# APPMARK™ 2018
## WHITE PAPER

Revision 1.0

**Revision History**

1.0 – December 4, 2018 – Initial release

## About BAPCo

Business Applications Performance Corporation (BAPCo) is a non-profit consortium with a charter to develop and distribute a set of objective performance benchmarks for personal computers based on popular computer applications and industry standard operating systems.

For more information about BAPCo or a complete list of the current membership, see our website at http://www.bapco.com/.

# Table of Contents

# 1    Introduction

APPmark™ 2018 is a new cross-platform performance and battery life benchmark for touch-enabled laptops, tablets, and 2-in-1s, featuring workloads created by experts to represent modern real-world apps and usage models. It supports Windows, iOS and Android.

The benchmark features demanding web browsing, email reading and writing, photo and video content creation, and video playback workloads. These workloads were developed using the latest software development tools and current best practices for each platform, in order to best represent the capabilities of all devices.

APPmark gives commercial and government IT decision makers, media, channel buyers, consultants, consumers, and system and component manufacturers an objective, easy-to-use tool to evaluate the performance and battery life of devices across the wide range of activities that a user may encounter.

APPmark is designed for those who want to:
- Evaluate and compare touch-enabled laptop, tablet, and 2-in-1 computing devices for purchase consideration based on system performance and application responsiveness.
- Evaluate and compare devices running different operating systems.
- Evaluate devices to better optimize the battery life and performance of the system.
- Provide useful information to their audiences to assist in the evaluation and purchase of devices.

Unlike benchmarks that measure battery life or performance in isolation, APPmark measures battery life and performance simultaneously, showing how well a system design addresses the inherent tradeoffs between performance and power management.

Unlike synthetic benchmarks, which artificially drive components to peak capacity or attempt to deduce performance using a static simulation of application behavior, APPmark measures platform-level user experience, using the same platform APIs and services used by third-party app developers, real user workloads and real data sets to accurately measure how overall system performance impacts real-world user experience.

APPmark builds upon BAPCo's 27-year history of building benchmarks to evaluate platform technologies. Benchmarks designed by BAPCo are the result of cooperative development among companies representing the breadth of the

computing industry. They harness a consortium of knowledge to better reflect today's and tomorrow's emerging business trends.

This document describes the methodologies employed in the development of APPmark. For detailed instructions on how to install and run APPmark, please refer to the documentation provided in the app or on the BAPCo web site ([www.bapco.com](www.bapco.com)).

APPmark 2018 is available as a free download from the Windows Store, the iTunes App Store, and Google Play. Just search for "APPmark 2018" on your device, or follow these links:

- **Windows Store** (for Windows devices):
  - [https://www.microsoft.com/en-us/p/appmark-2018/9p2hr889gsf2](https://www.microsoft.com/en-us/p/appmark-2018/9p2hr889gsf2)
- **iTunes App Store** (for iPad):
  - [https://itunes.apple.com/us/app/id1307147493](https://itunes.apple.com/us/app/id1307147493)
- **Google Play** (for Android devices):
  - [https://play.google.com/store/apps/details?id=com.bapco.appmark.PUBLISH](https://play.google.com/store/apps/details?id=com.bapco.appmark.PUBLISH)

Alternative installation methods are explained on the BAPCo web site:

- [https://bapco.com/products/appmark-2018/](https://bapco.com/products/appmark-2018/)

## 2    BAPCo Development Process

BAPCo creates benchmarks in accordance with the BAPCo Development Process, a set of milestones and checkpoints collaboratively developed and agreed upon by the BAPCo membership.

Early in the process, prevailing business personal computer usage models are identified and grouped into scenarios according to their fit within a workflow. Applications are selected for each usage model on the basis of market research and technical feasibility.

BAPCo members then join together with expert application users in development sessions to collaboratively develop a workload specification for each scenario, defining each user/PC interaction which is to be simulated by the benchmark.

The goal of the development sessions is to produce representative business application workloads for the benchmark.  Each application workload consists of three elements: the input data set, the tasks performed on the input data set, and the generated output.  An example of generated output would be an image generated through an iterative process of steps to create a desired appearance. These three elements of the workload are chosen to represent the workflow of a user skilled in each given application.

After the workload specifications are created at the development sessions, BAPCo developers implement the workloads according to those specifications while satisfying benchmarking constraints to ensure the stability of the benchmark, the consistency of results, and the feasibility of implementation and distribution of the benchmark.

## 2.1 Milestone Overview

The BAPCo development process is divided into six major phases (Initialization, Design and Planning, Implementation, Validation, Characterization and Launch). Each phase consists of a series of milestones, some of which may be worked on concurrently.

The membership must vote to close each milestone. Once all the milestones within a phase are complete, the membership must vote to exit the phase and enter the next phase. BAPCo members work in a collaborative process where decisions regarding products are sometimes made by majority vote rather than unanimously.

The following is the list of the development phases and the corresponding milestones. Some of these milestones are explained in greater detail in the following sections, as noted in this list.

1. **Initialization Phase**
    a. Milestone 0 – Committee kickoff
    b. Milestone 1 – Benchmark market and customer analysis
    c. Milestone 2 – Product positioning and Customer Value Proposition
    d. Milestone 3 – Preliminary Marketing Requirements Document
    e. Milestone 4 – Select product name
    f. Milestone 5 – Final Marketing Requirements Document
2. **Design and Planning Phase**
    a. Milestone 6 – Preliminary Engineering Requirements Document
    b. Milestone 7 – Usage model selection *(see section 2.2)*
    c. Milestone 8 – Application selection *(see section 2.3)*
    d. Milestone 9 – Define member resource commitments
    e. Milestone 10 – Define development infrastructure and tools
    f. Milestone 11 – Define scoring methodology *(see section 2.4)*
    g. Milestone 12 – Define application licensing requirements
    h. Milestone 13 – Plan and execute workload development sessions *(see sections 2.5, 2.6)*
    i. Milestone 14 – Define product release criteria
    j. Milestone 15 – Create implementation schedule
3. **Implementation Phase**
    a. Milestone 16 – Determine reference system *(see section 2.7)*
    b. Milestone 17 – Software engineering *(see section 2.8)*
4. **Validation Phase**
    a. Milestone 18 – Validation testing
    b. Milestone 19 – Define risk management plan

5. **Characterization Phase**
    a. Milestone 20 – Characterization testing
6. **Launch Phase**
    a. Milestone 21 – Product pricing
    b. Milestone 22 – Pre-launch materials
    c. Milestone 23 – Release to Manufacturing vote and sign-off
    d. Milestone 24 – Early press evaluation
    e. Milestone 25 – Duplicate and distribute media
    f. Milestone 26 – Post-launch materials

## 2.2 Usage Model/Scenario Selection

After the goals of the benchmark are defined at a high level in the Marketing Requirements Document (MRD) and Preliminary Engineering Requirements Document (ERD), usage models are chosen for inclusion in the benchmark and related usage models are grouped into scenarios.

For APPmark, BAPCo used market studies on device usage patterns to choose a wide variety of usage models relevant to users.

Those scenarios were then grouped into the following scenarios. For more detailed descriptions of each scenario, please see section 2.6.

### Productivity Scenario

The Productivity scenario models productivity and online activities, such as browsing professional and consumer web pages, uploading image files to a server, reading, creating and deleting emails, and taking short surveys and analyzing results. The web browsing portion of the scenario uses a variety of web pages based on the most popular sites for professional networks, social media, photo sharing and uploading service, news/finance/sports news outlets, blogs, banking services and online shopping. The email portion of the scenario includes compression/decompression and encryption/decryption operations to reflect common information security practices.

### Creativity Scenario

The Creativity scenario models media creation and consumption activities, such as viewing and manipulating photos, adding a variety of photo effects, performing High-Dynamic-Range (HDR) imaging operations, stitching existing 1080p and 4K video clips into new movies, and playing back these newly created movies.

### Video Playback Scenario

The Video Playback scenario consists of playing high definition video from local storage. This scenario contains 9 minutes of 1080p H.264 video content, encoded at an average bitrate of 8 Mbps at 29.97 fps. There is no performance rating for the Video Playback scenario.

## 2.3 Application Selection

After the usage models have been collected into scenarios, applications are chosen for the scenarios on the basis of market research and technical feasibility.

For APPmark, BAPCo reviewed market-leading store-based, touch-driven applications for Windows, iOS, and Android for each chosen usage model. Then, in order to accurately reflect real world application behavior within the restrictions of modern store-based runtime environments, BAPCo created new applications for these usage models from scratch, taking inspiration from the design and behavior of the market leaders.

These "application models" were created separately for Windows, iOS, and Android, built from the ground up using the same development tools, compilers, platform APIs and programming techniques an app developer would use for each platform.

Because these application models were purpose-built for cross-OS benchmarking, they have comparable user interfaces, features and output across all supported platforms. And because they rely upon the high-level functionality exposed by each platform, they take advantage of the efficiencies and optimizations intrinsic to each platform. See section 2.8.2 for more details.

## 2.4 Scoring Methodology

Next, BAPCo decides the types of results that will be produced by a benchmark and the scoring methodology that determines how those results are calculated.

Importantly, BAPCo determines the scoring methodology before determining the content of the workloads, which helps ensure that a methodology is chosen for its ability to generate results that correspond to user experience, not for the results it produces on a pre-determined set of workloads.

For APPmark, BAPCo evaluated the merits of a variety of scoring methodologies and chose a methodology on the basis of how it met the following criteria:

- The scoring methodology should give expected results:
  - The resulting score should differentiate between systems with different performance.
  - The resulting score should be repeatable and not have high variation.
  - The resulting score should not be affected by benchmark artifacts, such as the number of tasks within a scenario or resource utilization by the benchmark itself.
- The relative performance between any two systems should not be affected by the selection of the reference system.
- The scoring methodology should be easy to understand.

### 2.4.1 Performance Rating

APPmark measures system performance by measuring the time a device takes to complete tasks (the "response time") using representative application models, user inputs and workloads. In the APPmark scoring methodology, these task response times are used to generate performance ratings that reflect the user experience. The faster a device responds to the application workloads in APPmark, the higher its APPmark performance ratings will be. For more information on how task response times are measured, see section 2.8.

The performance rating is given in relative terms, with respect to a reference system. It is calculated in such a way that a device with performance equivalent to this reference system for a given scenario will have a performance qualification rating of 1,000. A device twice as fast as the reference system on a given scenario (or, equivalently, that responds in half the time on average) will have a performance qualification rating of 2,000.

A complete APPmark run will output two scenario performance ratings (one for Productivity and the other for Creativity), plus an overall performance rating.

> **Scenario Rating:** Each scenario has a performance rating calculated by taking the sum of the response times of tasks in that scenario as performed on the test system and then comparing it with the sum of those same task response times as performed on the reference system (see section 2.7). The reference sum is divided by the measured sum on the test system and multiplied by 1,000. The result is then rounded to the nearest integer.

> **Overall Rating:** The APPmark Overall Rating is calculated by taking the geometric mean of all the scenario ratings (prior to rounding). The result is then rounded to the nearest integer. If either scenario score is missing, no Overall Rating can be calculated.

A performance qualification rating $p_i$ is first calculated for each successfully completed iteration of a scenario (excluding any conditioning iteration and partial iterations):

$$p_i = 1000 \left(\frac{s_r}{s_t}\right)$$

Where:
$s_r$ = the sum of the response times of tasks as performed on the reference system, explained below.
$s_t$ = the sum of the response times of tasks as performed on the test system.

The reference sum $s_r$ is a fixed value BAPCo calculates by performing three runs of the scenario—each including a conditioning run—on a reference system (see section 2.7). For each successfully completed iteration among all three runs (excluding any conditioning iterations and partial iterations), the sum is taken of the response times of all the tasks in that iteration. $s_r$ is the arithmetic mean of all those iteration sums.

$p_i$ is rounded to the nearest 1/100th. For display purposes only, this is further rounded to the nearest integer.

Then the performance qualification rating $p_s$ is calculated by taking the arithmetic mean of the performance qualification ratings $p_i$ for all successfully completed scenario iterations $i = 1, \ldots, n$:

$$p_s = \frac{1}{n}\sum_{i=1}^{n} p_i$$

$p_s$ is rounded to the nearest integer.

### 2.4.2 Battery Life Rating

The battery life rating is calculated by measuring the actual battery duration observed (from a fully charged state to a fully depleted state), in whole minutes, while running all three scenarios in succession, interspersed with idle periods to simulate user idle time (as described in section 2.9.1), then repeating until the battery dies.

A battery life rating is only produced if the user selects the "Battery Run" option when starting the test, which will in turn select all scenarios for the test. APPmark can only produce battery life results for tests in which all scenarios were completed successfully.

### 2.4.3 Comparing Results Across Platforms

One of the key features of APPmark 2018 is the comparability of its performance and battery life ratings across different operating systems. The workloads in APPmark 2018 were designed to accomplish equivalent work across all supported platform OSes (see section 2.8.2), and all versions of APPmark 2018 use the same reference sum for scoring (see section 2.7).

APPmark 2018 ratings are not directly comparable to results obtained from any other benchmark, including any versions of TabletMark.

## 2.5 Workload Development Sessions

Once the usage models, scenarios, applications, and scoring methodology for the benchmark are decided, BAPCo members and application experts meet to create the application workloads that will be used in the benchmark.

For APPmark 2018, a workload development session was steered that consisted of a face-to-face meeting with representatives from BAPCo member companies and expert application users who had professional experience with the applications chosen for the benchmark.

In the workload development sessions, the experts take the lead, weaving the usage models supplied by BAPCo into a storyboard of user interactions with a series of applications. Each user/device interaction is written down in a workload specification, the specification later used to automate the workloads.

At the end of the workload development sessions, BAPCo comes away with a detailed workload specification for each of the benchmark scenarios and all of the input data sets needed to reproduce the workloads created at the sessions.

### 2.5.1. Input Data Set

Frequently in the sessions, the experts need raw digital content to serve as input data for a workload. Examples of such content might include a video to transcode, an email to modify, or photos to manipulate. When experts need such content, care is taken to ensure that they use something that is functionally representative of content they might use or encounter professionally.

For instance, if pictures are needed in order to create a photo slideshow, an expert might walk outside and take pictures using the same camera equipment he/she uses professionally. If a song track is needed as the background music for creating a movie, an expert might purchase a stock track from his/her usual online resource. Like the user interactions, all of these source materials are captured at the development session and used later in the development of automated workloads.

## 2.6    Scenario Workload Descriptions

The scenario workloads that were created based on the usage models built upon at the workload development sessions for APPmark 2018 are described below:

**Productivity Scenario**

**Web browsing:** The web workload models a user browsing a variety of professional and personal websites, scrolling through web pages, pausing at times to read page contents, uploading and viewing images, viewing video clips, performing online shopping, bank services and animated advertisements embedded into the pages. These websites are based on popular websites for professional networks, social media, photo sharing services, news outlets, finance and sports sites, online retailers, bank sites, blogs and BAPCo's own website.

**Email:** The email workload models a user opening an email client, looking at a list of email messages, opening messages, pausing to read messages, opening and saving attachments, searching through emails, sorting the messages and composing new emails.

**Survey:** The survey workload models a user filling out a survey and an administrator calculating and viewing the statistical results of the survey.

**Creativity Scenario**

**Photo editing:** The photo workload models a user loading a set of high-resolution pictures, adjusting them for optimal exposure, brightness, color saturation, rotation, sharpen, vignette, contrast, grayscale, temperature and crop. Finally, a high dynamic range (HDR) imaging operations are performed on a set of exposure-bracketed photos. The resulting HDR images show clear details across the entire brightness spectrum, from the lightest to the darkest areas.

**Video:** The video workload models a user starting a video editing application and loading a set of existing video clips (including 1080p and 4K content) and audio clips from the library. Some of the clips are selected, added to the video timeline, trimmed, and merged into a new video, adding a new background music track in the process which is shown in the audio timeline. The resulting movie file is then transcoded and played back in the application. The user proceeds to select a second set of video clips and creates another video timeline. This new movie is then transcoded and played back in the application.

### Video Playback Scenario

**Video Playback:** The video playback scenario models a user watching nine minutes of 1080p high definition video. The video formats used are representative of what can be found on popular video sharing sites.

## 2.7    Reference System

The reference system (sometimes also referred to as a "calibration system") is a system chosen as a reference point for all other benchmark performance results.

For APPmark 2018, BAPCo chose the configuration below for its wide availability and its representation of a typical mainstream touch-driven device at the time of release of the benchmark.

APPmark 2018 calculates ratings in such a way that a device with performance equivalent to this reference system for a given workload will have a performance rating of 1,000. A device twice as fast as the reference system on a given workload (or, equivalently, that responds in half the time on average) will have a performance rating of 2,000. This holds for both overall ratings and scenario ratings.

The reference system for APPmark 2018 has the following configuration:
- **Device Model:** iPad (2018 Model)
- **Processor:** A10 Fusion
- **Memory:** 2 GB
- **Storage:** 32 GB
- **Display:** 9.7" (2048 x 1536)
- **Graphics:** CPU-integrated
- **Networking:** Wi-Fi model (no LTE)
- **Operating System:** iOS 12.0
- **Operating System Language:** US English

A fresh operating system is installed on the system.

This reference system serves as the reference point for the performance ratings produced by all supported platforms of APPmark 2018 (Windows, iOS, and Android). The reference system is only relevant to performance ratings; battery life is expressed in absolute minutes of battery life, not in relation to the reference system.

The reference sum for each scenario is obtained according to the following process:
- Run APPmark 2018 on the reference system (conditioning run disabled, battery run disabled, 3 iterations)
- Calculate an overall sum for each of the 3 iterations by taking geometric mean of all scenario sums for that iteration

- Identify the median iteration by finding the iteration with the median overall sum
- The reference sum for each scenario is taken from that median iteration

Please note that the minimum requirements to run APPmark 2018 are listed in Appendix A and have no relationship to the reference system.

## 2.8    Benchmark Implementation

Once the workload specifications have been created, BAPCo begins the important work of translating the workload specifications into an automated benchmark.

### 2.8.1 Benchmark Framework

APPmark 2018 is built upon simulated user behaviors and interactions, using controls like buttons, text input boxes, and menus to navigate applications. (See Appendix B for screenshots of the benchmark in action.)

To ensure that APPmark 2018 has deterministic behavior, BAPCo uses a framework to collect system information, run workloads within a series of application models for each scenario, record performance measurements, calculate performance ratings, and display test results. The framework is kept lightweight, consuming a minimal amount of memory and compute resources, in order to ensure that performance measurements reflect the workload performance and do not include overhead from the framework.

The fundamental performance unit in APPmark 2018 is response time. Response time is defined as the time it takes a device to complete a task. A task can be initiated by a mouse click, a keystroke, or programmatically. The duration of each task is measured by the framework. Examples of tasks include loading documents, finding text in a document, performing an image manipulation, and transcoding a video.

The framework measures the duration time for each task. Simulated pauses mimicking user behaviors (e.g., reading a web page or email) are excluded from the measurement.

### 2.8.2 Cross-Platform Development Methodology

As explained in section 2.3, application models and workloads in APPmark 2018 were implemented from the ground up using standard development tools and practices for each target platform. They make full use of the APIs provided by each platform.

In order to ensure results remain comparable across these disparate platforms, the workloads were developed in a way that ensures all platforms:
- perform comparable work (e.g., using similar video codecs and bitrates, using the same encryption algorithm and security level),
- operate on comparable input data (e.g., similar file size, equivalent perceived image/video/audio quality),

- produce comparable output data (e.g., similar file size, equivalent perceived image/video/audio quality).

To illustrate with an example, the Productivity scenario includes email encryption/decryption operations. To accomplish this in a representative fashion on each platform, the following libraries were used:

| Platform | Library |
|----------|---------|
| Windows | Windows.Security.Cryptography |
| iOS | CCCrypt |
| Android | javax.crypto.Cipher |

**Table 2.8.2.1: Email Crypto Libraries by Platform**

To ensure comparability of results, the workload utilizes the same 128-bit AES-CBC encryption on all platforms.

For another example, the Creativity scenario includes video encoding operations that utilize the following libraries:

| Platform | Library |
|----------|---------|
| Windows | DirectX Video Acceleration (DXVA) |
| iOS | AV Foundation |
| Android | OpenMAX |

**Table 2.8.2.2: Video Codec Libraries by Platform**

Though the library varies from one platform to the next, the codec is consistent across all platforms. For instance, in the case of the 4K video encoding operation, all platforms use the following video encoding parameters:
- H.264/AVC codec
- 3840x2160 resolution
- 29.97 progressive frames per second
- 15 Mb/sec

This methodology was chosen to balance the desire to reflect real-world platform performance while ensuring the platforms are all being asked to perform work that is equivalent from the user's perspective.

## 2.8.3 Development Tools

The information in this section is current as of the initial release of APPmark 2018, but is subject to change in future releases as development tools are updated and

to support future devices and operating system releases. Please contact support@bapco.com with any questions.

Generally, the workloads were built using the following software development tools:

| Platform | Language | Tools | Compiler Options |
|---|---|---|---|
| Windows | C# | Visual Studio 2017 | Default |
| iOS | Objective C | Xcode 9.4.1 | Default (-Os and Automatic Reference Counting) |
| Android | Java | Android SDK 28, NDK 17.1, OpenJDK 1.8.0 | Default |

**Table 2.8.3.1: Development Tools by Platform**

For usage models that required functionality not available in platform APIs, the functionality was implemented using native C code, which was shared across all platforms. Only the HDR workload in the Creativity scenario was implemented in this way.

The HDR workload was built using the following tools and settings:

| Platform | Tools | Compiler Options |
|---|---|---|
| Windows | Visual Studio 2017 | default (/O2) + /Oi (generate intrinsic functions) |
| iOS | Xcode 9.4.1 | default (-Os and Automatic Reference Counting) |
| Android | Android NDK 17.1 | -O3 (enables many general optimizations)<br>--ftree-vectorize (implied by -O3)<br>--ffast-math (enables common math optimizations for code that doesn't require strict IEEE compliance)<br>--fomit-frame-pointer (implied by -O3, reduces memory consumption to support lower RAM—e.g., 1 GB—devices) |

**Table 2.8.3.2: Native Development Tools by Platform**

The workloads are compiled for the following platform/architecture combinations (using the nomenclature of each platform's build tools):

| Platform | Architecture Targets |
|----------|---------------------|
| Windows | x64, ARM |
| iOS | armv7, arm64 (packaged together in one .ipa) |
| Android | armeabi-v7a, arm64-v8a, x86, x86_64 (packaged together in one .apk) |

**Table 2.8.3.3: Architecture Targets by Platform**

## 2.8.4 Output Validation

As an additional check that the scenario workloads ran correctly and produced the desired output, APPmark 2018 checks various attributes of output files produced by the workloads to ensure they fall within acceptable bounds. These checks are performed each iteration.

## 2.9 Workload Characterization

Once the scenario workloads are implemented and validated against the workload specifications created at the development sessions, BAPCo members then run the benchmark on a wide variety of systems to ensure that the benchmark produces results that are valid, representative, and reproducible.

During this process, BAPCo members share data, raise concerns, and suggest workload changes. Any workload change requires a majority vote of the committee.

APPmark 2018 is a tool for measuring both performance and battery life. It is important that its workloads and power profiles of its activities are reasonably representative of user experiences and expectations. BAPCo members work together to arrive at an estimation of a representative level of user activity, and acknowledge that individual user experience could vary from the results reported by the benchmark.

The battery life reported by APPmark 2018 is an approximation of the battery life a user who performs that same workloads would expect from the same device.

### 2.9.1 User Idle Time

One way BAPCo members adjust the power profile of the workload is by determining an appropriate proportion of user idle time relative to active time.

Users often leave their systems idle for a period of time between sessions of active use. When measuring battery life, APPmark 2018 simulates this behavior by interspersing occasional periods of user inactivity throughout the workload. These idle periods are only relevant for battery life tests. In order to minimize test runtime, the idle periods are bypassed for tests that produce only performance results.

Inclusion of these idle periods for battery life tests better models real-world device usage and allows the hardware and software power management features of the device to behave in a realistic manner.

Furthermore, the scenario workloads themselves are paced at a human speed, pausing briefly between individual tasks to simulate the user reading, inspecting the output of a previous task, or considering the next task.

APPmark 2018 measures battery life by requiring that the battery be fully charged (at least 95%) prior to testing, and then performing the following activities in a loop (with each loop referred to as an "iteration") until the battery dies:

1. Productivity scenario (duration varies depending upon device performance)
2. User Idle period (duration is fixed, 3 minutes)
3. Creativity scenario (duration varies depending upon device performance)
4. User Idle period (duration is fixed, 3 minutes)
5. Video Playback scenario (duration is fixed, 9 minutes)
6. User Idle period (duration varies depending upon device performance, extends iteration to 50 minutes total)

Different systems may take different amounts of time complete the Productivity scenario (activity 1) and the Creativity scenario (activity 3). To compensate for this, APPmark 2018 waits at the end of the iteration (activity 6) to ensure the iteration has a fixed duration of 50 minutes. This ensures that all systems have done the same amount of work after each 50 minutes of battery life testing.

BAPCo sampled a variety of systems that met the APPmark 2018 minimum system requirements and found they all finished an iteration of the workload within 50 minutes. In the hypothetical case that a device does not finish an iteration within 50 minutes, the benchmark will allow the iteration to run to completion (past the 50-minute mark) and then omit the final idle period (activity 6) for that iteration, immediately beginning the next iteration.

During the idle periods, APPmark 2018 shows a solid-colored screen and prevents the device from entering any power state that might shut down the display or reduce its brightness (e.g., display dimming/standby, system standby/hibernation).

# 3    Workload Characteristics

This section provides data illustrating the workload characteristics of APPmark 2018.

## 3.1    Sensitivity Analysis Methodology

The series of tables below shows the sensitivity of APPmark 2018 to different system characteristics, including the amount of system memory, number of CPU cores, type of storage device, and display resolution.

Within each study only one system component (e.g., memory) is varied. All the other system components are held constant. To best illustrate the sensitivity, the minimal configuration is chosen as a baseline for each table and the ratings for the other configurations are shown as the percentage difference relative to that baseline.

A high-end Windows-based desktop system has been chosen as the basis for these studies for two reasons:
- Using a Windows-based desktop configuration offers the most flexibility in component selection and configurability, allowing many vectors of component performance to be measured by testing a wide variety of components and configurations.
- Using a high-end system with high-end components places the focus on one component at a time (the independent variable) without the performance sensitivity being heavily limited by any of the other components (the control variables).

Note that the Windows performance power slider may have a significant impact on both performance and battery life results. As this sensitivity study was performed on a desktop system, the performance power slider is not present in the OS configuration options and not relevant to the results of this study.

Also consider that these component sensitivities may vary somewhat from one system configuration to the next.

System configuration (unless otherwise specified):
- **Motherboard:** ASUS® PRIME Z370-A (Firmware 1601)
- **CPU:** Intel® Core™ i7-8700K (3.0 GHz, 6 cores, HT and Turbo disabled)
- **RAM:** 16 GB DDR4-3200 (dual channel, 8 GB sticks)
- **Graphics:** EVGA® GeForce® GTX 1080 Ti
- **Storage:** Intel® Optane™ SSD 900p (280 GB, PCIe card)

- **Operating System:** Microsoft® Windows® 10 "Redstone 4" (Build 17134.407)
- **Display Resolution:** 1920 x 1080 (1080p)

The OS and motherboard firmware were the latest available as of November 14, 2018, and the results of this study reflect the security updates and mitigations included in those releases.

For the study, the following components/settings were changed to isolate the benchmark sensitivities to each subsystem.

- **CPU Frequency**
  - 1.5 GHz
  - 2.0 GHz
  - 2.5 GHz
  - 3.0 GHz
- **CPU Core Count**
  - 2 cores
  - 4 cores
  - 6 cores
- **RAM**
  - 2 GB DDR4-3200 (single channel, OS-limited using one 8 GB stick)
  - 4 GB DDR4-3200 (single channel, OS-limited using one 8 GB stick)
  - 8 GB DDR4-3200 (single channel)
  - 16 GB DDR4-3200 (dual channel, 8 GB sticks)
- **Graphics**
  - ZOTAC® GeForce® GT 730
  - GIGABYTE® GeForce® GTX 1050
  - EVGA® GeForce® GTX 1080 Ti
- **Storage**
  - Seagate® Momentus Hard Drive (500 GB capacity, 2.5", 5400 RPM)
  - WD® Desktop hard drive (1 TB capacity, 3.5", 7200 RPM)
  - Intel® SSD 545s (256 GB capacity, 2.5")
- **Display Resolution**
  - 1920 x 1080 (1080p)
  - 2560 x 1440
  - 3840 x 2160 (4K)

In the tables below, "PR" represents the Productivity Scenario Rating, "CR" represents the Creativity Scenario Rating, and "Overall" represents the Overall Rating. All scores are given as percentages deltas relative to the baseline component.

## 3.2 Sensitivity to CPU Frequency

| CPU Frequency | PR | CR | Overall |
|---|---|---|---|
| 1.5 GHz | Baseline | Baseline | Baseline |
| 2.0 GHz | +32.9% | +23.1% | +27.9% |
| 2.5 GHz | +69.1% | +31.6% | +49.2% |
| 3.0 GHz | +103.6% | +44.0% | +71.2% |

Table 3.2.1: Sensitivity to CPU Frequency

## 3.3 Sensitivity to Number of CPU Cores

| CPU Cores | PR | CR | Overall |
|---|---|---|---|
| 2 core | Baseline | Baseline | Baseline |
| 4 cores | +5.9% | -0.8% | +2.5% |
| 6 cores | +6.6% | -0.3% | +3.1% |

Table 3.3.1: Sensitivity to Number of CPU Cores

## 3.4 Sensitivity to RAM

| Memory | PR | CR | Overall |
|---|---|---|---|
| 2 GB (single-channel) | Baseline | Baseline | Baseline |
| 4 GB (single-channel ) | +2.4% | +4.0% | +3.2% |
| 8 GB (single-channel) | +2.3% | +6.4% | +4.3% |
| 16 GB (dual-channel) | +3.0% | +7.3% | +5.1% |

Table 3.4.1: Sensitivity to RAM

## 3.5 Sensitivity to Graphics

| Graphics | PR | CR | Overall |
|---|---|---|---|
| ZOTAC GeForce® GT 730 | Baseline | Baseline | Baseline |
| GIGABYTE GeForce® GTX 1050 | -0.73% | +137.1% | +53.4% |
| EVGA GeForce® GTX 1080 Ti | -0.07% | +141.8% | +55.4% |

Table 3.5.1: Sensitivity to Graphics

## 3.6   Sensitivity to Storage

| Storage | PR | CR | Overall |
|---|---|---|---|
| Seagate® 5400 RPM HDD | Baseline | Baseline | Baseline |
| WD® 7200 RPM HDD | +5.5% | +4.0% | +4.7% |
| Intel® 545s SSD | +8.6% | +7.4% | +8.0% |

**Table 3.6.1: Sensitivity to Storage**

## 3.7   Sensitivity to Display Resolution

| Display Resolution | PR | CR | Overall |
|---|---|---|---|
| 1920 x 1080 | Baseline | Baseline | Baseline |
| 2560 x 1440 | -1.8% | -0.9% | -1.3% |
| 3840 x 1260 | -2.6% | -0.0% | -1.3% |

**Table 3.7.1: Sensitivity to Display Resolution**

## 3.8    Timeline of Activities

This chart shows the time spent in various workload and idle activities during one 50-minute iteration of APPmark 2018 on the reference system (see section 2.7). The durations of these activities and their proportions relative to each other may vary from one system configuration to the next.
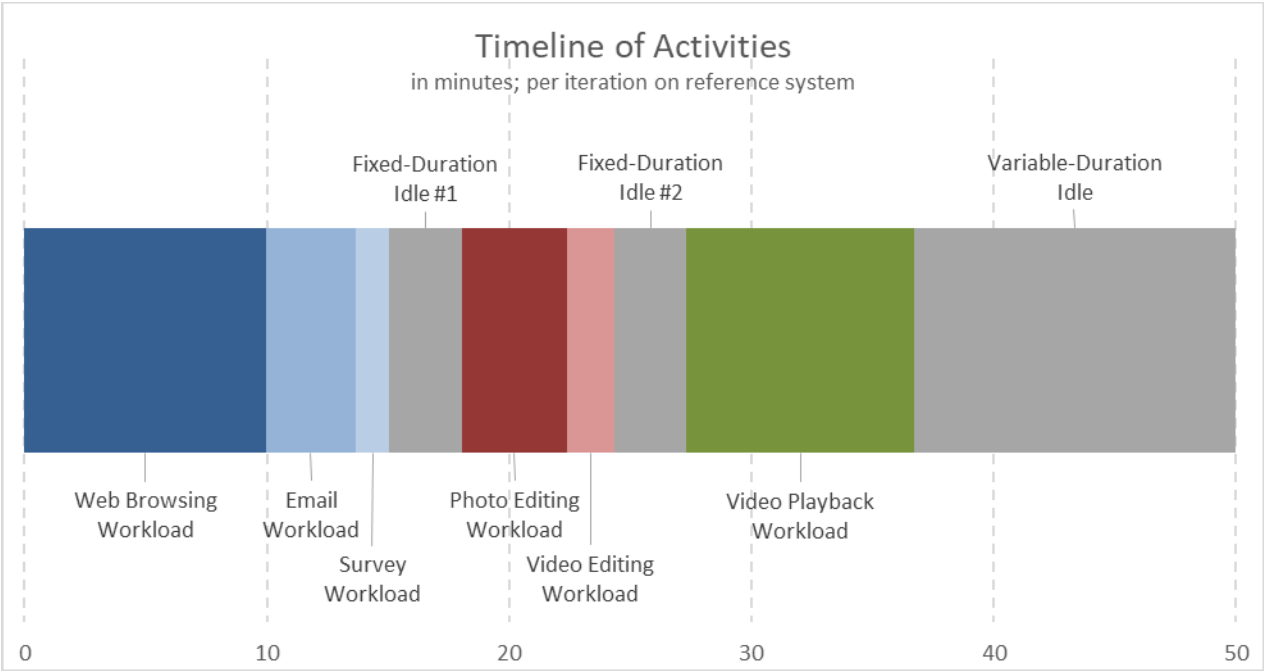


**Table 3.8.1: Timeline of Activities**

## 3.9    Time Spent Per Activity State and Scenario

The following charts show the relative time spent in each user activity state and scenario during a full APPmark 2018 test on the reference system (see section 2.7). These proportions may vary from one system configuration to the next.

## Time Spent Active and Idle
% time spent; on reference system
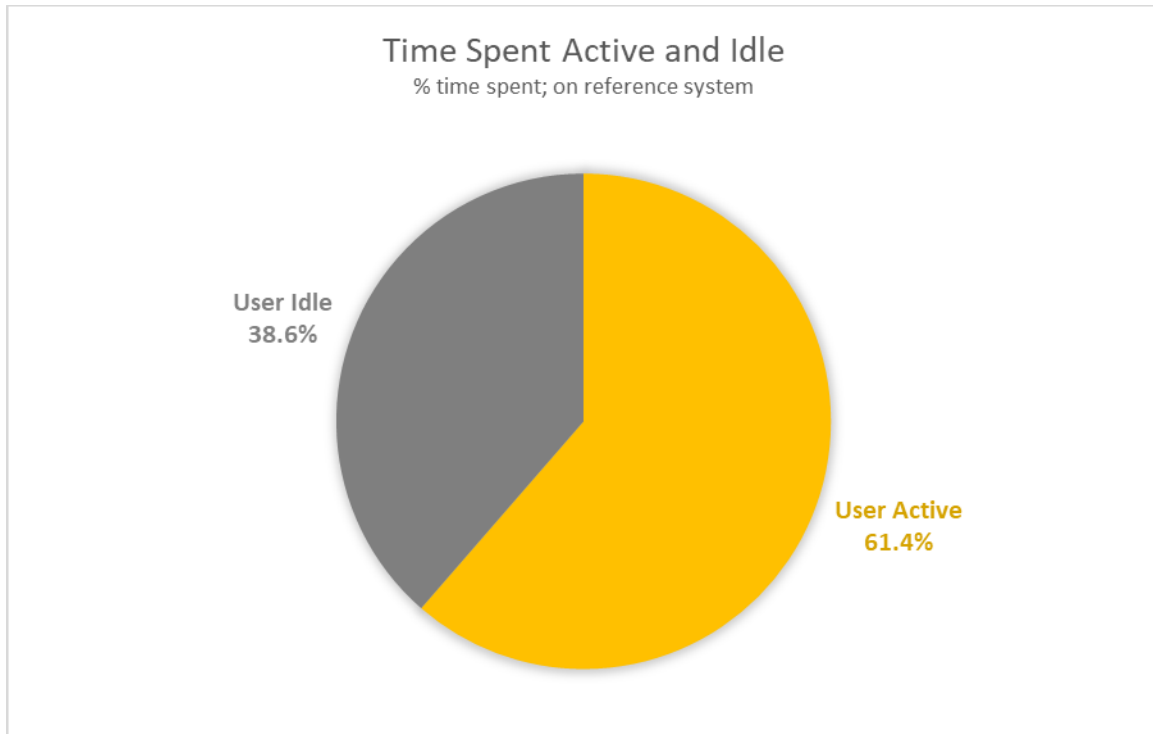


**Table 3.9.1: Time Spent Active and Idle**

## Time Spent Per Scenario
% time spent; on reference system
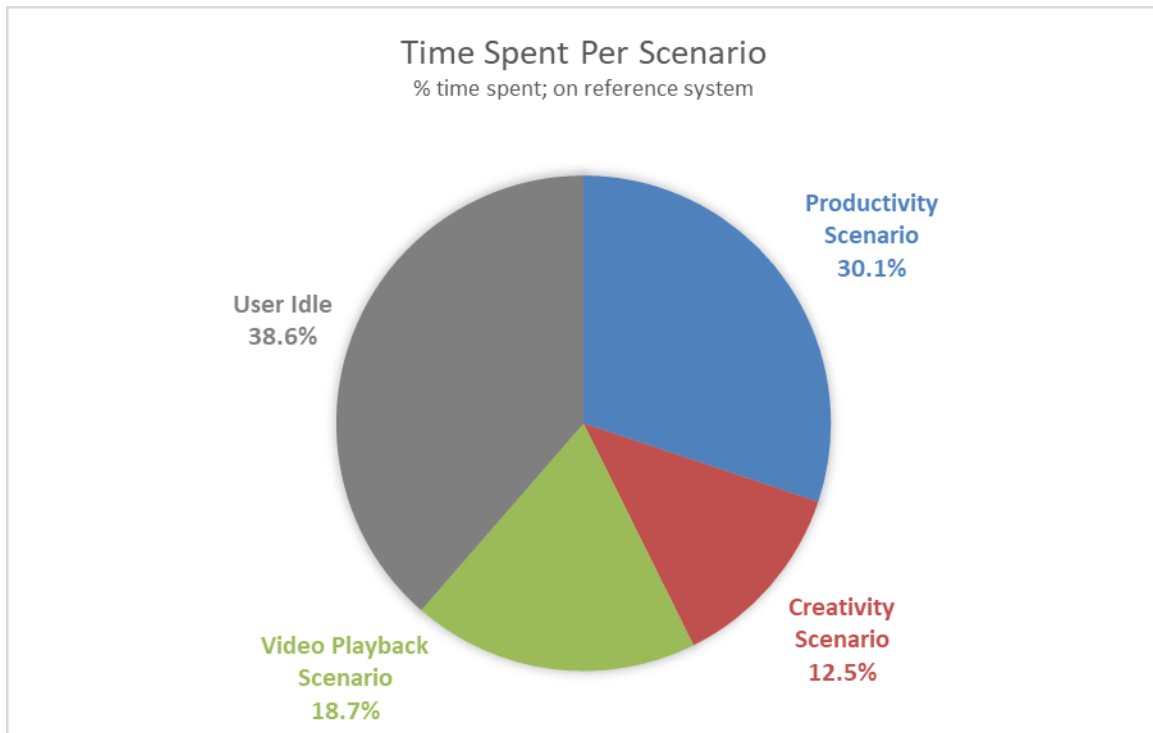


**Table 3.9.2: Time Spent Per Scenario**

# Appendix A: Minimum System Requirements

APPmark 2018 supports devices that meet these minimum requirements:

| Windows |
| --- |
| Windows 10 or newer |
| 1.3 GHz dual core processor (x86 or ARM) |
| 2 GB system memory |

**Table A.1: Minimum System Requirements for Windows**

| iOS |
| --- |
| iOS 10 or newer |
| A9 processor or newer |
| 1 GB system memory |

**Table A.2: Minimum System Requirements for iOS**

| Android |
| --- |
| Android 6.0 or newer |
| Dual core processor (ARM or x86) |
| 1 GB system memory |

**Table A.3: Minimum System Requirements for Android**

# Appendix B: Application Workload Datasets

This section shows the types of data inputs used by the workloads in each scenario.

| Workload | Input File Types | Key File Attributes |
|---|---|---|
| Web Browsing | .html, .jpg, .png, .gif, .svg, .js, .css | 40 web page bundles, totaling 137 MB in 873 files |
| Email | .txt | 1 encrypted and compressed email database file containing 16000 emails, totaling 39 MB, and 1 small index text file |
| Survey | .json | 1 web page bundle, totaling 278 KB in 7 files |

**Table B.1: Productivity Scenario Workload Datasets**

| Workload | Input File Types | Key File Attributes |
|---|---|---|
| Photo Editing | .jpg | 13 .jpg image files, totaling 17 MB, with these attributes:<br>• 13 at 5-9 megapixel resolution |
| Video Editing | .mp4, .aac, .mp3 | 6 .mp4 video files, totaling 166 MB, with these attributes:<br>• 4 at 16 seconds long, 29.97 fps, 1080p, 14-16 Mb/sec, H.264-encoded<br>• 2 at 11 seconds long, 29.97 fps, 4K, 16 Mb/sec, H.264-encoded<br>2 audio files, totaling 21.1 MB (one AAC, one .mp3) |

**Table B.2: Creativity Scenario Workload Datasets**

| Workload | Input File Types | Key File Attributes |
|---|---|---|
| Video Playback | .mp4 | 1 .mp4 video files, 57.5 MB, with these attributes:<br>• 60 seconds long, 30 fps, 1080p, 8 Mb/sec, H.264-encoded |

**Table B.3: Video Playback Scenario Workload Datasets**
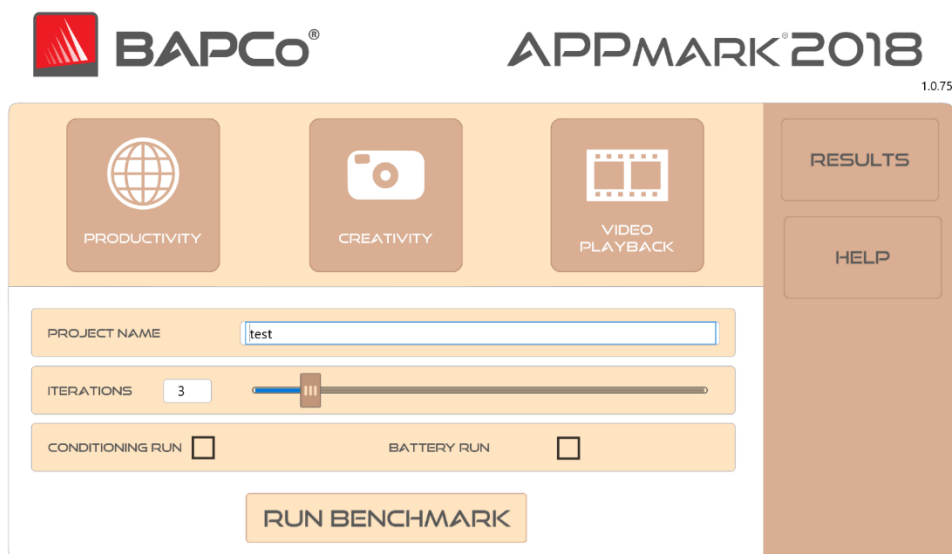
## Appendix C: Screenshots



**Figure C.1.1: Main User Interface (Windows)**
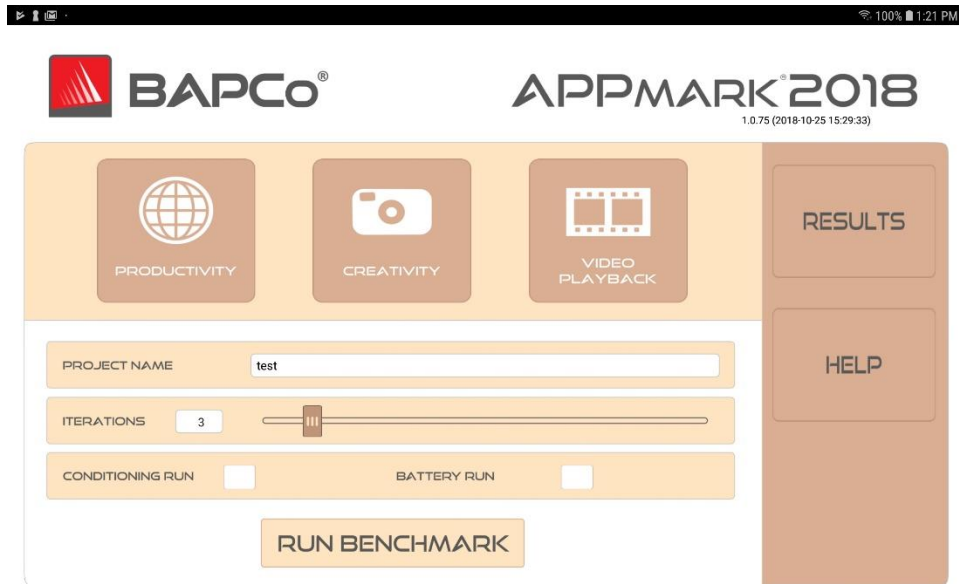


**Figure C.1.2: Main User Interface (iOS)**

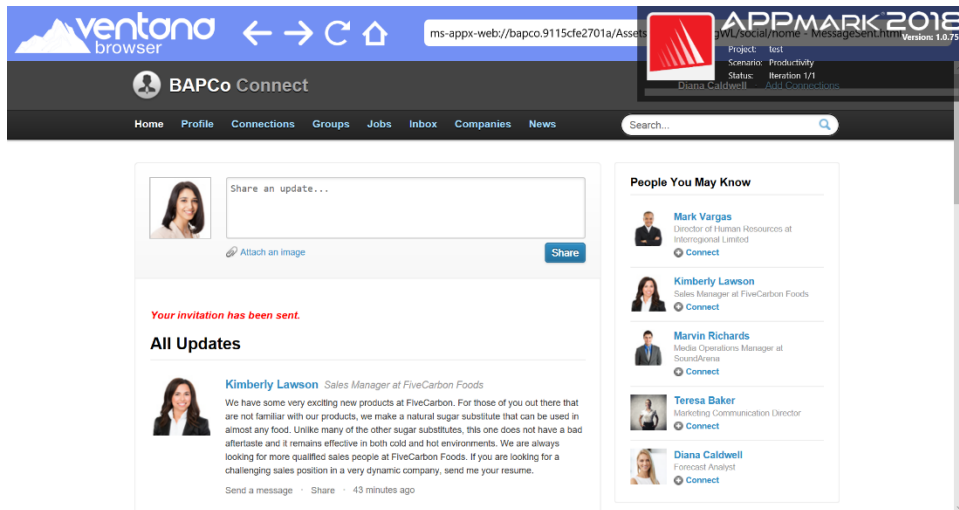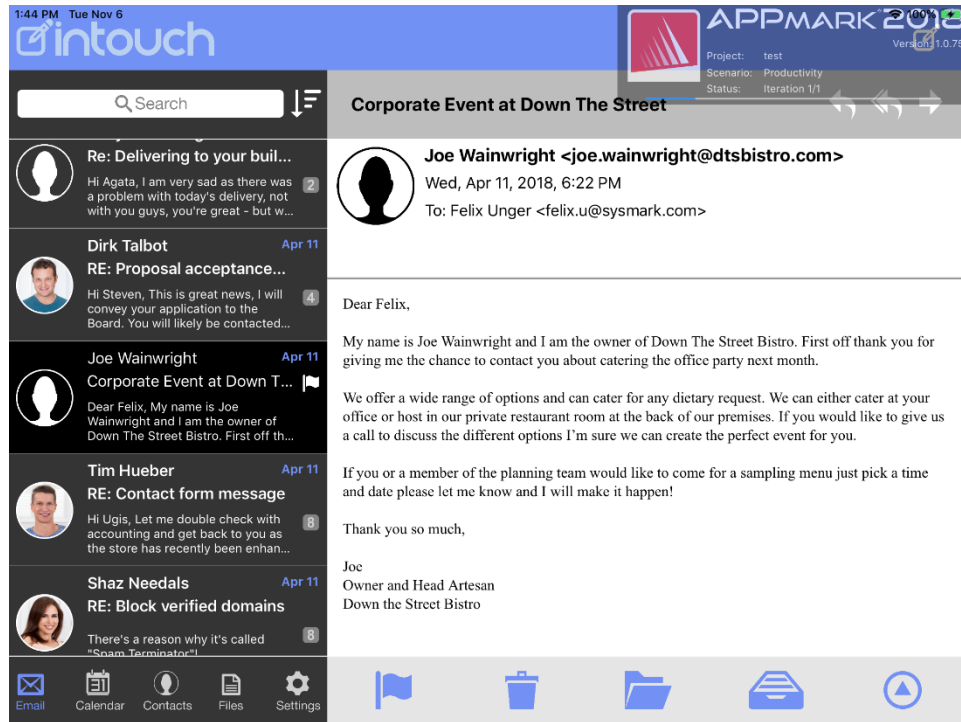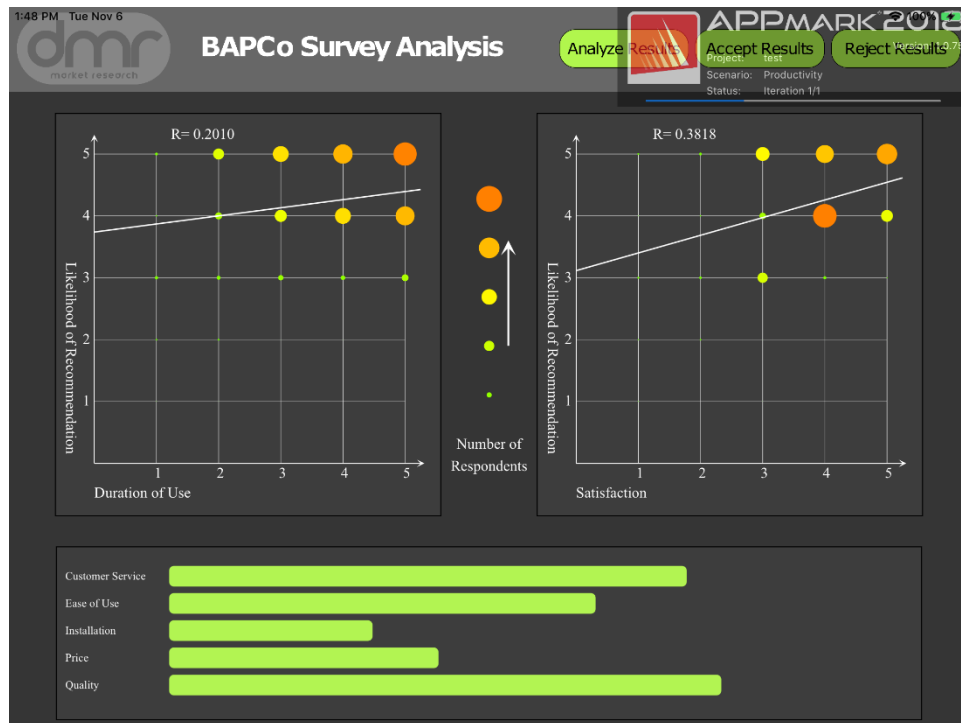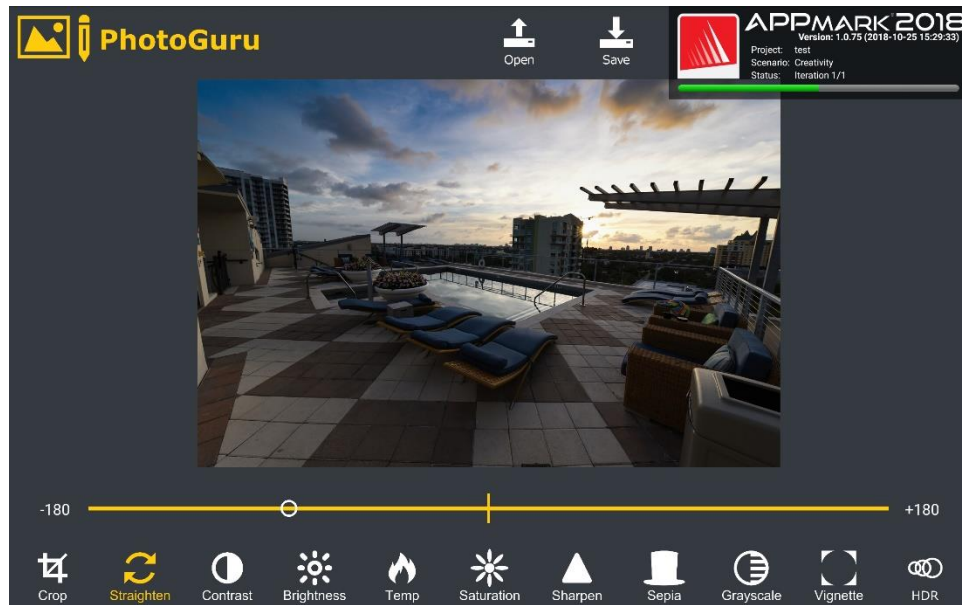**Figure C.1.3: Main User Interface (Android)**



**Figure C.2.1: Productivity Scenario (Windows)**

**Figure C.2.2: Productivity Scenario (iOS)**



**Figure C.2.3: Productivity Scenario (iOS)**

**Figure C.3.1: Creativity Scenario (Android)**



**Figure C.3.2: Creativity Scenario (Windows)**

**Figure C.4.1: Video Playback Scenario (iOS)**



**Figure C.4.2: Video Playback Scenario (Android)**

**Figure C.5.1: Detailed Results Screen (iOS)**