Master Test Plan

Author: Emily Carter
Version: v1.0
Beverage Booker

VERSION INFORMATION

Version	Date	Remarks	Author
v1.0	12/04/2020	First Test plan	Emily Carter

MANAGEMENT SUMMARY

Project objective

Beverage Booker is a system that will increase university cafes accessibility by giving them an online presence.

Test objective

The objective of the Master Test Plan (MTP) is to inform all who are involved in the test process about the approach, the activities, including the mutual relations and dependencies, and the (end) products to be delivered for the test project Beverage Booker.

Short description of the test approach

For this MTP the following test levels are acknowledged:

- Unit testing
- Integration testing
- System testing
- Acceptance testing

Table of Contents

1 Introduction

1.1 Project and project objective

Beverage Booker is a system that will increase university cafes accessibility by giving them an online presence. This project will support a mobile app that supports accounts, orders, reservations and payment methods for customers while giving an approximate time for order pickup.
It will provide the manager with the capability to edit the menu and other parts of the overall system.
Employees will be able to see the orders and update their status. Baristas and kitchen staff will be able to update order status for those using the app to signal completion of order.

This master test plan fits the project plan Beverage Booker.

1.2 Objective of the master test plan

The objective of the Master Test Plan (MTP) is to inform all who are involved in the test process about the approach, the activities, including the mutual relations and dependencies, and the (end) products to be delivered for the test project Beverage Booker.

The master test plan describes this approach, the activities and (end) products that need further elaboration in the other system test plans. These system test plans need to be abstracted from this master test plan.

2 Documentation

This section describes the documentation used in relation with the master test plan. The described documentation concerns a first inventory and will be elaborated, actualized and detailed at a later stage, during the separate test levels.

2.1 Basis for the master test plan

The following documents are used as the basis for this master test plan.

Document name	Version	Date	Author
Project Vision	v1.0		Jacob Kennedy

2.2 Test basis

The test basis contains the documentation that serves as the basis for the tests that have to be executed. The overview below describes the documentation that is the starting point for testing.

Document name	Version	Date	Author
Customer - Book a table	v1.0	26/03/2020	Benn Curby
Customer - Book event	v1.0	26/03/2020	Benn Curby
Customer - Create Account	v1.0	27/03/2020	Benn Curby
Customer - Empty Cart	v1.0	28//03/2020	Benn Curby
Customer - Fill Cart	v1.0	28//03/2020	Benn Curby
Customer - Place an order	v1.0	26/03/2020	Benn Curby
Customer - Search Menu	v1.0	28//03/2020	Benn Curby
Customer - Sign in	v1.0	27/03/2020	Benn Curby
Staff - Delete table booking	v1.0	28/03/2020	Benn Curby
Staff - Fill Order	v1.0	28/03/2020	Benn Curby
Staff - Search Table Bookings	v1.0	28/03/2020	Benn Curby
Staff - View Active Order	v1.0	28/03/2020	Benn Curby
Manager - Add menu item	v1.0	28/03/2020	Benn Curby

Manager - Delete an Event	v1.0	28/03/2020	Benn Curby
Manager - Delete Menu Item	v1.0	28/03/2020	Benn Curby
Manager - Edit an Event	v1.0	28/03/2020	Benn Curby
Manager - Edit Menu Item	v1.0	28/03/2020	Benn Curby

3 Test strategy

The time available for testing is limited; not everything can be tested with equal thoroughness. This means that choices have to be made regarding the depth of testing. Also it is strived to divide test capacity as effectively and efficiently as possible over the total test project. This principle is the basis of the test strategy.

The test strategy is based on risks: a system has to function in practice to an extent that no unacceptable risks for the organization arise from it. If the delivery of a system brings along many risks, thorough testing needs to be put in place; the opposite of the spectrum is also true: 'no risk, no test'.

The first step in determining the test strategy is the execution of a product risk analysis. This is elaborated in §3.1

The test strategy is subsequently based on the results of the risk analyses. The test strategy lays down what, how and when (in which test level) is being tested and is focused in finding the most important defects as early as possible for the lowest costs. This can be summarized as testing with an optimal use of the available capacity and time. The test strategy is described in §3.2.

3.1 Risk analyses

3.1.1 Product Risk Analysis

The product risks are determined in cooperation with the client and the other parties involved. Product risks are those risks associated with the final product failing to meet functional requirements and required system quality characteristics (NFRs) This product risk analyses (PRA) is comprised of two steps:

Product Risk	Characteristic	Description	Risk Class
1	Usability	To difficult to use or navigate	A
2	Reliability	Lose track of inventory in conjunction with sales and/or incorrect dollar transactions	A
3	Performance	Slow and unresponsive	B
4	Availability	Login fails, order failure, app crash	A
5	Security	Unauthorized or inappropriate use of users data	B

3.1.2 Technical Risk Analysis

Technical risks are determined in cooperation with the analyst/designers and programmers involved. Technical risks are development risks associated with failing to create a system that behaves according to specifications derived from requirements. (I.E. those aspects of development that pose particular challenges.) This technical risk analyses (TRA) is comprised of two steps:

Risk	Risk Area	Description	Risk Class
1	Place an Order (customer)	Unsuccessful transaction process	A
2	Shopping cart interactions	Unable to fill or empty cart	B
3	Booking Table (customer)	Unable to reserve a table due to incorrect seating amounts	C
4	Booking Event (customer)	Unable to reserve a event ticket due to incorrect ticketing amounts	C
5	Account Creation/login (customer)	Unable to create an account or login	A
6	Table booking (manager)	Unable to manage table bookings for correct seating amounts, unable to manage events	C
7	Event booking (manager)	Unable to manage event tickets for correct ticket amounts. Unable to manage events	C
8	Menu Items (manager)	Unable to manage menu items (delete, add or edit)	A
9	Fill order (staff)	Unable to successfully complete an order and notify customer	A
10	Table bookings (staff)	Unable to book tables or edit bookings	C

3.2 Test strategy

For each risk from the product risk analysis the risk class is qualifying the thoroughness of the test. Risk class A is the highest risk class and C the lowest. The test strategy is subsequently focused on covering the risks with the highest risk class as early as possible in the test project.

Risk	Description	RC	Test Level					
			SR	Unit	Int	FAT	UAT	ST
Usability	To difficult to use or navigate	A	●●●			●●●	●●●	●●
Reliability	Lose track of inventory in conjunction with sales and/or incorrect dollar transactions	B	●●		●●●	●●		●●
Performance	Slow and unresponsive	C	●●			●●		●
Availability	Login fails, order failure, app crash	C	●●			●●		●
Security	Unauthorized or inappropriate use of users data	A	●●			●●		
Place an Order	Unsuccessful transaction process	A	●	●●●	●●	●●	●●	
Shopping cart interactions	Unable to fill or empty cart	B	●	●●	●	●●	●	
Booking Table	Unable to reserve a table due to incorrect seating amounts	B		●		●●	●	
Booking Event	Unable to reserve a event ticket due to incorrect ticketing amounts	B		●		●●	●	
Account Creation/ login	Unable to create an account or login	A	●	●●●	●●	●●	●●	
Table booking (manager)	Unable to manage table bookings for correct seating amounts, unable to manage tables	C		●		●●	●	
Event booking (manager)	Unable to manage event tickets for correct ticket amounts. Unable to manage events	C		●		●●	●	
Menu Items (manager)	Unable to manage menu items (delete, add or edit)	A	●●			●●●		
Fill order (staff)	Unable to successfully complete an order and notify customer	A	●●		●●	●●		

Table bookings (staff)	Unable to book tables or edit bookings	C	●		●●	●	●	

Legend for the table above:

RC	Risk class (from product and technical risk analysis, where A=high risk, B=average risk, C=low risk)
SR	Static Review of the various intermediary products (requirements, functional design, technical design). Checking and examining artefacts without executing the software
Unit	Unit test and Unit integration test
Integration	Integration tests (low level (L), high level(H))
FAT	Functional acceptance test (alpha stage UAT)
UAT	User acceptance test (Beta stage UAT)
ST	System test (functional scenario testing (F), system quality scenario testing (S))
●	Limited thoroughness of the test
●●	Medium thoroughness of the test
●●●	High thoroughness of the test
<blank>	If a cell is blank, it means that the relevant test or evaluation level does not have to be concerned with the characteristic
RC	Risk class (from product and technical risk analysis, where A=high risk, B=average risk, C=low risk)

4 Test Levels

For this MTP the following test levels are acknowledged:

4.1 The Test Environments

Test Level	Test Environment
Unit Testing	Android Studio
Integration Testing	Android Studio
User Acceptance Testing	Emulator with multiple setups

4.2 The Test Tools

Test Level	Test Tools
Unit Testing	J-Unit
Integration Testing	Mockito
User Acceptance Testing	Emulator with multiple setups

5 Test Objectives

Risk	Test Goals	Risk Verification	Schedule
Place an Order	Ensure a successful transaction process	An order is passed from the customer to the sales system	During implementation of Place an Order & fill cart use case.
Shopping cart interactions	Ensure that the cart can be correctly interacted with to enable proceeding to place an order	Items are able to be added/removed from the cart and the cart can be then processed/passed correctly to the place an order use case	During implementation of Fill cart and Empty cart use cases
Booking Table	Ensure reserving a table with correct seating amounts	When a table is reserved it correctly deducts from the total available tables. The reservation is added to the user ID	During implementation of the book a table use case.
Booking Event	Ensure reserving a event ticket with correct ticketing amounts	When an event ticket is reserved it correctly deducts from the total available tickets for the event. The reservation is added to the user ID	During implementation of the book an event use case.
Account Creation/ login	Ensure that an account creation and login	An account can be successfully created with the ability to login and logout	During implementation of Create Account & Sign in use cases.
Table booking (staff)	Ensure table bookings can be managed for correct seating amounts,	The number of tables available can be adjusted to ensure correct availability	During implementation of the search table bookings use case.
Event booking (manager)	Unable to manage event tickets for correct ticket amounts. Unable to manage events	The number of ticket is can be adjusted per event to ensure correct availability	During implementation of the edit an event use case.
Menu Items (manager)	Unable to manage menu items (delete, add or edit)	All menu items can be managed via deleting, adding, or editing	During implementation of the edit menu item, add menu item & delete menu item use case.
Fill order (staff)	Unable to successfully complete an order and notify customer	Once a order is received it can be marked as completed and the customer will be notified that the order has been completed	During implementation of the fill order use case.
Table bookings (staff)	Unable to book tables or edit bookings	Tables can be booked & edited ensuring that the table numbers are correct	During implementation of the search table bookings & delete table booking use case.