Test Plan for

Routine+

Introduction

This test plan describes the testing approach and overall framework that will drive the testing of the Routine+ web application.

Purpose

This document provides the following guidance:

- Test Strategy: rules the test will be based on, including the givens of the project (e.g.: start / end dates, objectives, assumptions); description of the process to set up a valid test (e.g.: entry / exit criteria, creation of test cases, specific tasks to perform, scheduling, data strategy).
- Execution Strategy: describes how the test will be performed and process to identify and report defects, and to fix and implement fixes.
- Test Management: process to handle the logistics of the test and all the events that come up during execution (e.g.: communications, escalation procedures, risk and mitigation, team roster)

Project Overview

Routine+ is a powerful web application with the ability to provide routine management for users. Each user can log in his/her own account and manipulate the routine tasks.
The functionality includes sign up, log in, log out, create routine, edit routine and delete routine.
People can get access to their routines anywhere in the world through internet and easily maintain them.

Audience

- Project team members perform tasks specified in this document, and provide input and recommendations on this document.
- The stakeholders may take part in test to ensure the application is aligned with the business requirement.

Testing objectives

We will test the backend functionalities, frontend functionalities.

Features to be tested

Create new user, log in, post new routine, get routine, update routine, and delete routine.

Testing approach

Static Testing

Static testing is testing of a component or specifications without execution of that software. We will use Insomnia Tool to test requests in backend. That will allow us to do tests without the frontend.

Component Testing

Component level testing focuses on the functionality of each component being developed. All major frontend functions will go through component testing. Each part will be tested separately to guarantee the result. We will manually test the component with console information.

System Testing

The purpose of the system testing is to validate that the complete and integrated system complies with functional requirements and business requirements. We will take system testing once the functional part of the application done and the whole application built. People will use the application to carry out the system testing.

Defect Management

When defect detected, please announce that in the work group and inform the developer who in charge of that part. After fixed, the developer should upload new version of application and inform others.

Test Arrangement

Test objective	Test method	Test Frequency	Test member	Test deliverable
routine manipulation	static & component	Every time there's a change	Yuan Yao	test report
authentication	static & component	Every time there's a change	Junfeng Zhou	test report
advanced functions	static & component	Every time there's a change	Vy Thai & Weinien Chen	test report
application functionality	component & system	functional part finishes	team members	test report
application	system	the end of project	team members & stakeholders	feed backs