

Smart glass testing through Magnus E2E automation solution



Table of contents

Abbreviations	3
Introduction	4
Business challenges	4
Problem statement	5
Solution	5
Benefits	8
Conclusion	8
References	8
Author information	9

Abbreviations

VA	Virtual Assistant
STT	Speech to Text
TTS	Text to Speech
IoT	Internet of Things
BDD	Behavior-driven Development
CI	Continuous Integration
CD	Continuous Development
ATSG	Automated Test Script Generation (ATSG)
VM	Virtual Machine
UI	User Interface

Introduction

The smart glass market has undergone a constant evolution on the basis of different technologies, such as micro chromic, suspended particle display, micro blinds and control system-based (manual, remoted, mobile-based, voice-based).

The current market size of smart glasses is ~\$5.8B and is expected to reach \$9.4B by 2029 with a CAGR of 9.8%.

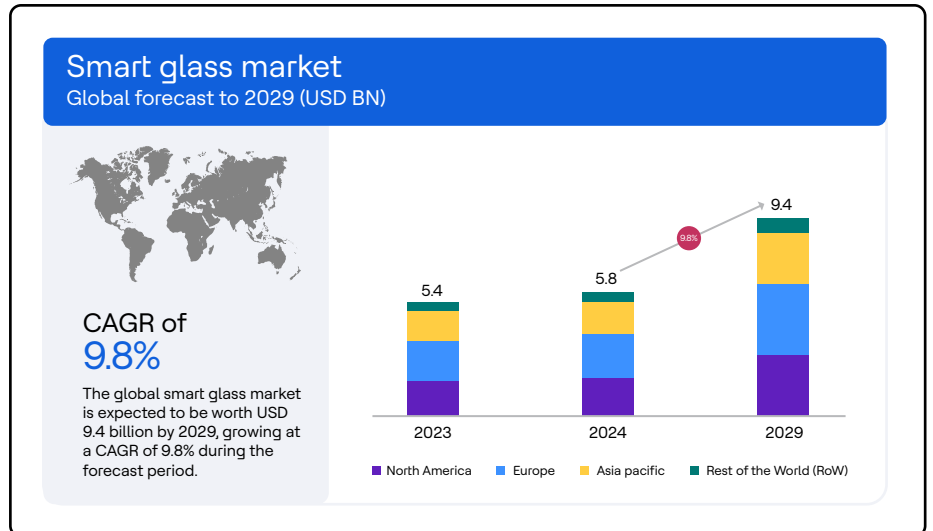


Figure 1. Smart glass market growth forecast

The enhancements in glass functionality and connectivity can be attributed to the increasing advances in electrochromic materials and IoT integration. These improvements make it imperative for the service providers to be equipped with handling the development of the application, updating firmware and providing testing solutions for the smart glass.

Business challenges

Testing of smart glass poses many challenges, including:

- Device-related inconsistency issues that are hard to reproduce and go undetected during manual testing.
- The complexity involved in porting scripts for different platforms like Android and iOS.
- Complicated setup configuration encompassing multiple devices, phones, VMs and several types of speakers.
- With multiple languages supported, differences in parameters, voice accents and commands for each language.
- Data and parameters change based on the linguistic bundle.
- Verifying the issue (such as distinction amongst the extracted text).

- Delicate devices being prone to hanging and overheating due to continuous usage.
- STT accuracy being dependent on the quality of speakers, surrounding conditions and the quality of Bluetooth and Wi-Fi bandwidth.

Problem statement

The surge in the demand for Virtual Assistant (VA) devices among the consumers has led to the introduction of several unique VA devices in the market by major players such as Saint-Gobain and Ray-Ban. Smart glasses another set of devices in this domain, have garnered significant interest from users. However, despite their promising applications, challenges persists regarding quality and accuracy, given the early stage of adoption. Establishing reliable testing methods smar glasses would cerainly facilitate quality improvement and wider adoption among consumers.

Solution

There are very limited solutions in the market for testing VA devices – and even the existing solutions fall short of fulfilling the unique requirements of smart glasses. In response to the scarcity of these testing solutions, HCLTech has upgraded its Magnus test automation framework tailored for smart glass testing.

Originally designed for device, web, mobile applications and VA device testing, Magnus now addresses the unique requirements of smart glasses. It offers support for Android and iOS platforms, as well as localization testing for multiple languages.

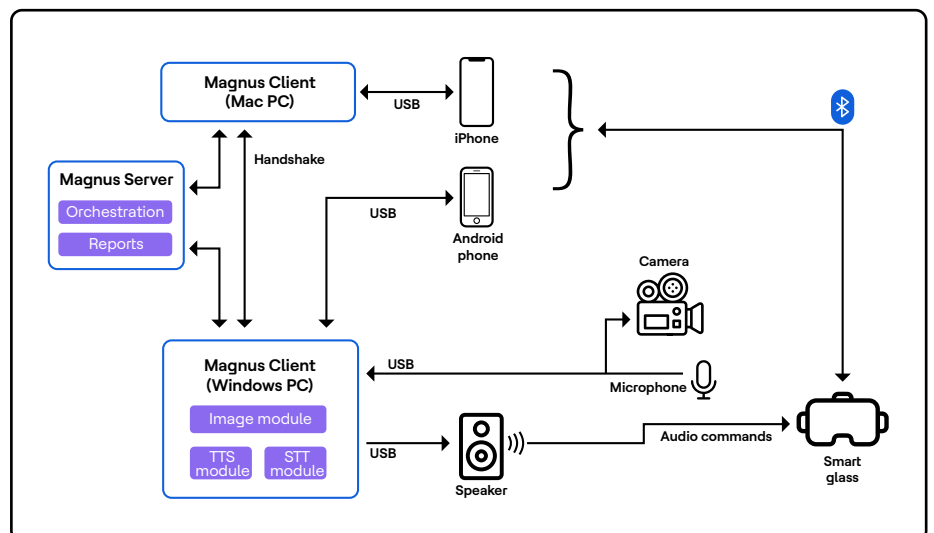


Figure 2. Automation setup for smart glass

Below is a streamlined test case flow for smart glasses:

- The smart glass device establishes a Bluetooth connection with the mobile phone.

- Magnus Client manages voice command transmission to the smart glass device and receives responses via microphone or camera, verifying them accordingly. Voice commands are generated by converting the input test data from text to voice data. Furthermore, Magnus client also navigates through the mobile phone application, capturing screen images for verification.
- Magnus Server is responsible for orchestrating the test scripts and generating the test report.

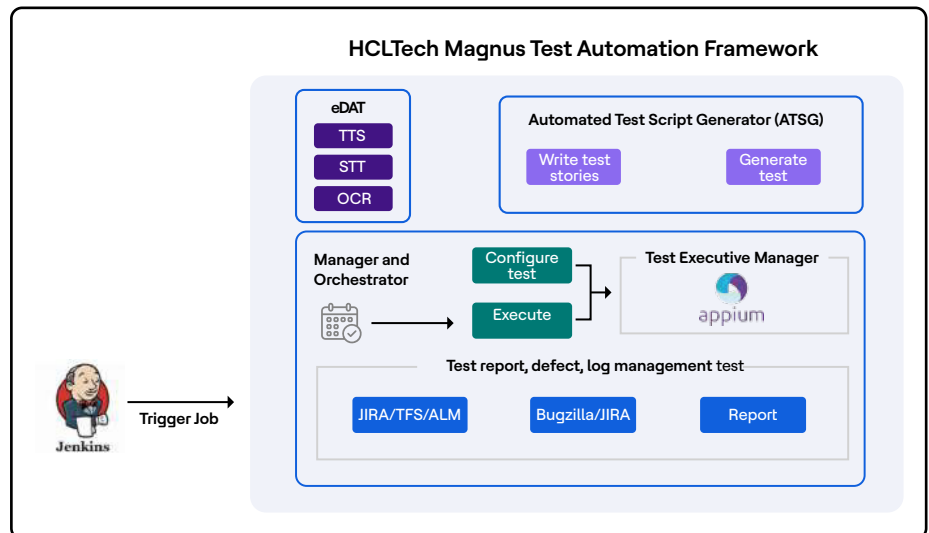


Figure 3. End-to-end test execution

Magnus encompasses several components of comprehensive test execution of test scripts including:

- **Automated Test Script Generator (ATSG):** ATSG, a Magnus-provided Studio supports the Behavior-driven Development (BDD) approach and features a library of pre-defined keyword phrases (Behaviors), an object repository and a test data repository for script generation. Users can develop test stories (sequence of test steps) and generate the automation script) using the keywords, object repository and test data repository. ATSG even provides the flexibility to add custom keyword phrases based on the custom requirements. The automation engineer can craft BDD script for smart glass test automation using built-in library of keywords. ATSG processes these BDD script to generate the automation script.eliminating the need for users to possess automation skills or understand the keywords implementation. Thus, even manual testers can use Magnus to automate smart glass test cases.
- **Test Manager and Orchestrator:**
 - It streamlines the configuration of both the mobile device and the system running the Magnus Client runs facilitating execution of the test cases seamlessly.
 - Test data ingestion occurs through configurable files, enabling localization testing for multiple languages without altering the test scripts. This feature empowers the testing of smart glasses supporting multiple languages.
 - Configure test plan helps in creating TestPlan, TestSuite, Test Configuration and Test environment setup.

- Execute test plan: BDD scripts are organized within a test plan, grouping test cases belonging to the same smart glass module/-functionality. Test plan execution can be initiated through Magnus UI or Scheduler or a Continuous Integration system like Jenkins. Once the execution is triggered, the test jobs are generated based on the test configuration. The test job along with supporting files, are dispatched to the host or client. The client executes the test scripts and transmits the execution details and pieces of evidence back to the orchestrator.
- Test Execution Manager: It facilitates the execution of generated scripts using the Appium tool or any preferred tool selected by the tester. Appium supports both Android and iOS platforms using the same test script. The core feature of the Magnus framework lies in its test execution orchestration capability, enabling touchless test execution and evidence collection. Test jobs are created on the Magnus server based on the test plan and configurations and then dispatched to the Magnus Client for script execution. ATSG has advanced orchestration keywords for posting events and waiting for events. The 'Post-event' sends message to other Jobs for decision making, while the 'Wait event' instructs the script to await instructions from other jobs. By combining Post and Wait events, users synchronize test execution flow across multiple clients or applications enabling seamless interaction between the smart glass device and the connected mobile device (Android or iOS based). Magnus communicates with test management systems, defect management systems and source management systems to manage the test process flow. It also offers the flexibility to integrate the new or custom tools into Magnus through the development of the connector for the respective tool.
- Electronic Device Automation Test Engine (eDAT): The eDAT engine supports testing of devices like smart glasses by providing the following libraries that help in the test automation of smart glass device testing:
 - Text to Speech engine (TTS): The TTS module converts the test input data from text to voice command that is ingested by the smart glass device.
 - Speech to Text (STT): The STT module captures the voice response from the smart glass device and converts it into the text string for validation against the expected test result.
 - Optical Character Recognition (OCR): The OCR module validates the scenarios that require text validation over the mobile screen. It is used to extract the text from the Android or iPhone device screen. The extracted text is validated against the expected test result to ascertain the test case pass or failure.
- Test report and defect logging: Upon completion of the smart glass test plan execution, a detailed test execution report is generated. The test report includes execution details and real-time device operations test evidence (screenshots) test logs and supports the PDF and HTML formats.
- CI/CD tools: Magnus provides support for integration with the CI/CD tools, such as Jenkins, which allows the jobs for the smart glass test scripts to be scheduled and executed without any manual intervention. The regression test suite execution is automatically triggered at the scheduled time through Jenkins.

Benefits

Achieving the test automation through the Magnus framework has provided the following benefits to the customer:

- Single script for functional and application localization testing with the ability to scale the automation scripts for additional language support with minimal effort.
- Detection of issues related to inconsistent behaviour of smart glass devices that were difficult to reproduce in manual testing.
- Common scripts for Android as well as iOS applications.
- Reduction in regression execution efforts by 35% over the manual approach.
- Localization test design effort reduction by 35%-40% for each new language script.
- Overall testing cost reduction by 20%.

Conclusion

Testing of smart glasses or VAs encompass complex setups involving multiple devices, like glasses, phones, speakers and more. To tackle the complexity of such setups, a solution supporting end-to-end testing of VA devices, test automation support for popular mobile devices based on Android and iOS, audio-to-text, text-to-audio conversion support and localization capabilities is essential to achieve sizable automation coverage. Magnus fulfils these requirements and has been successfully deployed for the regression test automation of a popular smart glass vendor.

References

- [smart glass Market Size, Share, Industry Report, Revenue Trends and Growth Drivers \(marketsandmarkets.com\)](#)
- [Magnus Test Automation Framework](#)

Author information



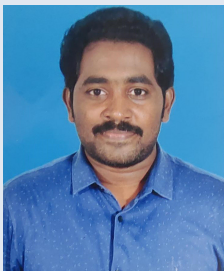
Subramanyam Palivela

Subramanyam Palivela has a Masters in Technology from the NIT Warangal. He is currently working as a Solutions Architect with the product testing team and has developed Automation Solutions using AI/ML and Deep Learning for Web, Desktop, Mobile and Microservices-based applications. He has also supported and developed Automation solutions for many Fortune 500 clients.



Abhishek Tomar

Abhishek Tomar works as Deputy General Manager with HCLTech. He is leading the PTS Solutions CoE Professional Services team and has vast experience in software development and testing for embedded systems in the Consumer Electronics domain. His interest lies in exploring Machine Learning methods to solve industry problems.



Srinivasareddy Vanukuri

Srinivasareddy Vanukuri holds a Bachelor of Technology in Electronics and Communication Engineering from JNTU. He is currently working as a Technical Architect for the device testing team. With over ten years of experience in the automation domain, he has been instrumental in developing and implementing advanced automation solutions across various industries.

HCLTech | Supercharging Progress™

HCLTech is a global technology company, home to 222,000+ people across 60 countries, delivering industry-leading capabilities centered around Digital, Engineering and Cloud powered by a broad portfolio of technology services and software. The company generated consolidated revenues of \$12.3 billion over the 12 months ended December 2022. To learn how we can supercharge progress for you, visit hcltech.com.

hcltech.com

