

**Test Plan**

**For**

**TIX ID**

*ChangeLog*

| Version | Change Date | By | Description |
|---------|-------------|----|-------------|
|         |             |    |             |
|         |             |    |             |
|         |             |    |             |

|  |           |
|--|-----------|
| <b>1 Introduction.....</b>                               | <b>2</b>  |
| 1.1 Scope.....   | 2         |
| 1.1.1 In Scope.....                                      | 2         |
| 1.1.2 Out of Scope.....                                  | 3         |
| 1.2 Quality Objective.....                               | 3         |
| 1.3 Roles and Responsibilities.....                      | 4         |
| <b>2 Test Methodology.....</b>                           | <b>5</b>  |
| 2.1 Overview.....  | 5         |
| 2.2 Test Levels.....                                     | 5         |
| 2.2.1 Unit Testing.....                                  | 5         |
| 2.2.2 Integration Testing.....                           | 6         |
| 2.2.3 System Testing.....                                | 6         |
| 2.2.4 User Acceptance Testing (UAT).....                 | 6         |
| 2.2.5 Regression Testing.....                            | 6         |
| 2.2.6 Performance Testing.....                           | 6         |
| 2.2.7 Security Testing.....                              | 7         |
| 2.3 Bug Triage.....                                      | 7         |
| 2.3.1 Severity (Tingkat Keparahan).....                  | 7         |
| 2.3.2 Priority (Prioritas Perbaikan).....                | 7         |
| 2.3.3 Keputusan Bug Triage.....                          | 8         |
| 2.4 Suspension Criteria and Resumption Requirements..... | 8         |
| 2.5 Test Completeness.....                               | 9         |
| <b>3 Test Deliverables.....</b>                          | <b>10</b> |
| <b>4 Resource &amp; Environment Needs.....</b>           | <b>11</b> |
| 4.1 Testing Tools.....                                   | 11        |
| 4.2 Test Environment.....                                | 12        |
| <b>5 Terms/Acronyms.....</b>                             | <b>14</b> |

# 1 Introduction

Strategi pengujian aplikasi TIX.ID bertujuan untuk memastikan bahwa seluruh fungsionalitas, performa, keamanan, dan kemudahan penggunaan aplikasi berjalan sesuai dengan spesifikasi yang ditentukan. Proses pengujian mencakup tahap perencanaan, desain, pelaksanaan, pelaporan, dan peninjauan hasil pengujian.

Metodologi pengujian yang digunakan adalah Agile Testing yang beriringan dengan siklus pengembangan perangkat lunak secara iteratif. Pengujian dilakukan berbasis risiko dan prioritas fitur yang paling krusial bagi pengguna. Jenis pengujian yang digunakan meliputi:

- Functional Testing
- Regression Testing
- Security Testing
- Usability Testing
- Performance Testing

Pengujian dilakukan pada aplikasi mobile untuk platform Android dan iOS, serta pada sistem backend (API & integrasi sistem pembayaran).

## 1.1 Scope

---

### 1.1.1 In Scope

Bagian ini mencakup aspek-aspek yang akan diuji untuk memastikan kualitas dan fungsionalitas aplikasi berjalan sesuai spesifikasi.

- Fungsionalitas Utama Aplikasi
  - Pemesanan tiket bioskop (Cinema XXI, CGV, Cinapolis)
  - Pembayaran melalui DANA dan metode pembayaran lain yang tersedia
  - Manajemen tiket digital (penyimpanan, pembatalan, dan penggunaan kode QR)
  - Pencarian dan pemilihan film berdasarkan lokasi dan preferensi pengguna
  - Notifikasi transaksi dan informasi terkait film serta promo
- Keamanan dan Perlindungan Data
  - Pengujian keamanan sistem login dan registrasi pengguna
  - Proteksi data pengguna termasuk informasi pembayaran

- Pengalaman Pengguna (Usability Testing)
  - Navigasi aplikasi yang intuitif
  - Responsivitas aplikasi di berbagai ukuran layar dan perangkat
- Performa Aplikasi (Performance Testing)
  - Uji kecepatan respons aplikasi saat pemesanan tiket
  - Pengujian aplikasi pada berbagai versi Android dan iOS
  - Evaluasi penggunaan sumber daya (RAM, CPU, baterai)

### 1.1.2 Out of Scope

Bagian ini mencakup aspek-aspek yang tidak akan diuji dalam lingkup pengujian aplikasi.

- Infrastruktur Sistem Eksternal
  - Pengujian server pihak ketiga di luar kendali tim pengembang
  - Performa jaringan internet pengguna yang dapat mempengaruhi pengalaman aplikasi
- Fitur yang Belum Dirilis atau Dalam Tahap Pengembangan
  - Fitur baru yang masih dalam tahap konsep atau perancangan
  - Layanan streaming film jika belum tersedia dalam ekosistem aplikasi
- Integrasi dengan Pihak Ketiga di Luar Layanan Utama
  - Pembelian merchandise atau layanan lain yang bukan bagian dari TIX ID
  - Interaksi aplikasi dengan sosial media yang bukan bagian dari fitur utama

## 1.2 Quality Objective

---

Tujuan dari kegiatan pengujian ini adalah untuk memastikan:

- Aplikasi Tix.id sesuai dengan persyaratan fungsional dan non-fungsional yang telah ditentukan
- Aplikasi dapat berjalan dengan stabil dan responsif di berbagai perangkat dan platform (Android, iOS)
- Semua bug dan isu yang signifikan ditemukan dan diperbaiki sebelum aplikasi diluncurkan ke publik
- Memastikan pengalaman pengguna (user experience) tidak terganggu oleh error atau masalah performa
- Aplikasi dapat menangani transaksi dalam skala besar dengan tetap stabil

## 1.3 Roles and Responsibilities

---

Detail description of the Roles and responsibilities of different team members like

- QA Analyst

QA berperan sebagai penguji utama dalam tim pengembangan aplikasi Tix.id. QA memastikan bahwa seluruh fitur seperti pembelian tiket bioskop, pemilihan kursi, metode pembayaran, dan notifikasi transaksi berjalan dengan benar dan sesuai dengan kebutuhan pengguna. QA menyusun dan mengeksekusi test case, mengidentifikasi bug, serta melakukan verifikasi ulang setelah bug diperbaiki. QA juga menguji aplikasi di berbagai perangkat dan kondisi jaringan untuk menjamin pengalaman pengguna yang konsisten.

- Developers Teams

Developer berperan sebagai pembuat dan pemelihara fitur-fitur aplikasi Tix.id, seperti proses login/register, integrasi metode pembayaran seperti DANA atau OVO, dan sistem pemilihan kursi bioskop. Developer menyediakan build terbaru untuk diuji dan segera memperbaiki bug yang ditemukan oleh QA. Developer juga bekerja sama dengan QA untuk menjelaskan logika fungsionalitas tertentu yang kompleks.

- Product Owner

Product Owner berperan sebagai representatif bisnis yang bertugas mengomunikasikan kebutuhan dan ekspektasi pengguna aplikasi Tix.id kepada tim teknis. Ia menetapkan prioritas fitur yang akan diuji, seperti fitur presale tiket film blockbuster atau integrasi dengan bioskop baru, dan memberikan klarifikasi bila terjadi ambiguitas dalam requirement. PO juga meninjau hasil pengujian untuk memastikan fitur siap rilis ke pengguna.

## 2 Test Methodology

### 2.1 Overview

---

#### **Metodologi Agile dengan Pendekatan Manual dan Automation Testing dalam Proyek Pengujian Aplikasi TIX ID**

Metodologi Agile dipilih karena memberikan fleksibilitas dan efektivitas dalam memastikan kualitas aplikasi melalui kombinasi manual testing dan automation testing. Manual testing digunakan untuk pengujian fitur baru, fitur kompleks, atau pengujian yang membutuhkan eksplorasi dan penilaian UX secara langsung. Sementara automation testing digunakan untuk pengujian regresi, fungsi utama yang stabil, serta pengujian berulang yang dapat meningkatkan efisiensi waktu.

Adapun alasan pemilihan pendekatan ini adalah sebagai berikut:

- Mendukung pengujian dalam metodologi Agile yang bersifat iteratif dan cepat.
- Menangani variasi fitur yang dinamis dan sering berubah, seperti promo dan integrasi bioskop.
- Mempercepat proses regression testing melalui penggunaan skrip otomatis.
- Menjaga konsistensi dan stabilitas aplikasi pada saat rilis build baru.
- Menghemat waktu dan sumber daya dengan mengotomatiskan pengujian yang berulang.
- Memberikan ruang bagi pengujian eksplorasi dan validasi kualitas dari sudut pandang pengguna.

### 2.2 Test Levels

---

Berikut adalah Test Levels yang mendefinisikan berbagai jenis pengujian yang akan dilakukan pada Aplikasi TIX ID:

#### **2.2.1 Unit Testing**

Tujuan unit testing adalah untuk memastikan bahwa setiap komponen atau modul aplikasi berfungsi dengan benar secara individu. Pengujian ini bertujuan untuk mendeteksi dan memperbaiki bug pada tingkat paling awal sebelum modul diintegrasikan ke dalam sistem yang lebih besar. Dengan melakukan pengujian ini, pengembang dapat menjamin bahwa setiap fungsi berjalan sesuai spesifikasi sebelum digunakan lebih lanjut dalam pengujian yang lebih kompleks.

### **2.2.2 Integration Testing**

Tujuan dari integration testing adalah untuk menguji bagaimana berbagai modul dalam aplikasi berkomunikasi satu sama lain. Pengujian ini bertujuan untuk memastikan bahwa integrasi antara frontend dan backend berjalan lancar, termasuk validasi komunikasi antara API dan sistem pembayaran. Dengan demikian, setiap transaksi yang dilakukan oleh pengguna dapat diproses dengan akurat tanpa kesalahan dalam pertukaran data antar komponen.

### **2.2.3 System Testing**

Pengujian sistem bertujuan untuk menilai keseluruhan aplikasi dalam lingkungan yang menyerupai kondisi operasional sebenarnya. Tujuan utama adalah memastikan bahwa semua fitur berfungsi secara end-to-end sesuai dengan kebutuhan bisnis dan ekspektasi pengguna. Dengan pengujian ini, tim QA dapat memverifikasi kesiapan aplikasi sebelum peluncuran serta mengidentifikasi potensi masalah yang dapat mempengaruhi pengalaman pengguna.

### **2.2.4 User Acceptance Testing (UAT)**

UAT bertujuan untuk memastikan bahwa aplikasi memenuhi kebutuhan dan harapan pengguna akhir. Pengujian ini dilakukan dengan mengundang pengguna internal atau perwakilan pengguna untuk mencoba aplikasi dan memberikan umpan balik. Tujuan utamanya adalah memvalidasi bahwa semua fitur bekerja dengan baik dalam skenario penggunaan sebenarnya, sehingga aplikasi siap digunakan secara luas.

### **2.2.5 Regression Testing**

Tujuan regression testing adalah untuk memastikan bahwa perubahan atau pembaruan yang dilakukan pada aplikasi tidak menyebabkan gangguan pada fitur yang sudah ada sebelumnya. Setiap kali ada pembaruan, pengujian ini dilakukan untuk menghindari efek samping yang tidak diinginkan. Dengan demikian, stabilitas aplikasi tetap terjaga setelah dilakukan perbaikan bug atau penambahan fitur baru.

### **2.2.6 Performance Testing**

Pengujian ini bertujuan untuk mengevaluasi kecepatan, stabilitas, dan responsivitas aplikasi saat digunakan oleh banyak pengguna secara bersamaan. Dengan melakukan performance testing, tim

pengembang dapat memastikan bahwa aplikasi tetap optimal dalam berbagai kondisi, termasuk saat terjadi lonjakan trafik tinggi.

### **2.2.7 Security Testing**

Security testing bertujuan untuk mengidentifikasi dan mengatasi potensi celah keamanan dalam aplikasi. Pengujian ini dilakukan untuk memastikan perlindungan data pengguna, sistem otentikasi yang aman, dan integrasi pembayaran yang terlindungi dari ancaman siber. Dengan pengujian ini, risiko kebocoran informasi pribadi dan eksploitasi sistem dapat diminimalkan.

## **2.3 Bug Triage**

---

Bug Triage bertujuan untuk menentukan jenis resolusi untuk setiap bug yang ditemukan. Selain itu memprioritaskan bug berdasarkan dampak dan urgensinya terhadap pengguna dan menjadwalkan perbaikan untuk semua bug yang masuk kategori "To Be Fixed". Bug akan dikriteriakan sebagai berikut:

### **2.3.1 Severity (Tingkat Keparahan)**

- Critical – Bug yang menyebabkan aplikasi crash atau menghambat fungsi utama.
- High – Bug yang secara signifikan mengganggu pengalaman pengguna tetapi masih bisa digunakan dengan workaround.
- Medium – Bug yang berdampak pada beberapa fitur tanpa menghambat pengalaman utama.
- Low – Bug minor seperti tampilan UI yang kurang optimal atau ketidaksesuaian teks.

### **2.3.2 Priority (Prioritas Perbaikan)**

- P1 (Urgent)  
Harus diperbaiki segera karena berdampak langsung pada fungsionalitas utama aplikasi.
- P2 (High Priority)  
Perlu diperbaiki dalam update terdekat untuk meningkatkan stabilitas dan pengalaman pengguna.
- P3 (Normal Priority)  
Bisa diperbaiki dalam siklus pengembangan berikutnya tanpa dampak besar.
- P4 (Low Priority)



Dapat diperbaiki di masa depan tanpa urgensi tinggi.

### **2.3.3 Keputusan Bug Triage**

- Bug yang masuk kategori Critical & High akan diperbaiki dalam update mendatang.
- Bug yang berhubungan dengan backend & API akan dikoordinasikan dengan tim pengembang backend.
- Bug kategori Low Priority akan dimasukkan dalam backlog untuk evaluasi lebih lanjut.

## **2.4 Suspension Criteria and Resumption Requirements**

---

Pengujian aplikasi TIX ID dapat dihentikan sementara apabila ditemukan kondisi-kondisi yang menghambat kelancaran, validitas, atau efektivitas proses pengujian. Salah satu alasan utama untuk menghentikan pengujian adalah ditemukannya bug dengan tingkat keparahan tinggi, seperti bug blocker atau critical, yang menyebabkan fitur utama aplikasi tidak dapat digunakan. Contohnya termasuk kegagalan total dalam proses pemesanan tiket bioskop atau transaksi pembayaran yang tidak dapat diselesaikan, yang secara langsung berdampak pada pengalaman pengguna. Selain itu, jika sistem backend, termasuk API atau integrasi dengan layanan pembayaran digital seperti DANA, mengalami gangguan atau downtime lebih dari satu jam, maka proses pengujian harus dihentikan karena ketergantungan sistem yang tinggi terhadap layanan backend tersebut.

Pengujian juga akan dihentikan apabila build aplikasi yang diuji terbukti tidak stabil, misalnya sering mengalami crash atau error saat dijalankan pada mayoritas perangkat pengujian yang digunakan tim QA. Hal ini menunjukkan bahwa versi tersebut belum siap untuk diuji lebih lanjut. Di sisi lain, apabila terdapat perubahan signifikan dalam requirement atau spesifikasi fitur, maka pengujian harus dihentikan sementara untuk memberi waktu bagi tim QA memperbarui test case, dokumentasi pengujian, dan strategi validasi lainnya agar tetap sesuai dengan kondisi terbaru. Selain itu, apabila lingkungan atau alat bantu pengujian mengalami gangguan teknis, seperti server staging yang tidak bisa diakses, kerusakan pada database pengujian, atau tool otomatisasi pengujian yang tidak berfungsi, maka kegiatan pengujian harus dihentikan sampai semua sistem kembali dapat digunakan.

Setelah kondisi-kondisi tersebut diatasi, maka proses pengujian dapat dilanjutkan kembali sesuai dengan resumption criteria yang telah ditetapkan. Pengujian hanya boleh dilanjutkan apabila bug blocker atau critical yang menyebabkan penghentian telah diperbaiki dan berhasil divalidasi melalui smoke testing, untuk memastikan bahwa bug tersebut benar-benar sudah tidak ada. Sistem backend dan integrasi pembayaran juga harus dipastikan telah kembali berjalan normal dan stabil. Selain itu, build aplikasi yang baru harus terbukti lebih stabil dibanding versi sebelumnya, dan mampu dijalankan tanpa crash atau error mayor pada perangkat uji. Jika penghentian disebabkan oleh perubahan requirement, maka test case dan dokumen pengujian harus diperbarui terlebih dahulu agar sesuai dengan perubahan tersebut. Terakhir, alat dan lingkungan pengujian yang sebelumnya mengalami masalah harus dipastikan telah berfungsi kembali, termasuk server staging, API, alat automasi, serta akses terhadap layanan-layanan pendukung lainnya. Dengan terpenuhinya seluruh kriteria tersebut, proses pengujian dapat dilanjutkan kembali dengan akurasi dan efektivitas yang optimal.

## **2.5 Test Completeness**

---

Pengujian aplikasi TIX ID akan dianggap lengkap (test complete) apabila semua kriteria berikut telah terpenuhi:

1. 100% test case coverage terhadap seluruh requirement fungsional dan non-fungsional yang ada dalam ruang lingkup pengujian.
2. Seluruh test case manual dan otomatis telah dieksekusi minimal satu kali.
3. Seluruh bug dengan tingkat severity medium hingga critical telah diperbaiki dan berhasil diverifikasi ulang (re-tested).
4. Bug minor yang masih terbuka telah dicatat dan dijadwalkan untuk diperbaiki di versi berikutnya dengan dokumentasi lengkap.
5. Seluruh fitur utama aplikasi (pemesanan tiket, pembayaran, manajemen tiket, notifikasi, dan pencarian film) berjalan sesuai ekspektasi tanpa error mayor.
6. Laporan pengujian telah selesai disusun, termasuk metrics, ringkasan bug, dan rekomendasi kualitas.
7. Stakeholder telah memberikan sign-off bahwa versi build tersebut siap untuk rilis atau digunakan lebih lanjut.

### 3 Test Deliverables

Test deliverables adalah artefak yang dihasilkan selama siklus pengujian perangkat lunak. Dokumen-dokumen ini berguna dalam memastikan bahwa setiap tahap pengujian berjalan sesuai rencana dan memenuhi standar kualitas yang ditetapkan. Berikut adalah daftar test deliverables yang akan dihasilkan dalam pengujian aplikasi TIX ID:

- Test Plan

Test Plan adalah dokumen yang berisi perencanaan pengujian perangkat lunak secara terstruktur dan menyeluruh. Dokumen ini bertindak sebagai panduan bagi tim QA dalam melakukan pengujian terhadap aplikasi TIX ID, memastikan bahwa semua aspek fungsionalitas diuji dengan metode yang tepat.

- Test Cases

Test Cases adalah sekumpulan skenario pengujian yang dirancang untuk mengevaluasi apakah suatu fitur atau fungsi dalam aplikasi bekerja sesuai spesifikasi. Setiap test case berisi langkah-langkah yang harus diikuti oleh tester serta hasil yang diharapkan.

- Test Strategy

Test Strategy adalah dokumen yang mendefinisikan pendekatan dan metode pengujian yang akan diterapkan dalam proyek pengembangan perangkat lunak. Berbeda dengan Test Plan yang bersifat lebih teknis dan operasional, Test Strategy lebih berfokus pada prinsip dasar pengujian dan pendekatan yang akan diadopsi oleh tim QA.

# 4 Resource & Environment Needs

## 4.1 Testing Tools

---

Untuk mendukung proses pengujian aplikasi TIX ID secara menyeluruh, diperlukan berbagai macam tools yang berfungsi dalam pelacakan kebutuhan, pencatatan bug, pengujian otomatis, serta validasi performa sistem. Penggunaan tools ini sangat penting untuk memastikan bahwa seluruh fungsionalitas aplikasi, baik pada sisi mobile (Android & iOS) maupun backend (API & sistem pembayaran), berjalan sesuai standar kualitas yang ditetapkan.

### 1. Requirements Tracking Tool

Untuk mencatat dan memantau kebutuhan fungsional dan non-fungsional aplikasi TIX ID, digunakan tools manajemen proyek seperti:

- Jira: Jira digunakan oleh tim pengembang dan QA untuk mencatat user stories, requirement fitur, dan skenario pengujian. Setiap fitur penting seperti pemesanan tiket, pemilihan kursi, dan integrasi pembayaran didokumentasikan secara detail di sini, termasuk acceptance criteria dan prioritas pengujian.
- Confluence: Digunakan sebagai ruang dokumentasi internal untuk mendeskripsikan spesifikasi teknis, flow aplikasi, dan referensi untuk test case. Tools ini mempermudah komunikasi antara QA, developer, dan product owner.

### 2. Bug Tracking Tool

Untuk melacak dan mengelola isu yang muncul selama proses pengujian:

- Jira (Issue Tracker): Selain untuk tracking requirement, Jira juga digunakan sebagai bug tracking tool. Setiap bug yang ditemukan, baik minor maupun critical, dicatat di sini lengkap dengan severity, steps to reproduce, dan screenshot/log pendukung. Tim QA dan developer akan berkoordinasi melalui tiket Jira untuk mempercepat proses debugging dan verifikasi.
- Bugzilla (opsional): Dapat digunakan sebagai alternatif jika diperlukan sistem pelaporan bug yang lebih terfokus dan fleksibel, terutama untuk tracking masalah lintas tim atau modul besar.

### 3. Automation Testing Tools

Automasi pengujian sangat penting dalam memastikan stabilitas aplikasi TIX ID, terutama karena aplikasi terus diperbarui secara berkala dengan berbagai promosi, film baru, dan integrasi mitra bioskop. Tools yang digunakan meliputi:

- Appium: Digunakan untuk mengotomatisasi pengujian UI pada aplikasi mobile TIX ID di platform Android dan iOS. Appium membantu menguji alur seperti pemesanan tiket, login, pembayaran, dan notifikasi secara efisien.
- Postman dan Newman: Untuk pengujian API backend seperti pemrosesan transaksi, validasi QR code tiket, dan sinkronisasi data user. Postman digunakan untuk eksplorasi dan pengujian manual, sementara Newman digunakan untuk integrasi ke pipeline otomatis.
- JMeter: Digunakan untuk performance testing, khususnya untuk menguji beban transaksi besar saat high traffic (misalnya saat promo film atau midnight sale). Ini membantu mengidentifikasi bottleneck pada server backend dan sistem pembayaran.

### 4. Continuous Integration and Build Tools

Untuk mempercepat proses pengujian dan memastikan regresi dicegah sejak awal pengembangan:

- Jenkins: Digunakan untuk mengotomatisasi build dan pengujian. Setelah developer mengunggah kode terbaru, Jenkins akan secara otomatis menjalankan pipeline pengujian seperti smoke test, unit test, dan regresi test menggunakan Appium dan Newman. Ini membantu memastikan bahwa tidak ada fitur utama yang rusak setelah update.

## 4.2 Test Environment

---

Bagian ini menjelaskan lingkungan minimum yang dibutuhkan untuk melakukan pengujian aplikasi, baik dari sisi perangkat keras maupun perangkat lunak. Lingkungan ini harus memenuhi spesifikasi untuk memastikan pengujian berjalan stabil, akurat, dan merepresentasikan kondisi riil pengguna.

### Minimum Hardware Requirements

- Processor: Intel Core i3 (generasi ke-4) atau setara, lebih tinggi disarankan
- RAM: Minimal 4 GB (8 GB direkomendasikan untuk pengujian paralel atau automation)

- Storage: Minimum 100 GB ruang kosong
- Display: Resolusi minimal 1366 x 768
- Koneksi Internet: Stabil dengan bandwidth minimal 5 Mbps untuk mengakses API, layanan backend, dan fitur berbasis cloud
- Software Requirements

Selain perangkat lunak spesifik dari klien, berikut **perangkat lunak umum** yang dibutuhkan dalam lingkungan pengujian:

- Sistem Operasi: Windows 8 dan versi di atasnya (Windows 10/11 direkomendasikan)
- Microsoft Office: Versi 2013 atau di atasnya (digunakan untuk dokumentasi, pelaporan, dan data testing)
- Email Client/Server: Microsoft Exchange (untuk pengujian pengiriman email notifikasi dan komunikasi sistem)

Browser:

- Google Chrome (versi terbaru)
- Mozilla Firefox
- Microsoft Edge

Tools Pengujian:

- Jira atau sistem pelaporan bug lainnya
- Postman (untuk pengujian API)
- Selenium/Appium (untuk automation testing)
- Emulator Android/iOS (jika diperlukan untuk pengujian lintas perangkat)

## 5 Terms/Acronyms

Istilah atau akronim yang digunakan dalam proyek

| TERM/ACRONYM | DEFINITION                    |
|--------------|-------------------------------|
| API          | Application Program Interface |
| AUT          | Application Under Test        |
| iOS          | IPhone Operating System       |
| Mbps         | Megabit per Second            |
| GB           | Gigabyte                      |
| RAM          | Random Access Memory          |
| CPU          | Central Processing Unit       |
| QA           | Quality Assurance             |
| PO           | Project Owner                 |