Fontys University of Applied Sciences

Event manager

Setup document

**ProP**

**Code Embassy**

Ivelin Slavchev

Vasil Sirakov

Lyubomir Yankov

Marina Tzenkova

Contents

[Preface 4](#_Toc477301813)

[Client 4](#_Toc477301814)

[Processes 4](#_Toc477301815)

[Use cases 4](#_Toc477301816)

[*Scenario 1:* Get information about event 4](#_Toc477301817)

[*Scenario 2:* Sign up user 4](#_Toc477301818)

[*Scenario 3:* Sign in user 5](#_Toc477301819)

[*Scenario 4:* Buying tickets 5](#_Toc477301820)

[*Scenario 5:* Reserving camping spot 6](#_Toc477301821)

[*Scenario 6:* Entering festival 6](#_Toc477301822)

[*Scenario 7:* Enter camping 7](#_Toc477301823)

[*Scenario 8:* Buy food from shop 7](#_Toc477301824)

[*Scenario 9:* Borrow item 7](#_Toc477301825)

[*Scenario 10:* Return borrowed item 8](#_Toc477301826)

[*Scenario 11:* Leave festival 8](#_Toc477301827)

[Functional requirements 9](#_Toc477301828)

[Overview 9](#_Toc477301829)

[Web application 9](#_Toc477301830)

[Entering festival application 9](#_Toc477301831)

[Camping application 9](#_Toc477301832)

[Shop management application 9](#_Toc477301833)

[Loan stand application 9](#_Toc477301834)

[Leaving event application 9](#_Toc477301835)

[Statistics application 9](#_Toc477301836)

[PayPal support application 10](#_Toc477301837)

[MoSCoW list 10](#_Toc477301838)

[Must have 10](#_Toc477301839)

[Should have 10](#_Toc477301840)

[Could have 10](#_Toc477301841)

[Won’t have 10](#_Toc477301842)

[User interface 11](#_Toc477301843)

[Check in application GUI 11](#_Toc477301844)

[Check out application GUI 11](#_Toc477301845)

[Camping application GUI 12](#_Toc477301846)

[Loan items application GUI 12](#_Toc477301847)

[Shop management application GUI 13](#_Toc477301848)

[Statistics application GUI 13](#_Toc477301849)

[Web application front end 14](#_Toc477301850)

# Preface

Setup document for an event management system developed for a music festival. The document contains a Client chapter which describes the client and the agreements made with them. The Processes chapter describes the use cases of the system and the handling of exceptional cases. Functional requirements chapter is used to define all requirements and to group them by applications. User interface chapter describes the how the user interface will look and what is planned for the user experience.

# Client

The client is a company that organizes musical festivals and events. The client wants a system that will help manage an upcoming music festival. The agreed deliverables are a website for the festival as well as some applications that will be utilized during the event itself.

# Processes

## Use cases

### *Scenario 1:* Get information about event

*Actor:* User

*Pre:* User is browsing for events.

*Post:* User has received information about festival.

*Steps:*

1. User visits website.
2. User visits Events page.
3. User receives information about each event.
4. User visits About page.

### *Scenario 2:* Sign up user

*Actor:* User

*Pre:* User has information about festival.

*Post:* User has signed up as visitor.

*Steps:*

1. User clicks Sign up and a sign up form is presented.
2. User fills required information.
3. User submits information.
   1. Username is taken.

*Solution:* System prompts user to choose another username.

* 1. Passwords do not match.

*Solution:* System prompts user to confirm password again.

1. User is redirected to Sign in form.

### *Scenario 3:* Sign in user

*Actor:* User

*Pre:* User has signed up as visitor.

*Post:* User has signed it.

*Steps:*

1. User clicks Sign in and a sign in form is presented.
2. User fills required information.
3. User submits information.
   1. Username/password combination is wrong.

*Solution:* System prompts user to enter details again.

1. User is signed in and is redirected to Dashboard page.

### *Scenario 4:* Buying tickets

*Actor:* User

*Pre:* User has signed in.

*Post:* User has bought tickets.

*Steps:*

1. User visits Event page.
2. User clicks on Get tickets and is taken to Tickets page.
3. User fills required information for reserving a ticket.
4. User submits personal information.
5. User is taken to payment form.
6. User submits payment information.
   1. Payment details incorrect.

*Solution:* System prompts user to enter details again.

* 1. Transaction failed.

*Solution:* System prompts user to make payment again.

1. User is redirected to Dashboard page and is presented with a QR code for his ticket.

### *Scenario 5:* Reserving camping spot

*Actor:* User, Staff

*Pre:* User has logged in and has bought tickets.

*Post:* User has booked a camping spot.

*Steps:*

1. User visits Camps page.
2. User fills required information.
3. User submits information.
   1. Participants given more than maximum amount.

*Solution:* System prompts user to reduce participants.

* 1. Chosen camping spot is taken.

*Solution:* System prompts user to choose a free spot.

1. User is taken to payment form.
2. User submits payment information.
   1. Payment details incorrect.

*Solution:* System prompts user to enter details again.

* 1. Transaction failed.

*Solution:* System prompts user to make payment again.

1. User is taken to Dashboard page and is presented with a QR code for his camping.

### *Scenario 6:* Entering festival

*Actor:* User, Staff

*Pre:* User has bought tickets.

*Post:* User has entered festival.

*Steps:*

1. User arrives at festival entrance.
2. User gets his QR code scanned by Staff.
3. System validates ticket.
4. Ticket is invalid/not paid for.

*Solution:* Refuse entry to User.

1. User is checked in.

### *Scenario 7:* Enter camping

*Actor:* User, Staff

*Pre:* User has entered festival and has paid for a camping spot.

*Post:* User is allowed to camp.

*Steps:*

1. User arrives at camping entrance with group.
2. User gets his QR code scanned by Staff.
3. System validates booking
   1. Booking is invalid/not paid for.

*Solution:* Refuse camping to User.

1. User is checked to camping spot.

### *Scenario 8:* Buy food from shop

*Actor:* User, Staff

*Pre:* User has entered festival.

*Post:* User has purchased food.

*Steps:*

1. User arrives at shop.
2. User chooses product.
   1. Item not available.

*Solution:* Prompt User to choose another item.

1. Staff chooses product from system.
2. User pays for product.
   1. Insufficient funds.

*Solution:* Prompt choosing of another item.

1. User receives product.

### *Scenario 9:* Borrow item

*Actor:* User, Staff

*Pre:* User has entered festival.

*Post:* User has borrowed an item.

*Steps:*

1. User goes to loan stand.
2. User chooses item.
   1. Item not available.

*Solution:* Prompt User to choose another item.

1. User is registered as borrower for the chosen item
2. User receives item.

### *Scenario 10:* Return borrowed item

*Actor:* User, Staff

*Pre:* User has borrowed an item.

*Post:* User has returned the item.

*Steps:*

1. User goes to loan stand.
2. Staff checks borrowed items.
3. User returns item.
   1. Item broken or lost.

*Solution:* Enforce additional fees on User.

1. User pays the loan fee.
   1. Insufficient funds to pay fees.

*Solution:* Enforce User to pay.

1. Item is logged as returned.

### *Scenario 11:* Leave festival

*Actor:* User, Staff

*Pre:* User has entered festival.

*Post:* User has left festival.

*Steps:*

1. User goes to festival exit.
2. Staff scans User.
3. System checks User for any borrowed items and pending payments.
   1. Borrowed item not returned.

*Solution:* Enforce User to return item and pay additional fees.

* 1. Unpaid fees.

*Solution:* Enforce User to pay.

1. User is checked out of festival.

# Functional requirements

## Overview

### Web application

* Provide information about festival
* Register visitors
* Offer tickets
* Offer camping spots

### Entering festival application

* Check ticket and payment
* Check in visitor

### Camping application

* Check visitor camping reservation

### Shop management application

* Offer food and drinks
* Provide sales information

### Loan stand application

* Offer items for rent
* Register loaned item and visitor
* Provide information about loaned items

### Leaving event application

* Check visitor payments and loans
* Enforce visitor to return item and pay fees if needed
* Check out visitor from event

### Statistics application

* Provide statistical information about festival
* Statistics about visitors – total number, expected amount of visitors, people that have left the festival
* Statistics about each visitor – paid entrance price, purchases at shop, borrowed items, total money transferred to event account, current status
* Statistics about finances – total balance of all visitors, total amount paid
* Statistics about camping – bookings and payments
* Statistics about shop and loan stand – total amount of sales, total sold units per article.

### PayPal support application

* Support PayPal payments
* Save logs to database

## MoSCoW list

### Must have

* Website with information about festival
* Ability to reserve and buy tickets
* Ability to rent camping spots
* Check and verify tickets at festival entrance
* Functionality for selling food and drinks at shop
* Functionality for loaning items at loan stand
* Ability to check out and leave festival
* Ability to provide statistical information about event and visitors

### Should have

* Check visitors before leaving event for unreturned items
* PayPal transactions
* Ability to refund tickets if conditions are met

### Could have

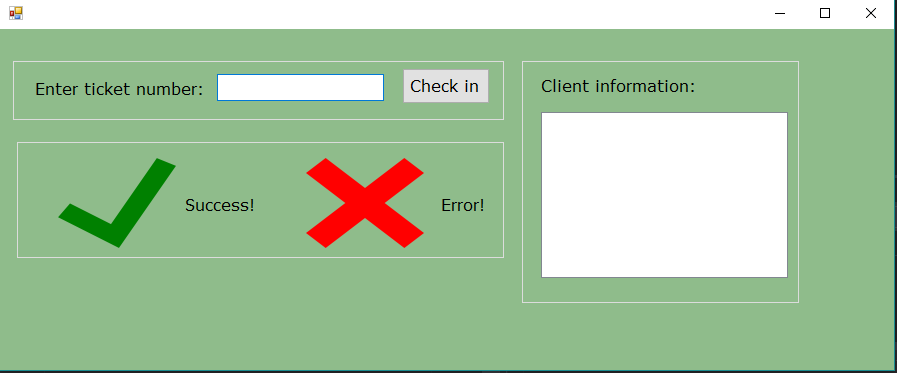
* Separate cheaper ticket for some artists only
* Tickets for certain days only

### Won’t have

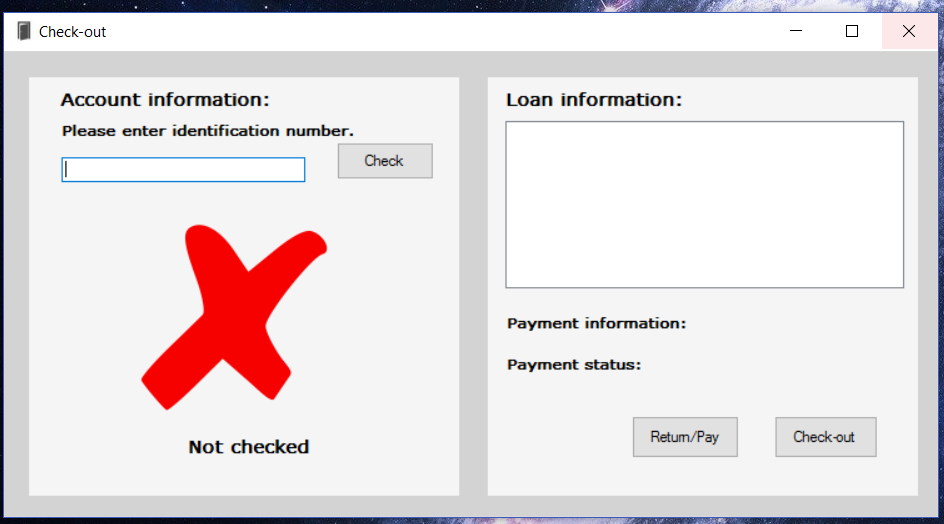
* User manual for site and apps

# User interface

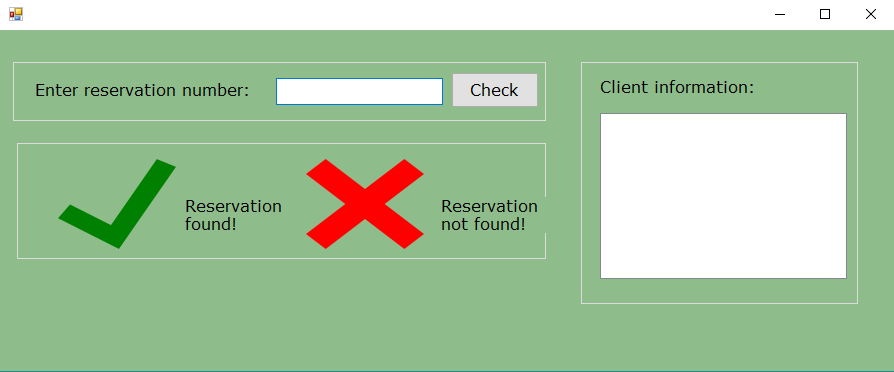
## Check in application GUI



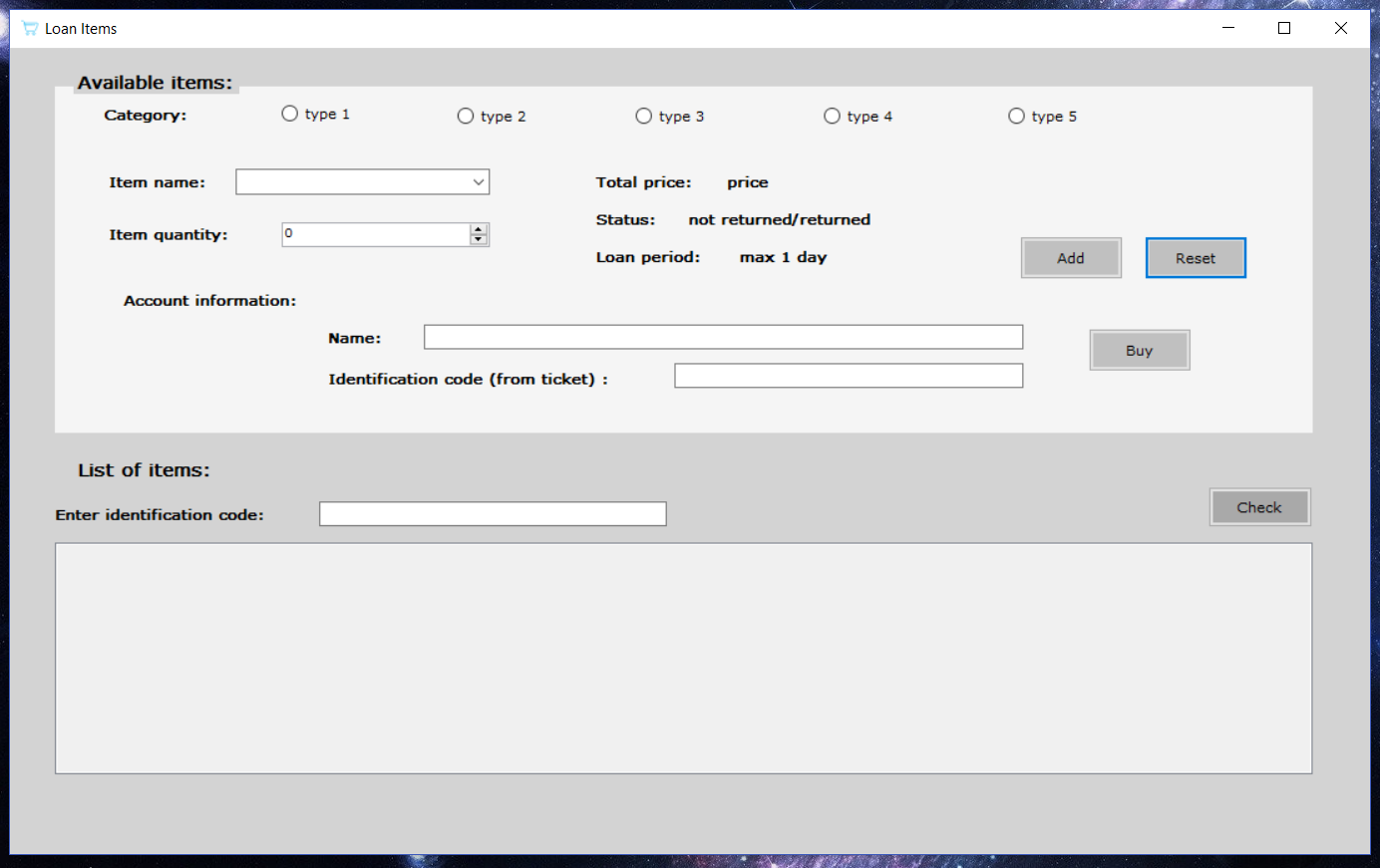
## Check out application GUI



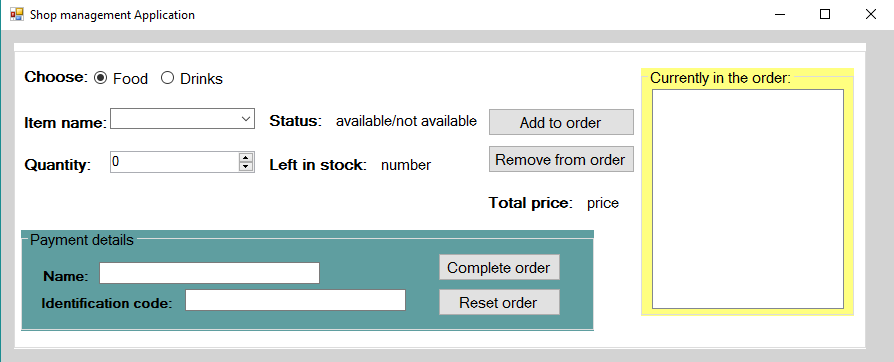
## Camping application GUI



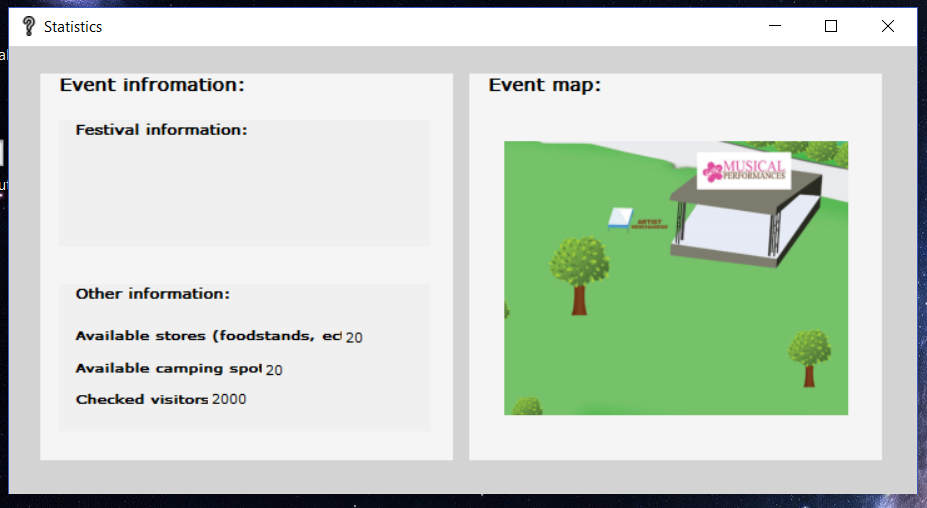
## Loan items application GUI



## Shop management application GUI



## Statistics application GUI



## Web application front end

