цьому декілька розширених функцій, які можна знайти як в iOS, так і в Android. Замість того, щоб створювати функції доступності з нуля, розробники можуть написати кілька рядків коду, щоб змусити їх спеціально налаштовані компоненти працювати зі службами доступності та функціями в мобільній операційній системі. Поки розробники належним чином використовують ці API, їх програма буде певною мірою «доступною», що може призвести до збільшення активних користувачів. Крім того, розробники можуть створювати спеціальні функції доступності для своїх додатків. Насправді багато додатків створені спеціально для допомоги та покращення можливостей корис-

тувачів з обмеженими можливостями і часто є функціями в певних частинах магазинів додатків. Так Apple і Google мають спеціальні команди з питань доступності, які постійно розробляють функції для своїх мобільних платформ.

iPhone, випущений у 2007 році, не містив функцій доступності; сліпі люди думали, що використовувати повністю пристрій з сенсорним екраном неможливо. Apple вперше представила функції доступності в червні 2009 року за допомогою iOS 3 (тоді її називали iPhone OS 3) та впровадження VoiceOver, першого зчитувача екрана на основі жестів, побудованого на запозиченій платформі Mac [11]. Також провідною розробкою є Assistive Touch – плаваюча екранна кнопка, яка при торканні відкриває палітру параметрів управління телефоном, включаючи фізичні елементи керування, проведіть пальцем та фізичні жести та швидкі ярлики до налаштувань та додатків, включаючи Siri. Найкраще, що Assistive Touch можна налаштувати, щоб надати користувачам ефективні засоби навігації до потрібних місць на своєму телефоні та встановити власні заміни жєстами. Ще однією чудовою інноваційною функцією є Touch Accomodations, яка випущена в iOS 9. Touch Accomodations, щоб надати користувачам, які мають тремтіння рук, можливість ефективно використовувати сенсорний екран, не викликаючи занадто багато помилок та реєструючи небажані дотики. Коли користувач вибирає компонент, під їх пальцем за допомогою таймера з'являється невеликий кружечок. Якщо користувач утримує на екрані весь проміжок часу, натискання реєструється.

Розробники Apple мають декілька інструментів для перевірки доступності своїх додатків. Вбудований у Xcode – інспектор доступності, який робить тестування доступності інтерактивним. Інспектор – це вікно робочого столу, яке показує всі властивості та значення, способи (дії, які можуть відбуватися з елементів на екрані) та положення об'єкта, який наразі обраний на екрані. Тестери можуть виконувати дії через інспектора замість того, щоб виконувати жест за допомогою миші або асистентної технології.

Android від Google широко розповсюджений у 2009 році (v1.5 Cupcake) і на той час не пропонував жодних функцій доступності [12]. З тих пір Android став міцною базою функцій доступності. Android містить спеціальний розділ у Налаштуваннях під назвою «Доступність», який дозволяє користувачам контролювати параметри доступності для операційної системи в цілому. Станом

на користувачів Android v7.1.2 дозволяється знаходити, переглядати або чути інформацію на екрані, дозволяючи TalkBack, BrailleBack, контрастні та кольорові параметри, підписи, збільшення та розмір шрифту чи дисплея. Крім того, вбудовані налаштування доступності дозволяють користувачам активувати голосовий доступ, що дозволяє користувачеві зробити вхід, використовуючи голос або зовнішній пристрій [13].

Заздалегідь встановлений Android-вбудований зчитувач екрану – це TalkBack, який взаємодіє з макетами, щоб надати корисну інформацію користувачеві щодо інформації на екрані. Для використання цієї функції до компонентів інтерфейсу користувача можна додавати описи вмісту або теги для представлень та об'єктів в елементах інтерфейсу користувача Android під час розробки або під час виконання. Ці описи вмісту TalkBack можуть бути зчитані за допомогою аудіо через динаміки або гарнітуру. Надаючи ці описи, TalkBack може передати користувачеві інформацію про орієнтовані на даний момент погляди, інформацію в частині тексту та навігаційну інформацію. Крім простого вимовлення цих тегів вголос, опис вмісту на зображеннях також використовується функцією підписів Android, яка розміщує підписи до зображень для розширеного пояснення зображення [14].

Крім TalkBack, BrailleBack та подібних функцій, ще одна методика, яку розробники використовують для забезпечення доступності в своїх додатках, – це створення AccessibilityService та AccessibilityEvent. Ці об'єкти дозволяють розробникам створювати сервіси, які можуть слухати певні події на пристрої та виконувати дії, реагуючи на ці події. Наприклад, подія в інтерфейсі користувача може запустити службу доступності для зміни розміру або кольору шрифту в програмі. Однак, хоча цей API корисний тим, що дозволяє розробникам створювати більші та більш досконалі сервіси доступності, додатки, що реалізують AccessibilityService, містять відставання в користувацькому інтерфейсі, що було послідовним у кількох випусках. Окрім стандартного інтерфейсу, пристрою потрібно передавати велику кількість даних службі AccessibilityService, щоб правильно реагувати на введення користувача [15].

Колекція посібників, навчальних посібників та документації щодо впровадження доступності в додатки для Android досить велика. Android має сайт, присвячений цим урокам, який охоплює уроки зручності використання матеріального

дизайну аж до конкретних технічних реалізацій користувацьких доступних представлень. Одним із найкорисніших розділів цього веб-сайту є контрольний список розробників доступності, який надає перелік завдань та ідей, які слід враховувати, оскільки розробник розробляє та буде їх застосування [16].

Проаналізувавши сучасний стан сфера доступності мобільних додатків для створення доступних власних мобільних додатків, було виявлено, що наявні інструменти, доступні розробникам, недостатньо і їх можна вдосконалити. Хоча кількість доступних функцій, включених у мобільну операційну систему, таких як iOS, Android та Windows, збільшується, розробники все ще втрачають у наданні доступних функцій в самих додатках. Найбільш корисні інструменти, доступні розробникам, це не програмне забезпечення, а підручники та керівництва, знайдені в інтернеті, які можна використовувати для створення доступного додатку. Однак можна стверджувати, що, хоча вони дають гарний старт для створення доступних додатків, вони також можуть бути дорогими, ексклюзивними та неповними. Крім того, значна кількість різних джерел для цих посібників ускладнює вибір правильного навчального посібника, що спричиняє фрагментарний простір керівних принципів та пропозицій щодо доступності.

Створення доступних додатків гарантує, що додатки функціонують, сприймаються та зрозумілі всім користувачам. Для досягнення цього сьогодні розробники повинні знайти баланс між функціональністю та естетичним дизайном своїх мобільних додатків. Крім того, програми повинні використовувати належну мову для спілкування з користувачами.

**Висновки.** Доступність для рідних мобільних додатків є важливою темою, оскільки це морально і законно потрібно робити. У процесі розробки доступних мобільних додатків розробники в даний час стикаються з перешкодами, такими як неможливість адаптувати раніше існуючі вказівки до їхніх проектів, використовуючи неповні та недостатньо бібліотеки для надання функцій, і складний час вибору, які підручники та посібники слід виконувати під час розвитку. Доступність для рідних мобільних додатків займалася багатьма компаніями та групами, але це призвело до безлічі вказівок, які є непосильними та все ще неповними. Компанії з мобільними платформами мають вбудовані функції доступності в свої операційні системи, але потрібно зробити більше, щоб переконати розробників у включенні доступності у свої програми.

Можна сказати, що перед сучасними розробниками постає проблема створення універсального інструменту для впровадження, а головне – тестування доступності створюваних ними додатків, яка б тестувала як статичні, так і динамічні проблеми доступності, і яку можна легко розширити для обробки інших вимог щодо доступності, які покладаються на статичні або динамічні дані про програму. Крім того, важливим є надання простого, але інформативного інтерфейсу для тестування, який би не вимагав від розробника специфічних навичок чи знань. Відстеження динамічного стану програми у часі у відповідь на введення користувача неймовірно корисно для широкого спектру проблем із доступністю. Існуючі рамки тестування доступності, у яких немає такої інформації, лише оцінюють доступність одного екрана за один раз і не охоплюють багато сценаріїв, де можлива доступність.

#### Список літератури:

1. Статистика інвалідизації в Україні [Електронний ресурс]. – Режим доступу: <http://ukrhelp.net/en/statistika-invalidizatsii-v-ukraine.html>.
2. Golyk V. Assessment of national disability, health and rehabilitation system in Ukraine. Reflections disability and rehabilitation subcluster meeting, March 11, 2016 / V. Golyk, O. Syvak // Український вісник медико-соціальної експертизи. 2016. № 1. – Режим доступу: [http://nbuv.gov.ua/UJRN/ujmse\\_2016\\_1\\_9](http://nbuv.gov.ua/UJRN/ujmse_2016_1_9).
3. Caldwell B., M. Cooper M., Reid L. G. Web content accessibility guidelines (wcag) 2.0 : WWW Consortium (W3C), 2008.
4. Macedo J., Gouveia T., Florentin F. Usability requirements for mobile accessibility: a study on the vision impairment. 14th International Conference on Mobile and Ubiquitous Multimedia (MUM'15): The OX Association for Computing Machinery, ACM. New York, NY, USA, 2015.
5. Android studio the official ide for android. [Електронний ресурс] – Режим доступу: <https://developer.android.com/studio/index.html>.
6. Native vs hybrid / phonegap app development comparison. [Електронний ресурс]. – Режим доступу: <http://www.comentum.com/phonegap-vs-native-app-development.html>.
7. You're using phonegap for all the wrong reasons. [Електронний ресурс]. – Режим доступу: <https://readwrite.com/2015/06/24/phonegap-apache-cordova-cross-platform-tools/>.

8. An introduction to native mobile accessibility – featuring deque university for ios/android [Електронний ресурс]. – Режим доступу: <https://www.deque.com/blog/introduction-native-mobile-accessibility-featuring-deque-university-iosandroid/>.
9. Web Content Accessibility Guidelines 2.0, W3C World Wide Web Consortium Recommendation 11 December 2008, Success Criteria 1.1.1, 1.4.3, 1.4.6, 2.1.1, 2.4.6, 2.5.5, 3.1.5, 3.2.1, 3.2.5. [Електронний ресурс] - Режим доступу: <https://www.w3.org/TR/2008/REC-WCAG20-20081211/Accessed Dec 29, 2018>.
10. Mobile Navigation Guidelines [Електронний ресурс]. – Режим доступу: <http://www.funka.com/en/our-assignments/research-projects/archive---research-projects/mobile-navigation-guidelines/>.
11. 5 years of voiceover: Look how far we've come [Електронний ресурс]. – Режим доступу: <https://www.applevis.com/blog/advocacy-apple-assistive-technology-ios-iphone-news/5-years-voiceover-look-how-far-weve-come>.
12. A brief history of android accessibility [Електронний ресурс]. – Режим доступу: <https://accessibleandroid.wordpress.com/2013/08/02/a-brief-history-of-android-accessibility/>.
13. Android accessibility overview – android accessibility help [Електронний ресурс]. – Режим доступу: <https://support.google.com/accessibility/android/answer/6006564?hl=en>.
14. Captions – android accessibility help [Електронний ресурс]. – Режим доступу: <https://support.google.com/accessibility/android/answer/6006554?hl=en>.
15. Working as intended [Електронний ресурс]. – Режим доступу: <https://www.xda-developers.com/working-as-intended-an-exploration-into-androids-accessibility-lag/>.
16. Testing your app's accessibility [Електронний ресурс]. – Режим доступу: <https://developer.android.com/training/accessibility/testing.html>.

#### **Kostrykina D.A. ANALYSIS OF THE CURRENT ACCESSIBILITY LEVEL OF WEB APPLICATIONS**

*Mobile devices have become the main method of communication and information search for people in the modern world. A large percentage of people living with disabilities, it is important that these devices and software are accessible to use for all. Although standards exist to ensure accessibility on the Internet, any of them is rarely used for domestic mobile applications. Today, developers are inferior to accessibility to increase the functionality of applications, which for the most part are aimed at users who do not have physical restrictions. Such a compromise should not exist, but most often it is caused by a lack of resources and software tools that are available to developers. For people with disabilities, such as hearing, vision, motor skills, and cognitive problems, using smartphones can be challenging.*

*Following the recommendations for accessibility and application development based on these user groups, applications can be modified to make them more accessible. However, while mobile applications are largely covered by the same accessibility standards as web applications, the form factor and interaction methods used by mobile devices create new problems that developers must solve during development. This article will describe in detail the current state of the sphere of accessibility of mobile applications and resources that help developers and designers to implement affordable web applications, analyze the main disadvantages of existing solutions and provide steps and requirements for the developed accessibility testing system.*

*Also discuss the problem prevalent of such tools, and will answer the questions of the introduction of such development tools on modern mobile applications. In general, the article should reveal the need for open source accessibility testing software.*

**Key words:** *web applications, accessibility, testing, algorithm, user interface.*