VIA UNIVERSITY COLLEGE
ICT ENGINEERING

Project Description

Betting system

Group 10, SDJI2-T-S16
Daniel Hamarik (239 857)
Atanas Latinov (239 841)
Stefan Alexiev (240 278)

Supervisors
Henrik Kronborg Pedersen
Jan Munch Pedersen

April 2016

Project Description

Background description

With the increasing number of sport events of various kind, both on a national and international level, and the huge and still growing spectator base of those events, it is only logical to use the power of today's technology and the internet to provide a system that enables people to bet on events on the other side of the planet in the comfort of their own home.

Nowadays, with the emerging of new events, many bookmakers fail to cover many of them that are for example very popular in other parts of the world. Not only that but also many people do not have the chance to place their bets because they are either far from a bookmaker or it is just impossible for them. It is clear that these complications and many more can be easily avoided just by using the power that today's technology provides.

From a business standpoint, the budget for such an endeavor would be hugely reduced since the betting shops are rendered useless. Therefore, there are no expenses for their staff, maintenance, bills, etc. Another thing is that the papers on which bets are written by the people, will not be used, thus less production costs and less pollution of the environment.

Considering the obligation to upkeep many betting shops as well as manage a number of employees in each of the shops, the business becomes unnecessarily more complex and slower. By storing the data digitally, it is safe to say that working with it is much faster and the user's productivity is increased tremendously.

Purpose

The purpose of this project is to create a smart betting solution, that enables its users to easily place bets, review match scores and see their betting history.

Problem formulation

The focus of the system is to provide events to the users and enable them to place their bets on them using a client/server architecture. The server will be responsible for collecting the bets of the users as well as store their previous bets and information about them in the form of betting tickets. The client will collect the data from the server on request.

Questions to be answered are the following:

How to store the bets of the users as well as their individual information?

How to make the system scalable?

How to make the list of events editable?

How to keep individual account balances?

How to distribute winnings according to match results?

How to enable users to bet on multiple events?

How to make the user authentication?

How to update event results?

How to add/remove funds from users' accounts?

How can the user select different matches from the event pool and place bets?

How can the user check his previous betting activity?

How can the user edit hir/her profile?

How can the admin add or edit matches?

Delimitation

Users can only place bets, see results, and view their betting history.

A web system is not required.

The system does not include credit card payments.

No real events and live results are necessary.

Betting during events is not possible.

Choice of model and method

What Partial problem	**Why** Why study this problem?	**Which** Which models/theories are expected to be used to solve the problem?
How to store information about bets and users?	*One of the main features*	*Database with java.* *UML class modelling.*
How to make the system scalable?	Enabling multiple users to use the program concurrently	Design patterns, Client/Server system

How to make the list of events editable?	Make it possible to edit, add or remove events.	Design patterns. Database
How to keep individual account balances?	Each client has a virtual wallet.	Database with java
How to distribute winnings according to match results?	Main functionality.	Algorithms
How to enable users to bet on multiple events?	Determine the possible win depending on the betting odds.	JavaFX Database
How to make the user authentication?	Secure users' accounts.	JavaFX Database
How to update event results?	To evaluate betting results.	PostgreSQL Java
How to add/remove funds from users' accounts?	To store the outcome of the bets depending on their results and distribute funds accordingly.	PostgreSQL Java
How can the user select different matches from the event pool and place bets?	To be able to make a betting slip.	JavaFX Java
How can the user check his previous betting activity?	To be able to review his/her previous betting activity and the results.	JavaFX Java PostgreSQL
How can the user edit hir/her profile?	To be able to commit profile information changes.	Regex Java PostgreSQL
How can the admin add or edit matches?	To be able to edit already existing events or add completely new ones.	PostgreSQL Regex JavaFX Java

Time schedule

The time scope is estimated at 250 hours. The system is developed using AUP as method. The time schedule is based on a Phase plan as global planning for the system and SCRUM will be used as framework for controlling it and each sprint will contain several of the AUP disciplines.

Inception phase is from 20/4/2016 to 6/5/2016 and final deadline is 3/6/2016. The estimate is that the Elaboration phase is from 7/5/2016 to approximately 13/5/2016 when the Construction phase starts. The Construction phase is estimated to end 31/5/2016 with the

last phase starting, the Transition phase.

References and expected sources

Anon n.d. *Java RMI Release Notes*. Available at: <http://docs.oracle.com/javase/1.5.0/docs/guide/rmi/relnotes.html> [Accessed 21 Apr. 2016].

Wollrath, A., Riggs, R. and Waldo, J., 1996. A Distributed Object Model for the Java System. *Coots*.

Ambler, S., 2002. *Agile Modeling: Effective Practices for EXtreme Programming and the Unified Process*. [online] J. Wiley. Available at: <http://www.ambysoft.com/books/agileModeling.html> [Accessed 27 Apr. 2016].

(Ambler, 2002)

Ambler, S. and Constantine, L., 2002. *The Unified Process Transition and Production Phases*. [online] CMP Books. Available at: <http://www.ambysoft.com/books/transitionProductionPhase.html> [Accessed 27 Apr. 2016].

Ambler, S., 2002. *Agile Modeling: Effective Practices for EXtreme Programming and the Unified Process*. [online] J. Wiley. Available at: <http://www.ambysoft.com/books/agileModeling.html> [Accessed 27 Apr. 2016].

(Ambler, 2002)

(Ambler and Constantine, 2002)

(Generic Methods (The Java™ Tutorials > Bonus > Generics), 2016)

Powell, G., 2006. *Chapter 8: Building Fast-Performing Database Models*. *Beginning Database Design ISBN 978-0-7645-7490-0*. Available at: <http://searchsecurity.techtarget.com/generic/0,295582,sid87_gci1184450,00.html> [Accessed 27 Apr. 2016].

Powell, G., 2006. *Chapter 8: Building Fast-Performing Database Models*. *Beginning Database Design ISBN 978-0-7645-7490-0*. Available at: <http://searchsecurity.techtarget.com/generic/0,295582,sid87_gci1184450,00.html> [Accessed 27 Apr. 2016].