



QrCompetition

*J2ME Prize-winning Competitions Software
Based on QR-codes Scanning.*

a.a. 2009/2010

Gaetano Catalli
Matteo Landi
Simone Mainardi



Introduction

- With this software users can participate to prize-winning competitions by scanning QR codes they found on magazines, newspapers, etc.
- A simple web administration front-end allows administrators to create and manage new competitions and to generate the relative QR codes through the Google charts APIs.
- If a user win, he/she will receive a challenge code which will be stored in the device and told to the operator while redeeming the prize. The operator should mark that challenge code as redeemed in the administration so that the user cannot redeem the price more than once.



Administration

- Creation of new Competitions

Competitions

Name

Prizes Qty

Expire Date

Notes

		July, 2010								
		Today								
wk	Sun	Mon	Tue	Wed	Thu	Fri	Sat			
25					1	2	3			
26	4	5	6	7	8	9	10			
27	11	12	13	14	15	16	17			
28	18	19	20	21	22	23	24			
29	25	26	27	28	29	30	31			

Select date

Powered by M2Advisor DevTEAM

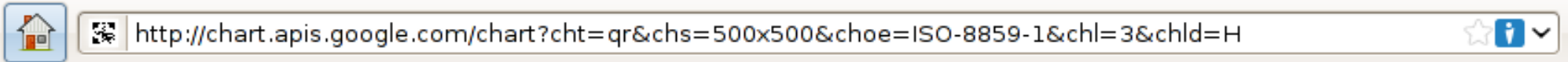
- Login the administration web front-end

Login

Username:

Password:

- QR code generated through google APIs



- Administration back-end written in Python and SQLAlchemy as ORM. Sqlite has been used as database but switches to others DBs such as MySQL or Oracle are simple one-line edits.



J2ME Mobile Application

- CLDC-1.1, MIDP-2.0
- Multithreaded:
 - Splash Screen Thread which keeps the user entertained while performing initialization operations
 - Snapshot Thread for taking snapshots of the camera that will be delivered to the...
 - CompWorker Thread which actually decode QR Codes and interact both with the remote server to check the result of the competition (e.g. Win+challenge, Lose) and with the RMS for recoding that information
 - TimerTask which periodically contacts the server for updating local RecordStore records. In this way the local storage is always up to date and does not contain obsolete information (e.g. expired competitions)



Mobile Application Features



- **Splash Screen**

Displayed during startup while initializing the application



- **Video Capture**

The user can take snapshots of the camera and the app will (try to) decode the information present in the QR code captured (if any)

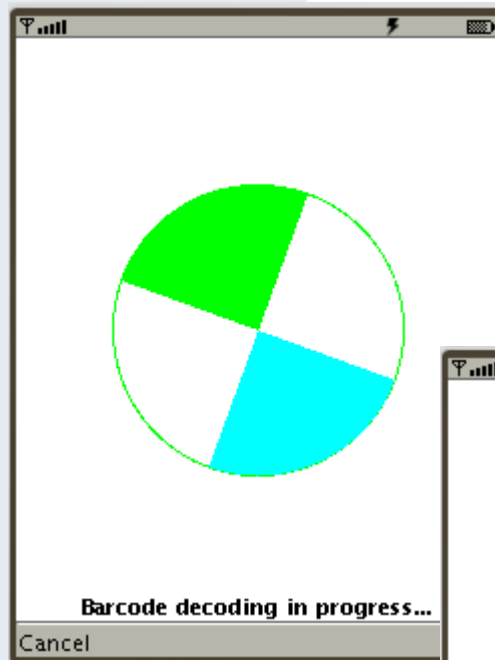


- **Menu**

From the menu the user can browse the history of the previous competitions he/she has won, lost or for which has redeemed the prize.

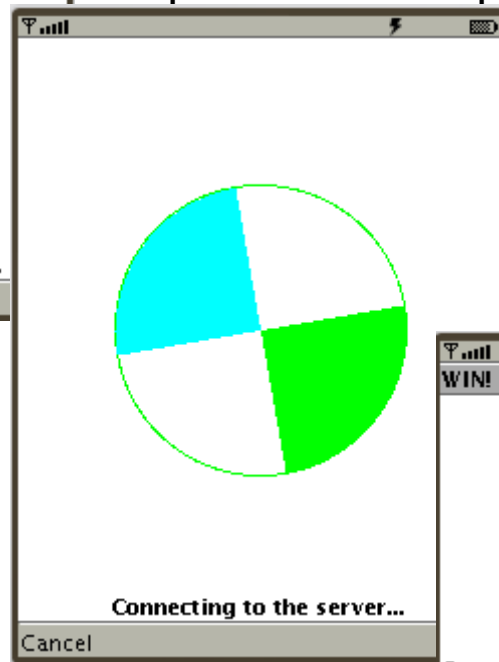


Mobile Application Features



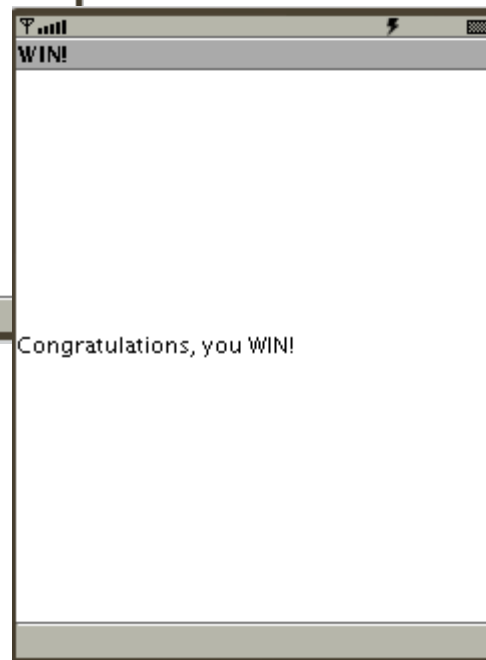
- **Barcode Decoding**

keep the user entertained while decoding the bar code. This operation is actually the longest and most expensive for the device. The user has the ability to stop the current operation



- **Server Connection**

Connect to the remote server and decide whether the user has won the competition.

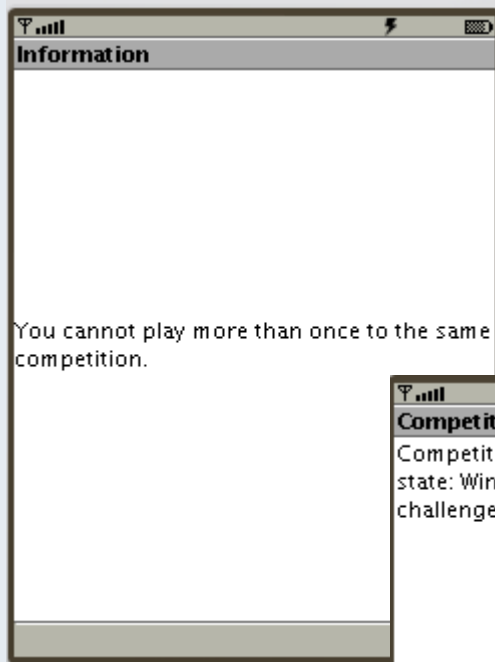


- **Win!**

A notification will be displayed on the screen after the response from the server has been received

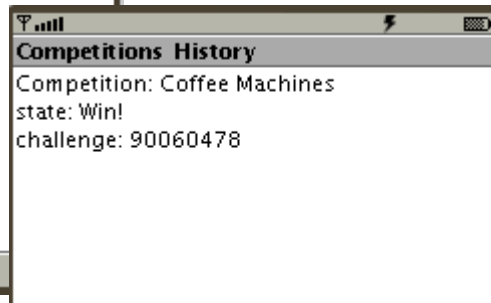


Mobile Application Features



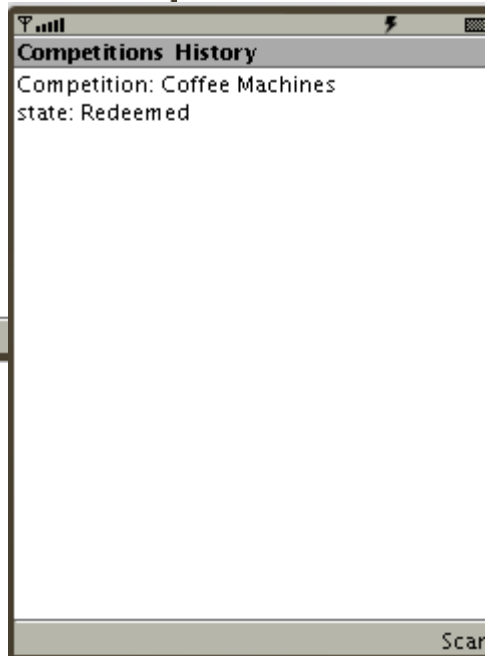
- **Only One Try per Competition**

The application does not allow the user to participate to the same competition multiple times



- **History**

Every previous competition played is displayed in the history. If the user won he/she can also read the challenge code generated by the server



- **History**

When the user redeems the prize an administrator should mark the relative challenge code as redeemed. In this way the next time the apps contacts the server can update the local Record Store accordingly.



Notes On The Implementation

Mobile Application

- JavaME <http://java.sun.com/javame/index.jsp>
- MIDP-2.0 <http://jcp.org/aboutJava/communityprocess/final/jsr118/index.html>
- CLDC-1.1 <http://jcp.org/aboutJava/communityprocess/final/jsr139/index.html>
- Wireless Toolkit 2.5.2 for CLDC <http://java.sun.com/products/sjwtoolkit/>
- NetBeans IDE <http://www.netbeans.org>



• Third Parties Libraries

- ZXing (ZebraCrossing) for QR-Codes decoding <http://code.google.com/p/zxing>
- JSON-ME for data exchange between the server and the mobile app using JSON <http://www.json.org/java/>

Administration

- Python 2.6 <http://www.python.org>
- SQLAlchemy 0.6.2 <http://www.sqlalchemy.org>
- Turbogears 2 (Python Web Framework) <http://www.turbogears.org>
- Sqlite 3.6.23 <http://www.sqlite.org>
- Google Charts APIs for QR-Codes generation <http://code.google.com/apis/chart/>

