GÖTÜR Project

System-Wide Requirements

Version 1.0

Prepared By:

MESE Company

Revision History

|  |  |  |  |
| --- | --- | --- | --- |
| **Date** | **Version** | **Description** | **Author** |
| 01/04/2022 | 1.0 | Draft 1 | MESE TEAM |

**Table of Contents**

[1. Introduction 4](#_Toc99822290)

[2. System-Wide Functional Requirements 4](#_Toc99822291)

[3. System Qualities 5](#_Toc99822292)

[3.1 Usability 5](#_Toc99822293)

[3.2 Reliability 5](#_Toc99822294)

[3.3 Performance 6](#_Toc99822295)

[3.4 Supportability 6](#_Toc99822296)

[4. System Interfaces 8](#_Toc99822297)

[4.1 User Interfaces 8](#_Toc99822298)

[4.1.1 Look & Feel 8](#_Toc99822299)

[4.1.2 Layout and Navigation Requirements 8](#_Toc99822300)

[4.1.3 Consistency 8](#_Toc99822301)

[4.1.4 User Personalization & Customization Requirements 8](#_Toc99822302)

[4.2 Interfaces to External Systems or Devices 8](#_Toc99822303)

[4.2.1 Software Interfaces 8](#_Toc99822304)

[4.2.2 Hardware Interfaces 9](#_Toc99822305)

[4.2.3 Communications Interfaces 9](#_Toc99822306)

[5. Business Rules 10](#_Toc99822307)

[5.1 Purchase Promotions 10](#_Toc99822308)

[5.1.1 Initial Purchase – ID 1 10](#_Toc99822309)

[5.1.2 Consecutive Purchase – ID 2 10](#_Toc99822310)

[6. System Constraints 10](#_Toc99822311)

[7. System Compliance 10](#_Toc99822312)

[7.1 Licensing Requirements 10](#_Toc99822313)

[7.2 Legal, Copyright, and Other Notices 10](#_Toc99822314)

[7.3 Applicable Standards 10](#_Toc99822315)

[8. System Documentation 11](#_Toc99822316)

GÖTÜR

System-Wide Requirements Specification

# Introduction

This document is aimed to supply system-wide requirements that are not described as use cases for the project GOTUR.

# System-Wide Functional Requirements

* The system will store all purchases with their order details for various purposes such as fraud detection and prevention, trend analysis.
* Access to the system shall use OAuth 2.0 protocol to authenticate all users.
* The system will store all login information with its metadata up to last 6 months to ensure the user security.
* The system will logout the user after 15 minutes of idling.
* Users will be able to request deletion of their account and all data related to their account.
* The core module of the system will be subject to intellectual property rights.
* The main dependency source will be an artifact repository such as mvnrepository. Through the Maven Artifact index, all licensing info will be available.
* The system shall comply the licensing requirements as per dependencies. This also applies to cases where an artifact usage from another source such as GitHub.
* All dependencies used are expected to be explicitly licensed as open source.
* All data, hence the database applications and clients are expected to be secure.
* The database instances will only be accessible by the application servers.
* Servers that run the database engines will reject all connections sourcing from the public internet.
* System will not store any passwords in plain text.
* All system will communicate over TLS.
* System will process online purchases only thorough BTK.
* System can use services such as Cloud Flare for security purposes.

# System Qualities

## Usability

* The system will allow a customer to create their accounts through a trusted credential providers such as Facebook and Google accounts.
* The system will require and verify only e-mail and phone number upon account creation for customers.
* The system will resemble the existing systems to familiarize the users and reduce the learning curve.
* A customer should be able to give an order within 1 minutes upon selection of desired goods.
* The system will provide step back functionality to allow users to edit their order related information.
* A customer will be able to clear all of their selections for a purchase at any time until successful order.
* System messages shown to user will include current state of the process as well as next actions that can be performed.
* Customers will be able to give ratings of their experience of using the system upon completion of their purchases.
* The system will provide next step to complete the order within 10 seconds.

## Reliability

* System will be available 7/24.
* Functional uptime of the system will be 164 hours per week (98%).
* Maintenance window for the system will be 4 hours per week.
* System will be subject to a general health check mechanism that checks the status of all system components every 30 seconds.
* The system will reserve the selected goods for the customer for 3 minutes to prevent out of stock incidences.
* Warehouses will reserve up to 10% of the goods to supply ongoing purchase operations.
* System will only support payment via credit/debit card to prevent accounting errors.
* System will run on multiple instances in isolated environments for redundancy.
* Upon significant service availability issues, mean time to repair shall be 1 hours.
* Upon detection of minor defects such as GUI glitches, mean time to repair shall be 24 hours.
* Upon detection of critical errors resulting with the general operation of the system is degrading, system will only allow idempotent operations.
* System shall have restore points for databases to allow rollback to a previous deployment for each release.
* System shall store read replicas for all databases up to 14 days for data integrity.
* Data older than 1 month will be removed from the actual database and archived in an isolated data vault.
* The system will allow maximum 4.5 released defects per function point.
* Upon detection of online payment issues, system will prompt to offer payment at the door.
* In case of an unforeseen error where repair time cannot be estimated, system will serve a static maintenance page to the users.

## Performance

* System will allow 100 concurrent users performing any operation on the system for the initial release.
* The web page will be loaded within 10 seconds assuming that the client meets the required conditions.
* System will respond within 5 seconds on a user input including communication overhead.
* A new instance for any system module shall startup in 15 seconds.
* An order shall be completed with maximum 15 transactions.
* The load will be balanced with round robin strategy when there is more than 1 identical instance running for the same system function.
* Overall order processing speed will depend on available couriers.

## Supportability

* System will run on Linux based environments.
* Principal development and operations shall be performed on Ubuntu Server LTS.
* System shall support containerization with Docker.
* User interface will be available on all major browsers.
* User interface will support different resolutions.
* All system modules shall support external configuration via variables or files.
* System shall scale up to support 10000 concurrent transactions.
* System will scale up by increasing the number of running servers upon reaching total computational load of 70% to prevent throttling.
* System will scale down by shutting down servers that is under 20% load (i.e. idle servers).
* System will provide FAQ for the customers.
* Every software component developed will have step by step documentation packaged with its source code to define the overall functionality and how to build the executable artifact from it.
* Software build process will be automated where applicable.
* Every operation that can be performed by non-customer users shall have a quick guide.
* System shall provide an instant messaging functionality to both customers and helpdesk operators.
* A new product shall be able to be added to the system within 10 minutes by the store manager.
* A new release will be provided every 3 weeks.
* Maintenance and fix releases will be provided whenever necessary.

# System Interfaces

## User Interfaces

### Look & Feel

* The product will have single web page for each type of user.
* Customers shall be able to add/remove items with a single click.
* Customers shall be able to add/remove multiple items from the same page.
* Store manager will be able to edit all details of available products.
* Product shall be created with conventional e-commerce look for customers.
* All planes and panels will be round tipped.

### Layout and Navigation Requirements

* Products shall be grouped together with their respective categories.
* Landing page will contain the most preferred products.
* Customers will be able to navigate back to prior page until successfully placing the order.
* Couriers will be able to mark the order status.
* Store manager will be able to list order history.

### Consistency

* System shall navigate to next page only on successful completion of current transaction is completed on the server side.
* System will show a loading prompt to user during the transactions.
* Undoable actions will prompt a confirmation message.
* System will store the latest session details to recover the current state of a user in case of connection errors.
* System will present the products in the same order and layout.
* System shall support different item listing preferences.
* All interactions will prompt a temporary success/failure message to the user.
* In case of an inconsistency between the user session and server state, the state at the server side will be considered to true.

### User Personalization & Customization Requirements

N/A

## Interfaces to External Systems or Devices

### Software Interfaces

* System shall process payments with official API’s offered by BTK.
* System can navigate customers to relevant pages for payment process steps if required.

### Hardware Interfaces

* The system will run on Amazon Web Services.
* No user data will be stored or processed in the on premise hardware.
* Hardware specs and Amazon Web Services availability rules will apply to the system.

### Communications Interfaces

* System will interact and process data with its own Rest API.
* System shall use a third-party API verified by the BTK for the payment process.

# Business Rules

## Purchase Promotions

### Initial Purchase – ID 1

* If a customer is using the system for the first time and the total cart value is greater than 50TL, 5% discount up to 10TL will be applied to that order.

### Consecutive Purchase – ID 2

* If a customer completes 10 different purchases greater than 100TL within a month then 50TL voucher coupon will be given to a customer.
* This rule applies only once per month.

# System Constraints

* System shall be implemented as a single monolithic application for rapid prototyping.
* System will be designed in modular way so that any given module can be extracted as a new software system to support scaling different factors of the application.
* The main programming language for the system implementation is Java 11.
* System application should be usable with any Java EE web application container that is compatible with the Servlet 2.4 and JSP 2.0 specifications.
* The view technologies that are to be used for rendering the application are Java Server Pages (JSP) along with the Java Standard Tag Library (JSTL).
* A single application server instance shall have at least 8GB of RAM, 2.4 GHz CPU and 20 GB free disk space.
* The general third-party technologies required for the system is as following:
  + Java SDK 1.11.x or later
  + Ant 1.7.x
  + Tomcat 5.x.x
  + JUnit 4.4
  + Maven 3.4.x

# System Compliance

## Licensing Requirements

N/A

## Legal, Copyright, and Other Notices

* The system will provide service to the customers that accepted the Distant Sale Agreement.

## Applicable Standards

* System will comply with KVKK.
* System will comply with BTK standards.
* System will comply with BDDK rules for payment processing.

# System Documentation

* All modules and architectural rules applied to the system will be documented with Markdown format for development and operations activities by the developers and IT employees that run the system.
* All documentation, including the User manuals, Quick Guides etc. will be stored on the version control system.
* User manual will be presented as Frequently Asked Questions (FAQ).
* FAQ will be created by the developer/designer that created the actual system or visual presentation.