Mobile and Web Testing
SFWR ENG 3S03: Software Testing

Alicia Marinache

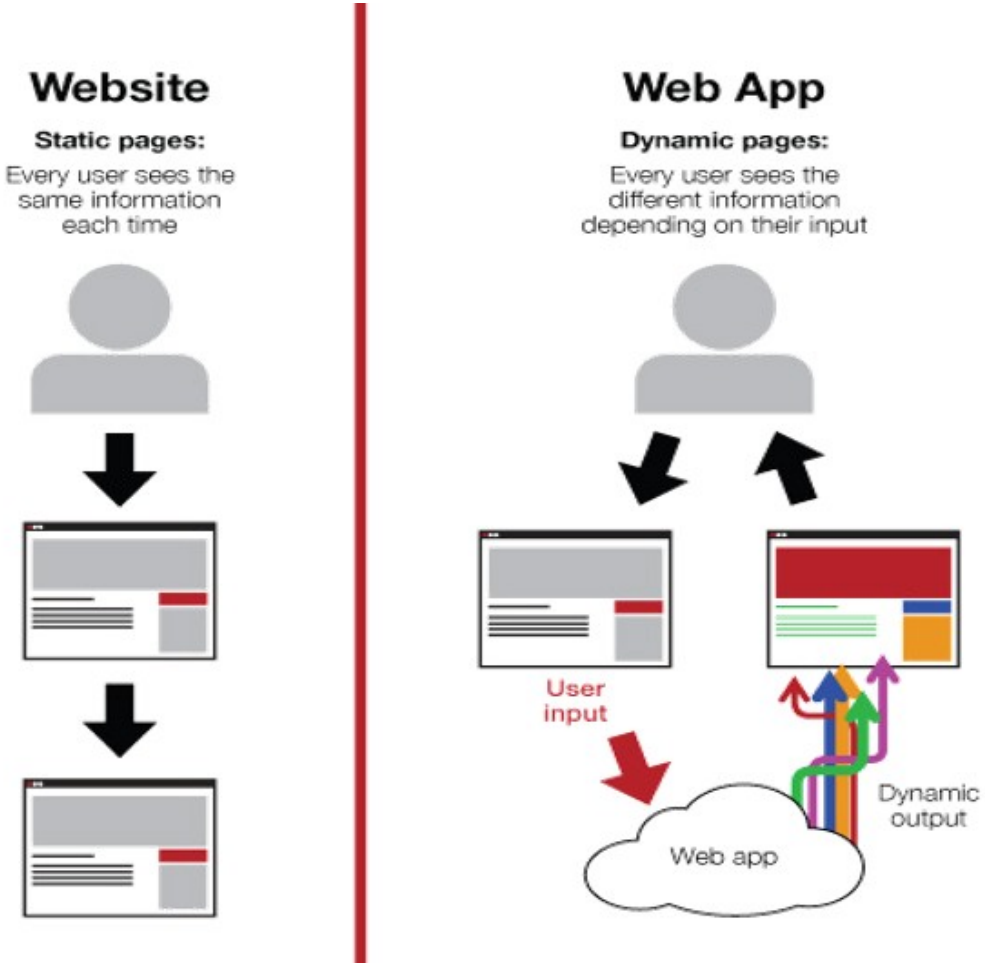
Department of Computing and Software, McMaster University
Canada L8S 4L7, Hamilton, Ontario

Acknowledgments: Material adapted from Dr.R.Paige

Objectives

- Understand Website vs. Web App, Mobile vs. Mobile App
- Explore Web and Mobile App Testing Types
- Explore Testing Automation

Website vs Web Application



Mobile Application

Web and Mobile Testing

- Web testing: website or web application works as expected
- Mobile testing: mobile device (hardware) works as expected
- Mobile application testing: applications (software) on mobile devices works as expected
- Performed before the application goes live for end users
- Ensures functionality and user acceptance
- Important: both mobile and web application are on the rise

Challenges

- Variety of browsers, mobile devices, and platforms
- Different network providers and input methods
- Hardware compatibility and OS fragmentation
- Variety of communication protocols
- Rich, complex applications

- Functionality Testing
- Performance Testing
- Usability Testing
- Compatibility Testing
- Security Testing
- Recoverability Testing

Functional Testing

- Verifies that the web or mobile app works according to its specifications.
- Includes tests for user interactions, feature availability, and expected behavior.
- Ensures that all functionalities work as intended, such as buttons, forms, and workflows.

Functional Testing Scenarios

- Web: Verifying that all links are working and forms are submitting correctly.
- Mobile: Verifying app navigation, input validation, and integration with mobile device features.
- Difference: Mobile apps need to be tested for device-specific features (e.g., GPS, camera) that web apps don't.

Performance Testing

- Ensures the web or mobile app performs well under varying conditions.
- Tests for speed, responsiveness, and scalability.
- Measures load times, transaction times, and server response to determine performance bottlenecks.

Performance Testing Scenarios

- Web: Load testing under heavy traffic, ensuring the app doesn't crash under high user numbers.
- Mobile: Testing performance on varying network conditions (e.g., Wi-Fi, 4G, offline).
- Difference: Mobile apps need to account for varying hardware capabilities, such as different processor speeds.

Usability Testing

- Measures how user-friendly the web or mobile app is.
- Checks navigation, clarity of content, and overall user experience.
- Focuses on how intuitive the app is for users with no prior training or instructions.

Usability Testing Scenarios

- Web: Verifying that the website layout is clear, buttons are well-positioned, and text is readable.
- Mobile: Testing the app's responsiveness on different screen sizes and touch interaction efficiency.
- Difference: Mobile apps need more attention to touch input patterns, while web apps rely on mouse clicks and keyboard.

Compatibility Testing

- Ensures the web or mobile app works across different platforms, browsers, and devices.
- Verifies that the app displays correctly and functions on all targeted operating systems, screen resolutions, and devices.

Compatibility Testing Scenarios

- Web: Testing how the website renders on different browsers (Chrome, Firefox, Safari) and devices (PC, Mac, tablets).
- Mobile: Testing the mobile app on multiple devices (Android, iOS) and OS versions.
- Difference: Mobile apps need to be tested on different hardware configurations (screen size, resolution).

Security Testing

- Verifies that the web or mobile app is secure against potential threats.
- Focuses on vulnerabilities like SQL injection, cross-site scripting (XSS), and data encryption.
- Ensures user data is secure and private, protecting against unauthorized access and attacks.

Security Testing Scenarios

- Web: Verifying SSL/TLS encryption for sensitive data, testing for XSS and CSRF vulnerabilities.
- Mobile: Testing for secure data storage, secure communication (HTTPS), and potential risks from app permissions.
- Difference: Mobile apps need to account for local storage vulnerabilities (e.g., unsecured SQLite databases).

Recoverability Testing

- Verifies that the web or mobile app can recover from failures or crashes.
- Focuses on backup systems, data recovery, and failover strategies.
- Ensures the system can restore data and functionality after unexpected crashes or downtime.

Recoverability Testing Scenarios

- Web: Testing if the website data is recoverable after server crashes or network failures.
- Mobile: Testing app data persistence across device reboots and network outages.
- Difference: Mobile apps might require testing for both offline and online recovery scenarios, which web apps rarely do.

Test Automation

- Automates repetitive browser actions, speeding up testing
- Both manual and automated testing are widespread and can provide good results
- Benefits of Mobile/Web Test Automation
  - Faster feedback and improved test efficiency.
  - Reduced expenses and faster time to market.

Manual vs Automated Testing

	MANUAL TESTING	AUTOMATED TESTING
ADVANTAGES	• Cost-effective in a short-term period • More flexible • Better simulation of user actions	• Cost-effective in the long-term • The running process is faster • Can be reused easily • Test results are easy to share
DISADVANTAGES	• Difficult to reuse • The running process is slow • Some cases can't be executed manually	• Automated tools may have limitations • Less efficient in the defining user-friendliness or positive customer experience

Web Testing: Selenium Overview

- Selenium is an open-source testing tool for web applications.
- Test scripts can be written in Java, Python, C#, and other languages.
- It supports browsers like Firefox, Chrome, Safari, and Internet Explorer.

- Universal tools
  - Eggplant: industry-leading test automation and monitoring platform
  - Ranorex: automated tool for mobile, web and desktop solutions
  - Appium: open-source test automation framework for mobile native, hybrid and web solutions
  - Kobiton: AI-powered tool for manual and automated testing
- OS-specific tools
  - Selendroid: test automation framework for native or hybrid Android apps
  - iOS Driver: open-source tool for iOS native, hybrid, or mobile web solutions

- Web and Mobile Testing ensures that applications function as intended across various environments
- Challenges in mobile testing include device fragmentation, network conditions, and OS variations
- Types of testing include all kinds we have discussed in the course
- Tools can be used to automate tests, speeding up execution and improving efficiency