

Client: Multinational FinTech Business

Industry: Financial Services

Technologies: Espresso, EarlGrey

Ten10 Services: Test Automation

Leading multinational FinTech business, providing users with advice based on their credit score and report via its website and mobile applications. The client wanted to increase the coverage and therefore quality assurance gained from an increased set of automated mobile UI tests, integrating with its CI pipeline and allowing developers to run a regression suite on their local branches.

What We Did

There was a significant mobile UI automation backlog including both the maintenance of existing and creation of new automated tests. In addition, the existing mobile UI automation solution had limited mobile and device coverage, focusing on Android using the Espresso test framework and one device and OS version. This meant that no feedback was offered regarding the compatibility of the application with other devices and versions.

Following an initial period on-site at the Client to establish relationships and the scope and plan for the project, Ten10's Leeds based Automation Factory implemented the project on an iterative, two week basis. With the scope of each iteration established for each iteration, this allowed the focus and scope of the project to adapt with changing business priorities and focus. The iterative approach also allows for capacity to be increased or decreased per sprint, using a cost effective blend of senior automation consultants and more junior resources.

Ten10's Automation Factory was used to design and implement additional test automation solution features, along with ongoing maintenance and creation of automated tests based on the prioritised backlog. Once the initial backlog was cleared, focus then moved on covering new functionality.

A greater level of assurance was achieved by increasing scope across common Android and iOS OS/ version pairs as recorded by the Client's analytics, with EarlGrey used to deliver automated tests for iOS.

Automated tests were integrated into the CI pipeline to validate pull requests before they could be merged into the main code base, helping to identify issues before they were brought into the main branch. It also reduced the number of required bug-fixes downstream and delays to the introduction of new features.

Challenges

To meet the goal of integrating the UI tests into the CI pipeline, automated tests had to be reliable when run on the available emulators for different device/OS combinations. Ten10 were able to resolve issues with screen sizes and other compatibility by making the tests more robust and issues with deprecated libraries were resolved by collaborating with the Client's dev team to produce a change to production code to conform to the requirements of the tests.

For device emulation, a comparison was completed between the default device testing feature of the CI pipeline and a 3rd party platform, triggering the run using a custom step in the pipeline. It was determined that the default method provided better reporting, was faster and was beneficial to the use case of validating pull requests as the feedback loop was shorter and the test coverage was clear.

Finally, a stubbing mechanism was implemented for the Client's backend, enabling the recording of HTTPS requests and playback of them to the app.

“Having tried and failed to find appropriately skilled and experienced resources to help us with mobile automation in our London office, we were initially sceptical that Ten10's Automation Factory service would have the skills and experience to meet our requirements and standards. These concerns quickly disappeared however, and we have been delighted in the quality of Ten10's technical experience and quality of work which has allowed our developers to focus on their primary tasks, resulting in a higher quality solution for our customers.” The Client.

Benefits

- Ten10's Automation Factory provided a high value, cost effective and flexible solution for the Client to address their mobile automation backlog and ongoing requirements.
- Greater automated test coverage provided confidence in new features developed for the app, reducing the amount of manual retesting required. Therefore changes made during bug-fixes could be checked before being pushed for review.
- Integration with CI reduced the risk of introducing bugs into the areas covered by the tests. Having confidence in the code being added to the main branch allowed the testers to spend less time running regression tests, allowing them to concentrate on other testing activities.
- By refactoring the existing Android tests to follow a consistent coding standard, it became easier to spot opportunities for shared step definitions and to limit code repetition, making the test more maintainable for new Developers on the team.