Software Test Plan - STP

“paybox”

Maya Shamir
Version 4.9.4
30.1.24

Version Control

CURRENT VERSION

Title	Software Test Plan - STP
File	
Author	Maya shamir
Version	4.9.4
Version Date	23.1.24

Table of Contents

1 DOCUMENT OVERVIEW.....4

1.1 INTRODUCTION.....4
1.2 OBJECTIVES 5
1.3 SCOPE.....6
1.4 REFERENCES..... **שגיאה! הסימניה אינה מוגדרת.**

2 SCOPE OF TESTING.....7

2.1 FEATURES TO BE TESTED 7
2.2 FEATURES NOT TO BE TESTED.....7
2.3 TESTING TYPES.....8
2.4 TEST STRATEGY AND APPROACH.....9

3 PLANED SMOKE TEST FOR "GOOGLE SEARCH"
שגיאה! הסימניה אינה מוגדרת.

3.1 TEST OBJECTIVES ..... **שגיאה! הסימניה אינה מוגדרת.**
3.2 <MODULE NAME> ..... **שגיאה! הסימניה אינה מוגדרת.**
3.2.1 <SUB MODULE NAME>..... ***שגיאה! הסימניה אינה מוגדרת.***
3.3 <MODULE NAME> ..... **שגיאה! הסימניה אינה מוגדרת.**
3.3.1 <SUB MODULE NAME>..... ***שגיאה! הסימניה אינה מוגדרת.***

1 Document Overview

1.1Introduction

This document serves as the Software Test Plan for" Paybox" <4.9.4>.

The purpose of this STP is to define the framework and Strategy for the testing of "paybox application.

The plan is tailored to support the Agile Scrum methodology, emphasizing flexibility, and iterative development.

Our objective is to validate the High Quality of "paybox" mobile.

We will verify "paybox" mobile behaves as expected by testing its features and functionality.

In alignment with Scrum principles, this document will try to stay as short and focused on Testing needs so it can be easily updated and evolve throughout project iterations.

1.2 Objectives

At a high level The primary objectives of this Software Test Plan for "paybox" are as follows:

- ✓ **Ensure Product Quality:**

To uphold the high standards of quality for which paybox is known, verifying that all features work as intended and meet user and business requirements.

- ✓ **Enable Efficient Development Cycles:**

To align testing activities with Scrum sprints, facilitating swift identification and resolution of defects, and supporting the development team in quick iterations.

- ✓ **Support Business Goals:**

To ensure that the testing process aligns with the overarching business objectives, contributing to the sustained success and growth of paybox

1.3Scope

- The scope of this document is only for version <4.9.4> of "paybox" product.

- This STP won't include the Test Planning and Test Execution of "paybox " on the following OS: Linux, MacOS

2 Scope of testing

2.1 Features to be tested

Here you'll state all the Modules Features you plan to test. <Note that because its evolving document that some features/Modules could be added / deleted while the project is on process depends on timetables and complexity>

- ✓ payment
- ✓ recieving money
- ✓ Withdrawal from balance
- ✓ Payment Request
- ✓ Opening a group
- ✓ Group payment
- ✓ Group members - paid, did not pay
- ✓ Group members - member details
- ✓ Group members - invite a friend
- ✓ Group members invite a member by link
- ✓ Group members You have added a member from your contacts
- ✓ Withdrawing money from the group
- ✓ Transferring money to a member of the group
- ✓ Distribution of group money to all group members
- ✓ Personal area - my actions
- ✓ Personal area - editing profile
- ✓ Methods of Payment
  Disconnecting from the system

2.2 Features not to be tested

- Sign up
- log in
- log out

2.3Testing Types

Outlined below are the test types that will be planned and performed during this project:

Functionality Verification:

To ensure all features "paybox", payment, receiving money, Payment Request
operate as intended across various platforms and devices.

Usability Assessment:

To evaluate the user interface for intuitiveness, ease of use, and accessibility.
This includes payment page is easily navigable and that the interface elements are responsive to user interactions.

Compatibility Testing:

To confirm that paybox works seamlessly across differe operating systems (, Android, iOS), and devices (desktops, tablets, smartphones).

User Interface Testing

Check if the application works with WhatsApp, Instagram, Tiktok, Twitter, Gimel and more>

Boundary Testing
In the test we will check the edge between 1-4,000

Security testing
Ensure that the application is safe to use.
The application maintains the user's bank account

and closes when not in use

recovery testing
Ensure that the software knows how to recover in extreme situations such as lack of reception, sudden shutdown of the application

2.4 Test Strategy and Approach

Our test approach is systematic and structured to ensure thorough and efficient validation of each build received from the Development team.

The following outlines our planned testing progression for each release cycle:

Initial Build Assessment with Smoke Testing:

Upon receipt of a new build, the Quality Assurance (QA) team will execute a Smoke Testing Suite.
This suite is designed to quickly check the stability of the build and ensure that the core functionalities of Google Search are operating as expected.

Only after a build passes the smoke test will it move forward in the testing process.

Focused Testing on New Features and Bug Fixes with Sanity Testing:

After the build has passed the Smoke Testing phase, the QA team will proceed to Sanity Testing.
This phase is targeted at the new features and bug fixes included in the release.

The objective is to ensure that specific updates are functioning correctly in the application without any immediate issues.

Comprehensive Regression Testing:

Following the Sanity Testing phase, comprehensive Regression Testing will be conducted.
This is critical to ensure that new code changes have not adversely affected existing functionalities of Google Search.

The Regression Testing will be extensive and is designed to cover all areas of the application that could potentially be impacted by the changes.

Incorporation of Exploratory Testing:

Parallel to the structured testing phases, we allocate approximately 20% of the total testing effort during the execution phase for Exploratory Testing.

This approach allows testers to go beyond predefined test cases and scenarios, using their insights and experience to uncover issues that may not have been anticipated in the test planning stages.

Iterative Feedback and Continuous Integration:

The testing strategy is aligned with the Agile Scrum framework, which advocates for continuous integration and iterative feedback.

Testing phases will be tightly integrated with the sprint cycles, ensuring prompt feedback to the Development team and allowing for quick iteration and refinement of the application.

The proposed testing approach ensures a balance between structured testing and the flexibility to discover unforeseen issues, making it highly effective in an Agile development environment.

By following this approach, the QA team contributes to the delivery of a stable, high-quality product that meets the rigorous standards expected of Google Search.