CrossWord.NET Crossword Generator Service

499637 - İlker Küçükil
499649 - Gökhan Örün

Supervisor : Yrd. Doç. Dr. Feza Buzluca

INDEX

- What did we aim to do ?
- Why ?
- CrossWord.NET Architecture
- Conclusions and Suggestions

What did we aim to do ?

- A crossword generator service which
  - can be accessed from many platforms
    - Browsers
    - Smart Devices
    - Windows applications
  - offers individual crosswords to users
    - Size
    - Category (History, Music, Chemistry, Geography, etc. or General)
    - Style (cryptic, quick, freeform, and coded, etc.)
    - Level
    - Everytime different questions
    - User statistic announced to user

What did we aim to do ?

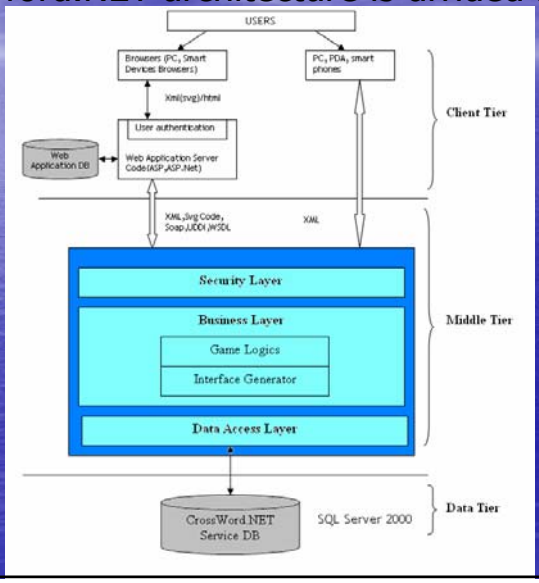
- have dynamically shaped database
  - Users can add questions
  - Questions are graded dynamically
  - Undesired questions removed via users statistics
- Can be marketable to
  - Web sites (educational, personal, commercial, newspaper)
  - Entertaintment based applications (GSM operators)
  - Smart device users
- Can be played via speech

Why ?

- Currently used crosswords
  - Daily limited number of crosswords
  - No opportunity to choose category
  - No smart device applications
  - No user-added questions
  - No feedback of users' knowledge level
  - Prepared by specialists with auxiliary programs

CrossWord.NET Architecture

- Crossword.NET architecture is divided into 3 tiers

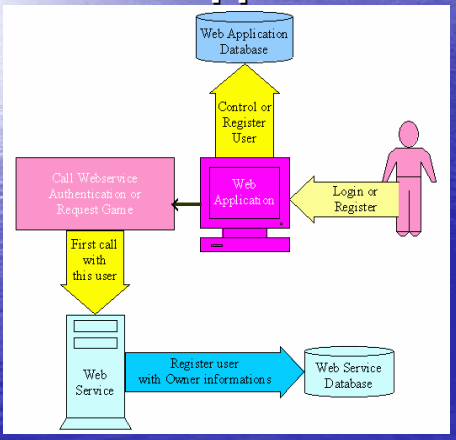


Client Tier

- How client applications interact with the service – WSDL, SOAP
  - WSDL document define web service
    - where a service lives
    - the input parameters
    - the outputs
    - the preferred protocol
  - information travels in XML format
  - this xml document is embedded in body tags of SOAP envelop
  - client applications send their requests
  - service sends response in SOAP envelop

Client Tier

- **Web Application**



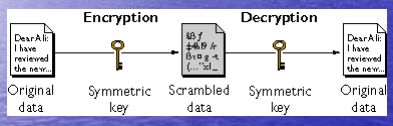
- Given an owner_id and passw to use our web service
- It's users can access crossword with their usernames
- It's users register web service db automatically at first login
- It's users can add question
- It's users can see their order

Client Tier

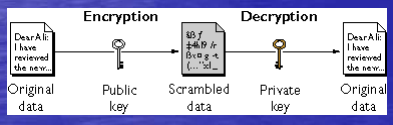
- **Smart Device Application**
  - Smart device owners need to download application assembly at first time
  - We give users username, owner id and password to use smart device application before download
  - Users can see their game results (point, correct and wrong answers, etc.)
  - Can see their order

Middle Tier

- **Security Layer**
  - Symmetric-key encryption method



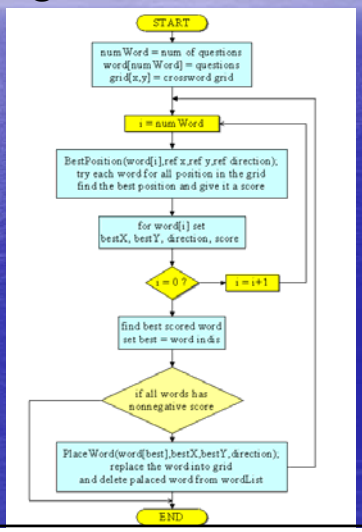
  - Public-key encryption method



  - After user first logged in, we encrypt the user name and id with our public key and send it to user
  - When user requests something, he must send this encrypted data
  - we decrypt this value with our private key to control that user is authenticated or not

Business Layer

- Game Logic



Business Layer

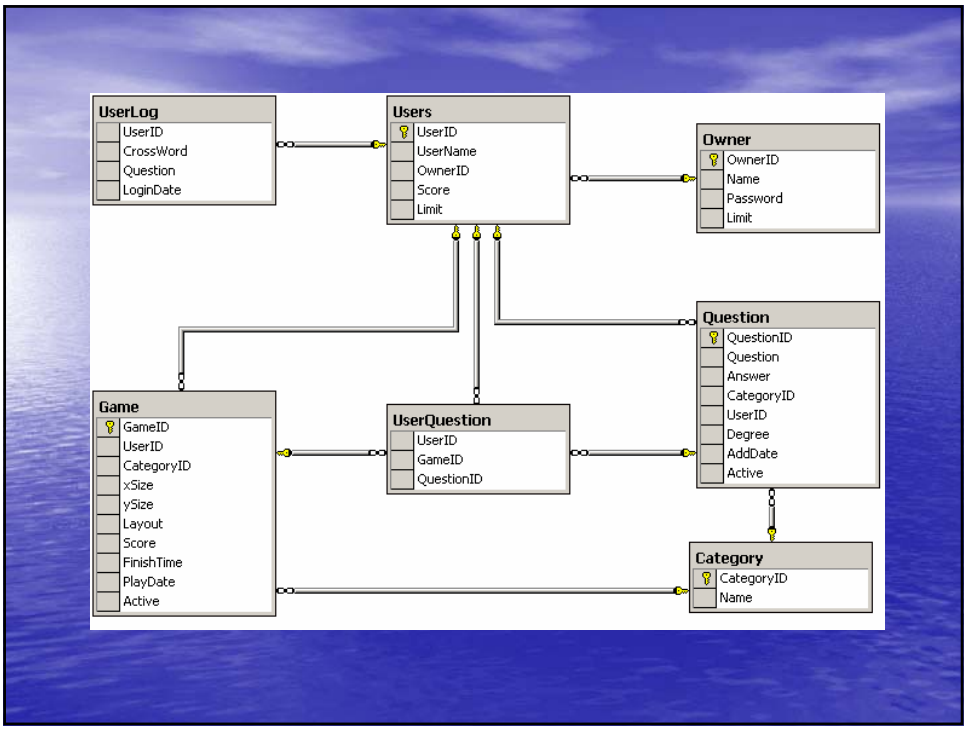
- Interface Generator
  - Game interface will be generated by **SVG** (Scalable Vector Graphics)
    - A w3c standart like wsdl and xml
    - XML based code
    - Can be added in soap envelop
    - Can be generated dynamic easily
    - Faster than applets and flash

Business Layer

- **Data Access Layer**
  - Stored Procedure calls
    - Upper layers can access database through this layer
  - Stored Procedures
    - Add Question
    - User Login
    - Get Category
    - Get top users
    - etc.

Data Tier

- **Tables**
  - **Owner Table**
  - **Users Table**
  - **Category Table**
  - **Question Table**
  - **UserQuestion Table**
  - **Game Table**
  - **UserLog Table**



Data Tier

- **Stored Procedures**
  - Adv. Of using stored procedures
    - Ease of maintenance
    - Fast app. - database interaction
    - Network bandwith conservation
    - Access to database objects that is both secure and uniform
    - Consistent, safe data modification
    - Execution plan retention and reuse
    - Sharing of application logic between applications

Data Tier

- **Sample Stored Procedure**
  - This example stored procedure adds a question into Question table

```
CREATE PROCEDURE [dbo].[AddQuestion]
    (@Question          nvarchar(250),
    @Answer             nvarchar(25),
    @CategoryID         int,
    @UserID             int,
    @Degree             int,
    @AddDate            datetime)
    AS
    INSERT INTO [dbo].[Question]
    ([Question],
    [Answer],
    [CategoryID],
    [UserID],
    [Degree],
    [AddDate])
    VALUES
        (@Question,
    @Answer,
    @CategoryID,
    @UserID,
    @Degree,
    @AddDate)
GO
```

Conclusions and Suggestions

- Suggestions
  - New crossword styles can be added
  - Crosswords can be playable via speech
  - better security techniques should be used to provide service and user security

Conslusions and Suggestions

- Conslusions
  - We learned how to devolop layered structured applications
  - and service-oriented architecture
  - We learned svg (scalable vector graphics)
  - We develop our knowledge of object-oriented programming techniques

Your Questions ?