Test Plan for “Huntd”(web/mobile)

▼ 1. Introduction

▼ 2. Test Objectives

2.1 Aspects and Objectives

2.2 Desired Outcomes

▼ 3. Test Strategy

3.1 Testing Scope

3.2 Out of test scope

▼ 4. Test Approach

4.1 Team Roles and Responsibilities

4.2 Testing Types

4.3 Tools to be used

4.4 Testing phases and activities

▼ 5. Test Deliverables

5.1 Before the testing phase

5.2 During the testing phase

5.3 After the testing phase

▼ 6. Test Environment

6.1 Hardware

6.2 Software

▼ 7. Test Schedule

7.1 All project tasks and estimation

7.2 Schedule to complete these tasks

▼ 8. Test Risk and Mitigation Factors

▼ 9. Test Execution

9.1 Test Data Preparation

9.2 Test Cases Execution

▼ 10. Test Criteria

10.1 Suspension Criteria

10.2 Exit Criteria

▼ 11. Test Reporting

11.1 Documentation and Reporting of Test Results

11.2 Format and Structure of Test Report

▼ 12. Defect Management

12.1 Process for Reporting, Tracking and Managing Defects

12.2 Priority Levels for Defects

12.3 Roles and Responsibilities in Defect Management

▼ 13. Test Closure

13.1 Criteria and Activities for Closing the Testing

▼ 14. Approval

1. Introduction

Huntd is a comprehensive job search and application tracking app designed to assist job seekers in their quest for employment. With its user-friendly interface and powerful features, this app is essential for individuals actively seeking job opportunities.

Job seekers of all experience levels and industries can benefit from Huntd. Whether you're a recent graduate exploring entry-level positions or a seasoned professional looking for new career opportunities, this app provides the tools and resources necessary to streamline your job search process.

Huntd is particularly valuable for users who want to stay organised and efficient throughout their job search journey. With features like personalised job recommendations, application tracking, and a resume builder, this app ensures that users never miss an opportunity and can effortlessly keep track of their progress.

2. Test Objectives

The objective of testing is to verify the functionality, usability, and stability of the *Huntd*

job search and application tracking platform across both web and mobile (Android) interfaces.

Testing will focus on ensuring a seamless and reliable experience for both candidates and recruiters

across core user flows.

2.1 Aspects and Objectives

Functional Testing

Goal	Testing Area
Verify that features work as expected	Sign in Sign up Chats Profile Candidates list

UX/UI Testing

Goal	Testing Area
Ensure the interface is user-friendly and consistent	Layout consistency across pages/devices Button placement, font sizes, mobile responsiveness

Cross-Browser Testing

Goal	Testing Area
Ensure consistent behavior and appearance across different desktop browsers	Chrome, Firefox, Edge. Check layout rendering, buttons, forms, navigation. Focus on Main Page, Sign In/Up, Candidates List, Chats, Profile

Compatibility Testing

Goal	Testing Area
Ensure the app functions correctly on different devices, OS, and screen sizes	Desktop (Windows), Mobile (Android) Portrait/Landscape orientation Responsive behavior of Candidates list, Job tables, forms, banners

Mobile Testing

Name	Goal	Testing Area	Tools
Resolution Testing	Ensure proper UI behavior on different screen resolutions	Test the app on Android devices with various screen sizes and resolutions (e.g., 720p, 1080p, 1440p) Check that UI elements are visible, not cut off, properly scaled, and usable	Real Android devices, Android Studio Emulator, Chrome DevTools
CPU Testing	Verify app stability and responsiveness before, during, and after CPU	Monitor how the app behaves when CPU is under load. Compare performance before, during, and after CPU usage.	CPU Monitor
Connection Testing	Verify app behavior under network interruptions and changes	Test switching between Wi-Fi and mobile data, loss of connection, airplane mode; check how forms, login, and chats behave	Real Android devices, Android Studio Emulator, Chrome DevTools
Interruption	Verify app stability during interruptions	Simulate incoming calls, SMS, push notifications, screen lock/unlock, and app minimization; ensure the app resumes properly without data loss or crashes	Real Android devices, Android Studio Emulator, Chrome DevTools

2.2 Desired Outcomes

- All core features (registration, login, candidate listing, profile switching, chat) function correctly according to product requirements.
- System is stable and responsive on both web and Android platforms, including during interruptions (calls, app minimisation) and network changes (Wi-Fi ↔ mobile data, offline mode).

- Mobile app remains usable under CPU load, without crashes or critical performance issues on low/mid-end Android devices.
- UI and layout are consistent across screen sizes and resolutions (including small to large Android devices), with no broken or overlapping elements.
- Authentication and access control work as expected for both authorized and unauthorized users (e.g., filters, chats, posting).
- A reliable set of test cases is created and executed, enabling defect identification, retesting, and reuse for regression or UAT cycles.

3. Test Strategy

3.1 Testing Scope

Module Name	Login Type	Description
Main Page – For Companies	Logged out	Landing for recruiters. Shows product info, partner logos, comparisons. Encourages sign-up and leads to candidate list.
Main Page – For Engineers	Logged out / Logged in	Default main page. Logged out: registration form + links to Jobs/Web3. Logged in: banner promoting mobile app replaces form.
Sign Up	Logged out	Users can register as either recruiter or candidate via email or social accounts (Google, GitHub, LinkedIn).
Sign In	Logged out	Sign in with valid credentials; links to Sign Up and Forgot Password available.
Candidates List	Logged out / Logged in	Visible to all. Filters only usable when logged in. Recruiters can view cards; unauthorized users prompted to sign in.
Candidate Profile	Logged in	Opened in new tab. Shows full candidate info, except contacts (hidden until chat is initiated). Experience sorted oldest to newest.
Filters (on list)	Logged in only	Filter candidates by role, tech stack, salary, English, location. Hovering over filters shows sign-in prompt for guests.
Chat	Logged in	Recruiters can initiate chat. Candidate can accept or reject. Contacts only revealed after chat starts.
New Job / Vacancy Form	Logged in (recruiter)	Authorized recruiters can post jobs manually or via ATS import.
Job Listings	Logged out / Logged in	Everyone can see limited jobs. Full list and 1-click apply only for logged-in users.
Web3 Companies	Logged out / Logged in	List of 100 companies (paginated 10 per page). Clicking logo leads to company's vacancies in new tab.

Module Name	Login Type	Description
Subscribe to Vacancies	Logged out / Logged in	Form at the bottom of jobs page to subscribe by role, experience, and email.
Profile Page	Logged in	Manage recruiter and candidate profiles. Edit data, switch roles, change password, connect social accounts.
Settings	Logged in	Activate/deactivate candidate profile. Admins can edit profiles anytime.
Footer	All users	Contains Web3 top companies, social media links, docs, FAQ, and pricing.
Mobile App	Logged in	Android MVP. Supports selected features only: IAM, some chat, and profile settings. Test only implemented parts.

3.2 Out of Scope Testing

- Admin functionality;
- Question/feedback form.

4. Test Approach

4.1 Team Roles and Responsibilities

Team Member	Role	Responsibility
A.Voroshylova	QA Engineer	WEB Decomposition
A. Kharyakova	QA Engineer	Mobile Decomposition
A.Stolyarenko	QA Engineer	Integration Jira and TestRail Creating Test Plan
A.Brukh	QA Engineer	WEB Decomposition
R.Rasenko	QA Engineer	RTM Permission Table

4.2 Testing Types

In this testing cycle, the **Huntd platform** (web and Android mobile app) will be tested using the following test types:

System Testing

All tests will be conducted on a fully integrated and deployed version of the system (web and mobile). The goal is to verify that all modules work correctly together and meet the product requirements.

Exploratory Testing

Unscripted sessions will be used to discover unexpected bugs or usability issues across both platforms. This will help identify edge cases and behaviours not covered by predefined test cases.

Smoke Testing

Each new build (web and mobile) will undergo smoke testing to ensure that the most critical features (e.g., registration, login, candidate list, chat) work properly before deeper functional testing begins.

Functional Testing

All functional modules will be tested based on requirements, including:

- User registration and login
- Candidate listing and filtering
- Profile management
- Chat functionality
- Job listings and application flow
- Mobile features like chat

GUI Testing

User Interface testing will be performed on:

- Web version: validation of layout, element visibility, responsiveness, and proper behaviour of forms
- Mobile version: validation of UI on different Android screen sizes and orientations, ensuring buttons, text, and interactions behave as expected

Compatibility Testing

- **Web version:** The website will be tested on modern browsers on **Windows** and **macOS**, including:
  - Chrome (latest)
  - Safari (macOS)
  - Microsoft Edge (Windows)
- **Mobile version:** The Android app will be tested on:
  - Devices with different screen resolutions and OS versions
  - Android Studio Emulator and physical devices (if available)

Mobile Testing

Focused testing of mobile-specific conditions:

- **CPU Load Testing:** App behaviour before, during, and after high CPU usage
- **Screen Resolution Testing:** Layout consistency across various screen sizes (720p, 1080p, 1440p)
- **Network & Connection Testing:** Behaviour on connection loss, aeroplane mode, Wi-Fi/mobile switch
- **Interruption Testing:** Stability on incoming calls, notifications, and app minimisation

4.3 Tools to be Used

During this testing cycle, the following tools and platforms will be used to support the QA process:

TestRail

Used to **create and maintain test cases**, **execute test runs**, and **generate test reports**.

Test cases are organised by feature and platform (web/mobile). Results are tracked and used for reporting QA status per sprint or release.

Jira

Used to manage QA-related tasks and bugs.

- **Epics and user stories** describe features to be tested
- All **found defects are logged as bugs**, with proper severity, reproduction steps, and screenshots/videos
- Jira links bugs to test cases and helps coordinate regression testing

---

Notion

Used to create and maintain the **Test Plan.**

Android Studio

Used to test the Android mobile version of Huntd.

- **Screen Resolution Testing** — different screen sizes and pixel densities
- **Connection Testing** — simulate network changes, offline mode
- **Interruption Testing** — simulate calls, notifications, backgrounding
- **CPU Monitor** — used to monitor performance and debug issues

---

Chrome DevTools (for web UI testing)

Used to:

- Inspect layout, styling, and responsive behaviour
- Check console errors and network requests
- Test on different screen sizes

4.4 Testing phases and activities

1. Analysis Phase

- Analyse feature descriptions and requirements for recruiter and candidate flows
- Identify testable functionality and edge cases.

---

2. Planning and Preparation Phase

- Define the testing scope and out-of-scope.
- Develop a Test Plan in Notion based on feature requirements.
- Create test cases in TestRail for both web and Android.
- Prepare test data for typical and boundary-value scenarios.

---

3. Execution Phase

- Execute test cases manually using real data.
- Log all bugs in Jira
- For mobile (Android), run additional tests: resolution, CPU behaviour, interruptions, and connection handling.

---

4. Review Phase

- Analyse test execution results and validate coverage.
- Review and finalise defect reports, with prioritisation and links to related test cases.
- Update test documentation based on test findings.
- The project concludes for the QA team after this phase:

  All deliverables (test plan, test cases, bug reports) will be reviewed by mentors as part of the final evaluation.

5. Test Deliverables

5.1 Before the testing phase

Test Plan

Permission Table

Requirement Traceability Matrix

5.2 During the testing phase

Test Cases

Test Runs

Bug Reports

5.3 After the testing phase

Test Report

Mentor Review

6. Test Environment

This section describes the hardware and software configurations used during the testing cycle of the Huntd web and mobile application.

6.1 Hardware

Workstation: Acer Aspire A315-24P

- **Processor**: AMD Ryzen 3 7320U with Radeon Graphics
- **Memory**: 8 GB LPDDR5 RAM
- **Storage**: 256 GB NVMe SSD
- **Display**: 15.6-inch Full HD (1920 × 1080), LED
- **OS**: Windows 11 Home
- **Usage**: Web application testing, TestRail/Jira access, test documentation, Chrome DevTools, Android Studio (emulator)

Mobile Device: Xiaomi Redmi 10

- **Processor**: MediaTek Helio G88
- **Memory**: 4 GB RAM
- **Storage**: 64 GB internal
- **Display**: 6.5-inch FHD+ DotDisplay (2400 × 1080 px)

- **OS**: MIUI 14, based on Android 13
- **Usage**: Mobile testing of the Android app — resolution, CPU load, interruption, and connection testing.

6.2 Software

Platforms

- **Operating System**: Windows 10
- **Android Version (Real device)**: Android 13
- **BrowserStack**: For cross-browser and cross-platform compatibility testing (e.g., macOS, Safari)

🌐 Browsers

- Google Chrome (latest stable version)
- Mozilla Firefox (latest stable version)
- Microsoft Edge (latest)
- Optional: Safari (via BrowserStack only)

📱 Mobile Testing Tools

- **Android Studio** with Emulator
- **Logcat & Profiler** (performance and error monitoring)
- **Chrome DevTools** (for responsive and UI testing on web)

🌐 Network Configuration

- **Connection Type**: Wi-Fi
- **Speed**: Minimum 10 Mbps download / 5 Mbps upload

7. Test Schedule

7.1 All project tasks and estimation

Task	Member	Estimate effort
Analysing requirements	All participants	2 man-hours
Creating the test plan	A.Stolyarenko	6 man-hours
Mobile Decomposition	A.Kharyakova	6 man-hours
WEB Decomposition	A.Voroshilova A.Brukh	6 man-hours

Task	Member	Estimate effort
RTM(Requirement Traceability Matrix)	R.Rasenko A.Kharyakova	4 man-hours
Permission Table	R.Rasenko	3 man-hours
Creating Test Cases for WEB	All participants	10 man-hours
Creating Test Cases for mobile	A.Kharyakova	8 man-hours
Review Test Cases	All participants	2 man-hours
Test Cases Execution	All participants	12 man-hours
Create Bug Reports	All participants	5 man-hours
Writing Test Reports	R.Rasenko	3 man-hours

7.2 Schedule to complete these tasks

Task	14/07	15/07	16/07	18/07	19/07
Analysing requirements	✅				
Creating the test plan		✅			
Mobile Decomposition	✅				
WEB Decomposition	✅				
RTM(Requirement Traceability Matrix)	✅	✅	✅	✅	
Permission Table		✅			
Creating Test Cases for WEB		✅	✅		
Creating Test Cases for mobile		✅	✅		
Review Test Cases				✅	✅
Test Cases Execution			✅	✅	
Create Bug Reports			✅	✅	
Writing Test Reports					✅

8. Test Risks and Mitigation Factors

Risk	Impact	Mitigation Plan
Lack of Skilled Testers or Tools	High	- Share knowledge via mentoring. - Use free/open-source testing tools. - Ask for help from other teams if possible.
Limited Testing Time	High	- Prioritize tests for critical features and business flows. - Focus on smoke and testing of the critical path. - Collaborate closely with yeam members
Incomplete Test Coverage	High	- Regularly review test cases and add missing ones. - Analyze test reports to find gaps. - Involve the team in coverage reviews.
Not Enough Time to Test All Browsers/OS	Medium	- Identify and focus on browsers/OS most used by users. - Use risk-based testing. - Use browser emulators or cloud testing services.
A Team Member Becomes Sick or Unavailable	Medium	- Distribute tasks among the team. - Keep documentation updated for easy handover.

9. Test Execution

9.1 Test Data Preparation

Creating test data sets required for test scenarios:

- **User Data:** Various user profiles created with different roles and permissions, including candidates with diverse skills, experience, salary expectations, and recruiters with different hiring criteria.
- **Job Postings:** Sample job listings with varying roles, required technologies, salary ranges, and locations to validate job search and filter functionalities.
- **Chat Scenarios Data:** Test data to simulate recruiter-candidate interactions, including chat initiation, contact sharing permissions, and chat management actions.
- **Registration Data:** Data sets for user registration through email and social login providers for both candidate and recruiter flows.

9.2 Test Cases Execution

Preparation:

- Ensured all test cases were documented, reviewed, and approved by the team.
- Prioritised test cases based on risk, business criticality, and impact on core application features.

Execution:

- Executed test cases according to the planned test cycles and sprints.

- Recorded outcomes of each test case execution as pass, fail, or blocked.
- Tracked overall progress and coverage of test execution.

Defect Reporting:

- Reported defects found during test execution with detailed reproduction steps, screenshots or logs, defect severity, and expected vs. actual results.
- Assigned defects to appropriate development team members for resolution and tracked their status.

Retesting and Regression Testing:

- Retested functionalities after defect fixes to verify the effectiveness of corrections.
- Executed regression test cases to ensure that new changes did not negatively impact existing features.

10. Test Criteria

10.1 Suspension Criteria

- 10% of P0/P1 tests failed, which should lead to the testing suspension until the development team fixes all the corresponding bugs.
- 30% of P2/P3 tests failed, which should lead to the testing suspension until the development team fixes all the corresponding bugs.

10.2 Exit Criteria

The test execution will be stopped no later than the last day of the sprint. The exit criteria should be met to complete the testing cycle:

- The mandatory Execution rate is 95%.
- The mandatory Pass rate is 100% for P0/P1 tests.
- The mandatory Pass rate is 80% for P2/P3 tests.
- All necessary artefacts collected: test cases and bug reports.
- The product should not have known bugs with severity Critical and Major, and bugs with Priority High at the time of finishing testing.
- The list of allowed bugs is agreed upon with the developers and managers.

11. Test Reporting

11.1 Documentation and Reporting of Test Results

Test Results Documentation:
All test results will be documented in TestRail, indicating the status of each test case (Passed, Failed, Skipped, Blocked), with the following issues reporting into Jira.
Daily summaries of test activities, progress, and any blockers will be discussed during team daily meetings.

11. 2. Format and Structure of Test Report

1. Execution Rate for Huntd WEB app
2. Execution Rate for Huntd mobile app
3. Huntd WEB bugs Priority
4. The coverage of the WEB features by Test Cases
5. The coverage of the mobile features by Test Cases

12. Defect Management

12.1 Process for Reporting, Tracking, and Managing Defects

Defect Reporting:

1. **Identification:**

   Defects identified during testing (web or mobile) are analyzed, confirmed, and logged in Jira.

2. **Reporting Details:**

   Each defect report includes:

- **Summary:** Clear and concise title of the issue
- **Description:**
  - Preconditions (user is logged in)
  - Steps to reproduce
  - Expected vs. actual results
  - Priority level
  - Notes
- **Attachments:** Screenshots, videos (if needed), and browser or device logs.
- **Environment:**
  - Web: Chrome(latest)
  - Mobile: MI (Android 12)

Defect Tracking:

1. **Assignment:**

   Defects are assigned to the appropriate developer based on the module (e.g., registration, filters, chat, etc.).

2. **Status Updates:**

   The development team updates the status of each defect in Jira using standard workflow statuses (e.g., **To Do**, **In Progress**, **In Review, Testing, Done**).

   QA regularly reviews progress and verifies fixed issues.

Defect Management:

1. **Resolution and Verification:**

   After a fix is applied, QA rechecks the issue. If resolved, the defect is marked as **Done** or **Closed**.

   If not, the defect is **Reopened** with updated steps, screenshots, or logs for further investigation.

2. **Regression Testing:**

   Once a defect is resolved, related features are checked to ensure the fix did not introduce side effects.

   For example, after fixing chat-related logic, access to the candidate profile or filters is also verified as part of regression.

12.2 Priority Levels for Defects

Level Type	Description
Blocker	Prevents further progress; critical and halting functionality.
Highest	Requires immediate attention; significant impact on functionality and usability.
High	Important but not immediate; impacts significant functionality with workarounds.
Medium	Moderate impact; addressed in regular development cycle.
Low	Minor impact; resolved in future releases.
Lowest	Trivial impact; cosmetic issues or minor enhancements.

12.3 Roles and Responsibilities in Defect Management

QA Team Responsibilities:

- Maintain test documentation, including the Test Plan, Requirement Traceability Matrix (RTM), Test Cases, and Defect Reports.

- Prepare and execute test scenarios, focusing on both functional flows and permission-based testing (e.g., candidate vs. recruiter roles).
- Identify and report defects in Jira with clear reproduction steps, severity levels, and supporting evidence (screenshots, logs).
- Ensure traceability between defects and related requirements or test cases.
- Participate in retesting and regression testing once defects are resolved.
- Collaborate to track defect statuses and retest timelines.

Mentor Responsibilities:

- Review test documentation and bug reports to ensure quality and consistency.
- Provide feedback on test coverage, reporting style, and defect prioritization.
- Help the team resolve blockers related to test environments, unclear requirements, or complex defect investigation.
- Facilitate learning and growth by supporting testers in decision-making and encouraging best practices.

13. Test Closure

13.1 Criteria and Activities for Closing the Testing Phase

Test Closure Criteria:

- All planned and prioritized test cases have been executed.
- Test execution met the defined exit criteria (see section 10.2), including execution rate and required pass rates.
- All critical and major defects were resolved and retested.
- Remaining minor or low-priority bugs were documented and approved for future releases.
- All test artifacts (test cases, bug reports, traceability matrix, test plan) were completed and reviewed.
- Test coverage and quality goals were achieved as planned.

Test Closure Activities:

1. **Sprint Summary Report:**

   A summary report was created to reflect testing progress, key findings, passed/failed tests, and defect trends across the sprint.

2. **Defect Review and Status Update:**

   All defects were reviewed, updated in Jira, and either marked as resolved or deferred with clear justification.

3. **Metrics Collection:**

   Test metrics (execution rate, pass rate by priority level, defect density) were analyzed to evaluate overall testing effectiveness.

4. **Knowledge Transfer & Retrospective:**

   The QA team documented lessons learned, improvement points, and shared insights for future sprints or projects.

5. **Final QA Sign-off:**

   The QA team gave final sign-off confirming that testing was completed according to plan and no critical issues remain.

14. Approval

Objective:

To obtain formal approval of the test plan from the key participants of the QA team in the learning project, ensuring it has been reviewed, agreed upon, and validated before executing test activities.

The following team members participated in the review and informal approval of this test plan:

Role	Name	Approval	Date
QA Engineer	A. Kharyakova	Approved	15/07/25
QA Engineer	A. Voroshilova	Approved	15/07/25
QA Engineer	A. Stolyarenko	Approved	15/07/25
QA Engineer	A. Brukh	Approved	15/07/25
QA Engineer	R. Rasenko	Approved	15/07/25
Mentor			