

EventTicket

Part 1. Creating the system vision

1. Business Context

This system is a perfect solution for any event planners. It offers you the mobility of attendances check-in, the possibility to keep track of your visitors and of course collect information about your costumers.

2. Business Opportunities

This simple implementation gives you great possibilities of usage for this system in different circumstances, it could be used where ever a ticketed or web promoted event is present. Simple web interface gives you a variety of possibilities in creating your events and keep track of your audiences. Is it a MOVIE PREMIERE, PROFFESIONAL CONFERENCE or a SOCIAL THEATER event, one can create a schedule for it, make a simple configuration for the place, time, number of available tickets or dedicated sessions, then launch the event and wait for your costumers. Simple way to get a QR ticket on the go, makes this solution perfect for modern society.

3. Problem Description

- Problem

Integrating a complex mechanism in a simple solution for automating event attendance.

- Concerns

Lots of social events need a dedicated phone number or a place to buy a ticket. All events are strongly promoted in social networks, but none give a flexible possibility to buy a ticket on the move.

- Consequences

Main barrier for users is the idea of giving money on a 2D black and white image. The culture of a granny at the entrance with a pack of tickets will pursue for some time.

- Successful solution

Key points in this solution:

- Automation of the purchasing a ticket
- No intermediates for your events
- Strong promoting over social networks
- Simple user database creation
- Targeted advertising
- Cheap costs

- Task of the system

If you attend an event, it should be simple to get a ticket. If you plan an event it should be simple to keep track of your attendances

Part 2. Identifying the stakeholders

Stakeholders	Roles
Customers	value, quality, customer care, friendly interface.
Suppliers	providers of products and services used in the end product for the customer, equitable business opportunities.
Creditors	new contracts, liquidity, targeted ads.
Owner(s)	have interest of the success of his/her business.
Investors	have interest earning income from investment, targeted ads.

Main stakeholders are all the customers that will purchase tickets or organize events, simplicity and intuitiveness. Owners, creditors and investors have interest in earning a database of users for targeted ads.

Main responsibility is to keep the system operating: the main server and all client qr_verifier terminals, as well the availability of the web service for client devices.

Part 3. Documenting the functional requirements of the system

The functional requirements of the selected system define the functionality the system is to provide. From these you will later derive the use-case scenarios for your application. In the event that your application interacts with some third-party services, investigate the feasibility of the used services.

The functional requirements should be provided in the following format:

- Requirement
 1. Available 24/7 web based service;
 2. Secured main server;
 3. Secured connection between server-client and server-terminal;
 4. Log in as a customer and event planer;
 5. RESTful web service Client – Server;
- Description
 1. Main server should be available 24/7 hosted in a secured place with full maintenance;
 2. Server should have a secure shell tunneling in order to communicate securely with it;
 3. All the information shared between client terminal and server should have a certain level of security;
 4. Web server should have a registration form and a user management system in order to differentiate type of users;
 5. There should be a simple, cacheable and light weighted api to optimize data transfers.

One of the main goals is to guarantee that each ticket would be unique and there will be no possibility to generate valid tickets outside our system.

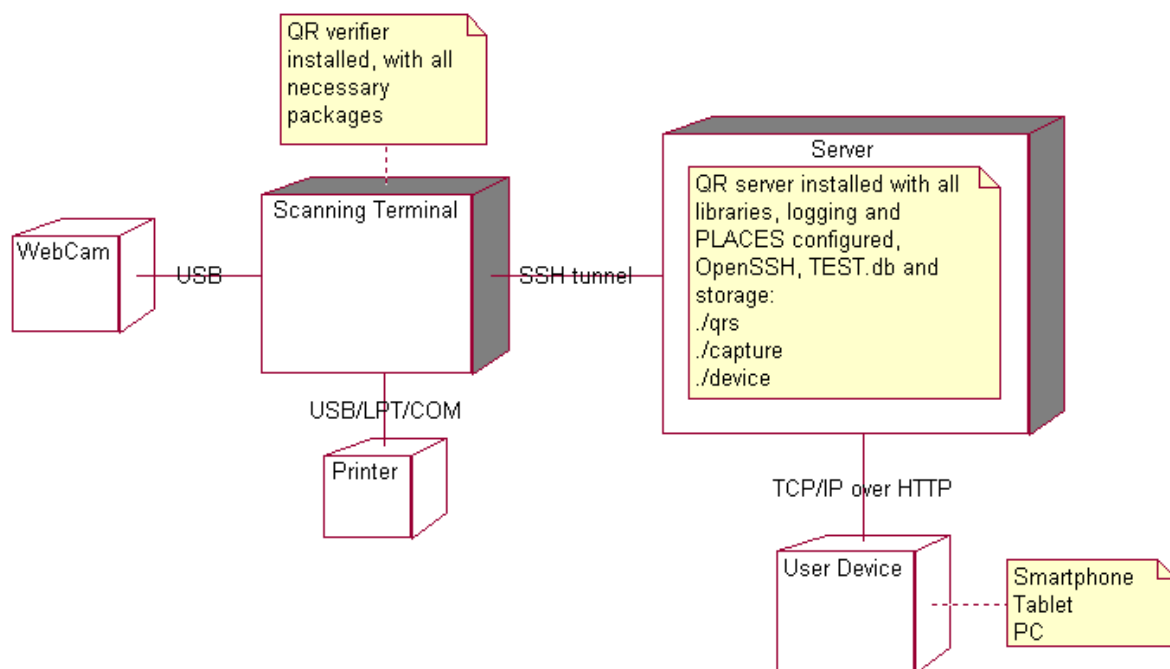
Part 4. Non-functional requirements

The main server, requester and a client prototype should be provided as soon as possible in order to show a prototype to the investors and in order to obtain billing api from Moldcell, Maib and Orange.

- Requirements
 1. First prototype;
 2. General web service design;
 3. Mobile design;
 4. Set of configurations for different events;
 5. Terminal designs.
- Description
 1. Raw prototypes should be ready within 2 weeks;
 2. Web site design in order to start building the template;
 3. Mobile view design for website, that will show up in apps layout views;
 4. Event simple configurations in order to differentiate events and set DBs for events;
 5. Designing the terminal stations, reading device, printing device.

Main system should be operating on a hosted dedicated server, together with the web service; the terminal service should be running on several types of devices: laptop, mobile device, nettop, micropc(arduino).

Part 5. Architectural sketch



Functional grouping is not needed because the system is simply structured, each functional module should be developed together in order to implement base functionalities; request qr, generate qr, verify scanned qr. All side functionalities can be developed in a later stage.

Part 6. Select an architecture

Client/server

Rich Client vs. **Thin Client**

RESTful API

There are 2 client/server layers, one for customers that connect to the web service and the second one is the web service and the main qr server. First is a thin client representing a web view in an app and the second is a rich client that communicates with the main server through RESTful api.

Part 7. Architectural principles

Loose coupling

The components of the system will be loose coupled, so that there are as few dependencies between modules leading to flexibility of module interchange. For the QR encoding/decoding process, the component should not care about the other components that are going to use its output, so that it can be easily replaced if there is a need.

Separation of concerns

The components responsible for the QR processing will be separate, from the component responsible for the RESTful service, as well as the one from the information visualization.

Information hiding

The system responsible for ticket identification will not send the actual information via the service for the ticket validation process, but rather a hash of a part of the data, to be able to identify and validate the original information, but without exposing the critical/private data of the ticket.

Interface segregation

The interface for generating and scanning the QR codes will be separate from the ticket aggregation and visualization interface, making sure there are no hardcoded dependencies between visualization and processing.

Modularity

This system's modularity is based on the usage of the RESTful service that is the main component used for communication of data between different endpoints. Because the service has a well-defined protocol, many front ends can be used to display the information. Also because the protocol is strictly defined, we can change the implementation underneath the backend at any time.