CS691 – Computer Science, Spring 2023

Pace University

SYSTEM TEST PLAN



ParkEz

Author: Tom Cookson

Project Manager: Krunal Bhavsar

Date: 5/2/23

**Table of Contents**

[INTRODUCTION 3](#_Toc133416652)

[TESTING SCOPE 3](#_Toc133416653)

[TESTING OBJECTIVES 3](#_Toc133416654)

[Features to be Tested 4](#_Toc133416655)

[Features not to be Tested 5](#_Toc133416656)

[TEST PROCESS DEFINITION 5](#_Toc133416657)

[Test Process Phases and Tasks 5](#_Toc133416658)

[Deliverables 6](#_Toc133416659)

[APPROACH TO SYSTEM TESTING 6](#_Toc133416660)

[Approach to Functional Testing 6](#_Toc133416661)

[ENTRY/EXIT CRITERIA 6](#_Toc133416662)

[Entry Criteria 7](#_Toc133416663)

[Exit Criteria 7](#_Toc133416664)

[ENVIRONMENTAL NEEDS 7](#_Toc133416665)

[ROLES AND RESPONSIBILITIES 8](#_Toc133416666)

[TEST CYCLES AND SCHEDULE 8](#_Toc133416667)

[RISKS AND CONTINGENCIES 9](#_Toc133416668)

# INTRODUCTION

This document outlines the System Test Plan for ParkEZ, establishing a common understanding among stakeholders about the scope, objectives, and approach to performing system testing. The plan addresses key topics such as features to be tested, environmental requirements, testing entry/exit criteria, test schedule, roles and responsibilities, and risks and contingencies.

# TESTING SCOPE

The testing scope for ParkEZ includes two perspectives - the functional scope and technical scope.

At the current state of development, the functional scope includes the following modules of the ParkEZ system:

* User Experience
* User Registration/Login
* Customer Service Tickets.

The technical scope includes the following architectural components:

* Web browser
* Application server
* Database server

At later stages of development Camera Feed Integration and Parking Detection will also be integrated into this document.

# TESTING OBJECTIVES

The primary objective of this System Test Plan is to focus on functional testing to assess the stability of the system implementation. Non-functional testing, such as performance monitoring, will require specialized tools that are not available for this project.

The foundation for creating functional tests and evaluating the system functionality is based on the following resources:

* Business Requirements Document (BRD)
* Functional Requirements and User Stories
* Requirements Composition Table (supplementary requirements)
* Database Design and Data Dictionary Specifications

## Features to be Tested

This section lists all core features that will be tested grouped by the application modules below. The number after each feature corresponds to the original Functional Requirement specification in the RCT and in the User Stories.

Account Management

* Account Summary (1.1)
  + All types of users can view a summary of their account, and it shows the correct information for that account.
* Account Login (1.2)
  + Both types of customers and all employee roles can log on.
  + Authentication fails if password is incorrect and validates if password is correct.
  + Customers / employees log on to the correct account.
  + Session management provides access to an account correctly until user logs off.
* Customer Care (1.3)
  + Account should be able request customer care. A test should show that customers can place a request and employees with appropriate roles should see that request on their account.
* Account Creation (1.4)
  + Accounts can be created through registration on the website for customers.
  + Accounts can be created through other administrator accounts for employees.
* Account Modification (1.5)
  + Accounts should be able to change personal information related to account include email and password.
* Account Cancelation (1.6)
  + To test whether a user can successfully cancel or delete an account. Once an account is canceled, it should be confirmed they can no longer log in.

Besides the core features in the scope of testing, the function testing also will cover crosscutting concerns that are applicable to the context of the individual core features (refer to the RCT).

## Features not to be Tested

As mentioned above, system performance will not be tested for the lack of required tools. Also, usability and security will not be tested as well.  
  
Also, major features not yet implemented cannot be test now, but will be implemented in the future and amended to this document. These features include:

# Parking Lot Management

# Tracking of how many spots taken at parking lots

# Access archived footage of parking lots

# Detect overparking

# Track license plate

# Find best available spot in lot

# ParkEz lot analysis offered to other websites via API

# Advertising Management

# Create, modify, delete ads.

# View ad statistics.

# Payment

# Payments should be able to be able to be entered, validated and processed through site, possible with help of third party service.

# TEST PROCESS DEFINITION

## Test Process Phases and Tasks

The test process comprises five phases: test planning, design, preparation, execution, and reporting. These phases encompass the following tasks:

* Test Planning
  + Define the scope and objectives of testing
  + Establish roles and responsibilities
  + Outline the testing approach
* Test Design
  + Generate test ideas and define a strategy for designing test cases
  + Develop test case specifications
  + Measure test coverage
  + Identify requirements for test data
* Test Preparation
  + Configure the test environment
  + Provide test data
  + Install ParkEz software in the test environment
* Test Execution
  + Execute all test cases
  + Identify and report software defects
  + Assess the system stability
  + Validate all target features
* Test Reporting
  + Summarize and communicate test execution results
  + Present defect metrics
  + Evaluate the test exit criteria
  + Generate a test completion report and submit for stakeholder approval
* Obtain stakeholder Obtain stakeholder signoff on system testing

## Deliverables

The ParkEz test process deliverables include:

* System Test Plan document
* Test Design specifications
* Test Case specifications
* Software Defects in
* Test Execution Logs for
* Test Completion Report

# APPROACH TO SYSTEM TESTING

## Approach to Functional Testing

Functional testing for ParkEz will employ the Black-box method:

* Test cases will be designed using black-box techniques like boundary-value analysis, equivalent-class partitioning, cause-effect graphing, decision tables, and state-transition testing when applicable.
* Test execution will be performed manually, from the user perspective, and based on formal test case specifications.

Test execution results will be documented and reported in test execution logs.

# ENTRY/EXIT CRITERIA

This section defines the Entry and Exit Criteria for ParkEz test execution, ensuring a shared understanding of the conditions that determine when the test execution can commence and when it can conclude.

## Entry Criteria

* The test Entry Criteria include the following items:
* ParkEz application build is produced and deployed to the test environment
* ParkEz System Test Plan is produced and approved
* Test environment is ready for testing
* Test Designs and test case specifications are completed

## Exit Criteria

The test Exit Criteria include the following items:

* All requirements in the scope of testing are covered by test cases
* All test cases have been executed
* Zero defects of Critical and Hi-severity remain open
* Open defects of Medium and Low severity have known workarounds
* Test Summary report is produced and published

# ENVIRONMENTAL NEEDS

The Test Environment should be available to start test execution. It includes a laptop with container running the web server and database, and internet browsers (Chrome, Firefox, Edge and Safari) to access the application. The architecture of the test environment is shown below.

Authentication & Authorizing  
Members

Encrypting  
User   
Passwords

Monitoring   
Activity

User

Validations  
Input

Storing/Retrieving/  
Manipulation Data

Error Handling

User Authentication

Database Servers

Application Servers

Web Servers

Application

Web Servers

Database

Browser

# ROLES AND RESPONSIBILITIES

The project team has several members assigned to various project roles, including Project Manager, Product Owner, Lead Business Analyst, Lead Developer, DBA, and Lead QA Analyst. Their responsibilities are outlined in the table below:

|  |  |
| --- | --- |
| **Project Role** | **Role Responsibilities** |
| Project Manager | Reviewing and approving the System Test Plan, managing the test environment preparation, and tracking the testing schedule and results.  Product Owner Contributing to the test plan and test case specifications, reviewing test results |
| Lead QA Analyst | Designing a test plan, establishing a test repository, developing test case specifications, executing testing, and reporting defects. |
| Product Owner | Contributing to the test plan and test case specifications. Reviewing test results. |
| Lead Business Analyst | Contributing to the test plan and test case specifications, reviewing test results. |
| Lead Developer | Establishing and maintaining the test environment, assisting the Lead QA Analyst throughout the testing process. |
| DBA | Assisting the Lead Developer in establishing and maintaining the test environment. |

# TEST CYCLES AND SCHEDULE

The system test execution for ParkEz will be conducted in three test cycles, focusing on different aspects of the application:

Cycle 1. User Interface and Functionality

* This cycle concentrates on testing the user interface, navigation, account management and customer support.

Cycle 2. Advertising and basic machine learning

* This cycle focuses on advertising and basic parking lot occupancy recognition (how many spots are free in one camera).

Cycle 3. Payment and full parking features

* This cycle concentrates on payment, full lot occupancy recognition (including multiple cameras, tracking license plate and API for external websites)

See the schedule of the test execution cycles in the project plan.

# RISKS AND CONTINGENCIES

This section highlights potential risks and contingencies that may occur during the system testing process for ParkEz:

* Changes to the implementation scope or existing functional requirements can impact the test execution schedule.
* A large number of defects may require more time to fix and complete testing.
* Due to machine learning and computer vision necessary in project, project may turn out to be more complicated than originally expected, which may delay testing.