Casino High Level Design

By Avigail Ben-Tovim and Karina Barenbaum

Contents

[ERD 2](#_Toc5657108)

[Isolation level 3](#_Toc5657109)

[Datatypes 3](#_Toc5657110)

[Tables 3](#_Toc5657111)

[Assumptions 4](#_Toc5657112)

[Procedures 6](#_Toc5657113)

[Maintenance 11](#_Toc5657114)

# ERD



Isolation level

snapshot

# Datatypes

* Username and Password are both required to be 10 characters and so are both each defined as nvarchar(10).
* FirstName and LastName are both required to be 20 characters and so are both each defined as nvarchar(20).
* Address and Email are both required to be 100 characters and so are both each defined as nvarchar(100).
* Country is defined as nvarchar(15).
* Gender is defined as nchar(1).
* BirthDate is defined as datetime.

# Tables

|  |  |
| --- | --- |
| **Table Name** | **Explanation** |
| reference.utbl\_GitPasswords | The known passwords taken from git |
| reference.utbl\_Gender | Holds the drop down values ‘F’, ‘M’ |
| reference.utbl\_Country | Holds the drop down values for ‘England, ‘France’, ‘Israel’, ’USA’, ‘Ireland’ |
| reference.utbl\_SymbolTable | Holds 6 different symbols for the slot machine game |
| admin.utbl\_players temporal table | Player’s details |
| admin.utbl\_transactions | The player’s transactions which can be: ‘bet’/’withdrawal’/’deposit’/’win’/’loss’ |
| admin.utbl\_CompanyDefinitions | Company Definitions |
| admin.utbl\_CreditCard | Credit Card information of the registered Player |
| Security.utbl\_ transactions\_Audits | All data changes from inserts, updates and deletes from any table are documented here |
| games.utbl\_cardtable | Holds 4 sets of 13 consecutive numbers representing cards for the blackjack game |
| games.utbl\_Games temporal table | Information on game transactions |
| Security.CasinoManagers | The casino game managers |

# Assumptions

Most of the DB activities take place at night and in the early morning. This is why most of the maintenance tasks are done from 5am onwards.

The screens below are not supported in the code but assumed to be used to receive the player’s inputs.

Welcome GUI Screen

The first screen the player sees and has the option if to login or register.

Registration GUI Screen

For Registration there will be a GUI screen. The data for Country and Gender will come from the reference tables reference.utbl\_Country and reference.utbl\_Gender and the UserName and Password will be free text up to 10 characters, firstName and LastName will be free text up to 20 characters, Address and Email will be free text up to 100 characters and entering the BirthDate field will bring the player to a pop up date picker screen.

Address and Gender fields will not be required.

Login GUI Screen

For Registration there will be a GUI screen consisting of username and password.

Support GUI Screen

Used by Support to create and email a new random password for the inputted username. Sends output the username, emailPassword, emailusername, emailFromAddress

Lobby GUI Screen

Shows the player her current bankroll and three options to choose from ‘game ground’, ‘cashier’ or ‘administration office’ and sends as output the chosen action and username to the usp\_lobby stored procedure.

Cashier GUI Screen

Called from the Lobby GUI Screen when the player choses the ‘cashier’ action.

Gives the player the option to either deposit or withdraw money.

Money Withdrawal GUI Screen

Player enters here their requested withdrawal amount and shipping address.

Money Deposit GUI Screen

Player enters credit card details and requested amount for deposit.

Administration Office GUI Screen

Called from the Lobby GUI Screen when the player choses the ‘administration office’ action.

Gives the player the option to either change her password or personal details, not including password or username and sends output of choice and username.

Personal Details Change GUI Screen

Called from the Administration Office GUI Screen when the player choses the ‘change personal details’ option.

The screen is the same as the Registration GUI Screen but just with the fields UserName and Password disabled. Sends as output the personal details to be changed.

Password Change GUI Screen

Called from the Administration Office GUI Screen when the player choses the ‘change password’ option.

The screen is the same as the Registration GUI Screen but just with the Password field enabled. Sends as output the requested password.

Game Ground GUI Screen

Called from the Lobby GUI Screen when the player choses the ‘game ground’ option.

Shows the player the games options, either blackjack or slot machine and sends the username and action as output.

BlackJack Game Form GUI Screen

Called from the Game Ground GUI Screen when the player chooses to play BlackJack. Receives from the player as input the ammount of cards chosen, decided bet amount and username and sends them as output to the usp\_blackjack procedure. When the player finishes the game, they have the option to return to this screen for another round.

Slot Machine Game Form GUI Screen

Called from the Game Ground GUI Screen when the player chooses to play SlotMachine. Sends as output the username to the usp\_slotMachine procedure. When the player finishes the game they have the option to return to this screen for another round.

Company Management GUI Screen

For Company Management to redefine company constant definitions. Gives the option of update, delete and insert from the utbl\_CompanyDefinitions table and sends the option to the procedure

# Procedures

**usp\_welcome procedure**

The player reaches the Welcome GUI Screen and has two choices. To register as a new player or to login with an existing username and password.

‘Register’ sends the player to the Registration GUI Screen and the player will be asked to fill in her details.

When the player has entered all required information, the information will be sent to the usp\_validate\_playerDetailsprocedure with a flag ‘N’ notifying that this is for registration.

‘Login’ sends the player to the usp\_Login procedure

**usp\_validate\_playerDetails**

This procedure is used for validating player details for registration and reused to validate player details for change of personal details that are not password or username.

The procedure receives the inputted data together with the flag if this is for new registration or change.

In the instance of a new registration:

Username Validation

The inputted username will be validated to check if it already exists with an existing player in capital letters or small letters. If it already exists, a random number will be added to the username. This new username will be too validated to check if this also exists with an existing player. If not, the player will be given the username as an option or to choose a new username altogether and will be sent back to ‘Registration GUI Screen’ to reenter her details.

Password Validation

The inputted password will be validated with the udf\_PasswordSyntaxValid and udf\_PasswordExtValid functions.

If the Password passes this validation, it will then be checked against the reference.utbl\_GitPasswords table to check if it exists as a known password.

If the validation fails for any of the above, the player will receive a message stating the rules for a new password and will be sent back to the ‘Registration GUI Screen’.

Birthdate Validation

The inputted birthdate will be validated to check if the player is over 18. On the event of the player being younger than 18, the player will receive a message stating that she cannot participate under the age of 18 and will be sent out the game.

Email Validation

Used for validating the inputted email address for a new player and also for personal details change request.

Validates that the Email is not null, includes ‘@’ and doesn’t already exist with another player.

If the validation fails for any of the above, the player will receive a message stating the rules for a new password and will be sent back to the ‘Registration GUI Screen’.

New Player Addition

On the event of a new player registration and all details have passed validation, the new players details with additional values of: NumFails will be set to 0, IsBlocked will be set to ‘N’, LoginTime will be set to the current date and time, IsConnected will be set to ‘N’ will all be inserted to the admin.utbl\_players table and the admin.utbl\_PlayerBankroll will receive a new entry for the new player with a welcome bonus as configured for the 'welcomeBonus' value in the admin.utbl\_CompanyDefinitions table and will be sent to the usp\_Login procedure for login

Personal Details Change

In the event of personal details change and all details have passed validation, the players details will be updated in the admin.utbl\_players table and will be sent to the Login GUI Screen to login.

**usp\_Login**

This procedure is used for validating player details for login.

The procedure receives the inputted username and password from the Login GUI Screen and checks if the username exists. If the user exists, the password is validated. If it is the wrong password, NumFails will be incremented by 1, a check will be made to see if the player is blocked by checking how many times the player has failed to login against the 'logonTimes' value as configured in the admin.utbl\_CompanyDefinitions table. If the player has not reached the 'logonTimes' value, she will receive notification and is sent back to the Login GUI Screen to try again. Otherwise, the player is blocked, will be disconnected and will be told to contact support in order to get unblocked. On calling support, support will send the username to the usp**\_**autoPasswordChangeprocedure via the Support GUI Screen.

If the player password entered matches the existing password, the user will be connected, IsConnected will be set to ‘Y’ , NumFails will be updated to 0 , loginTime will be updated to the current date and time. The current bankroll will be calculated and the player will be sent to the casino lobby GUI Screen to view her bankroll and either choose to play a game, go to cashier or admin. With her choice, the player will be sent to the stored procedure usp\_lobby.

**usp\_autoPasswordChange**

This procedure allows for support to change the users password by providing a random password. It receives its inputted values of username, emailPassword, emailusername and emailFromAddress from the Support GUI Screen and validates the random password to ensure it was not used in the past by the user, this check is made by calling the function udf\_IsPasswordInPast.

Before reaching this stored procedure, an email account and profile must be created. This is done in the usp\_createEmailAccountProfile procedure.

**usp\_createEmailAccountProfile**

Deals with the prerequisites that must be done for sending and receiving the password notification email from administration. Should be run only once at installation:

* Creates a Database Mail account
* Create a Database Mail profile
* Add the account to the profile

Grant access to the profile to all users in the msdb database

**udf\_IsPasswordInPast**

This function receives the username and password to check if the requested password was used by the same user in the past. In addition, also checks if the password is a known password from git.

**udf\_PasswordSyntaxValid**

Receives as input the password and does a series of validations:

* Its length is at least 5 characters long.
* Includes at least 1 digit.
* Includes at least 1 capital letter.
* Includes at least 1 small letter.
* Does not include the word ‘password’ in any form.

**udf\_PasswordExtValid**

Receives as input the password and checks it for external validation of per customer needs (exit point).

**usp\_lobby**

This procedure is used to call the screen that corresponds to the action requested in the Lobby GUI Screen, ranging from ‘game ground’, ‘cashier’ or ‘administration office’. Holds the players current bankroll

**usp\_Cashier**

Called from the Cashier GUI Screen, receives players choice of action if ‘deposit’ or ‘withdrawal’ and sends request to appropriate screen.

**usp\_MoneyWithdrawal**

Called from the Money Withdrawal GUI Screen in the case of withdrawal request. Receives as input the username, withdrawal amount and shipping address.

This procedure will add withdrawal transaction to the utbl\_Transactions table so amount is removed from the bankroll and will print the shipping address for the check

**usp\_MoneyDeposit**

Called from the Money Deposit GUI Screen in the case of deposit request. Receives as input the username, deposit amount and credit card details.

The encrypted credit card details are added to the utbl\_CreditCard table for every transaction and will add the deposit transaction to the utbl\_Transactions table.

**usp\_admin**

Called from the Administration Office GUI Screen in the event that the chosen action was ‘administration office’ and receives as input the player’s choice if to change her personal details (not password or username) or password.

If the requested action was ‘PersonalDetailsChange’ the Personal Details Change GUI Screen will be shown. This screen will send the outputs from the player to this procedure as input. The procedure checks which fields have been modified by checking them against the saved fields in the utbl\_players table.

Checks the validation of the newly inputted personal details by reusing the usp\_validate\_playerDetails procedure and sending in the ‘PersonalDetailsChange’ choice. If personal details are valid, the players details in the utbl\_players table will be updated.

If the requested action was ‘passwordChange’ the Password Change GUI Screen will be shown.

Checks the validation of the newly inputted password by dbo.udf\_IsPasswordInPast to check if password was already used by the user in the past or exists in the git passwords table and by dbo.udf\_PasswordSyntaxValid that checks the syntax of the password so it conforms to the casino password rules. If the password passes these validations, the players passsword in the utbl\_players table will be updated.

**usp\_gameGround**

Called from the Game Ground GUI Screen in the event that the chosen action was ‘game ground’.

Receives as input the requested game and username.

**usp\_BlackJack**

Called from the BlackJack Game Form GUI Screen when the player chooses to play BlackJack. Receives the username and chosen number of cards, the inputted the bet amount is validated against the players balance. It the validation passes, the player will be able to start a new round.

The player gets the number of requested cards which are randomly selected from the utbl\_cardtable table. The requested cards are then deleted to ensure they will not be chosen again for this game.

The sum amount of player card values are validated. If more than 21 then the player has lost the game, the utbl\_Games table is updated, the utbl\_transactions table is updated with a ‘loss’ and the player is sent back to the Game Forum GUI Screen.

If the player’s sum of card values is more than the dealer’s sum of card values but the dealer’s card values are less than 21, the dealer takes a random card from the utbl\_cardtable. This card too is deleted to ensure they will not be chosen again for this game. The dealer continues to take a card until either the sum of values exceeds 21 or exceeds the player’s sum value.

If the dealer’s cards are higher than those of the players but lower or equal to 21, the player loses, the utbl\_Games table is updated, the utbl\_transactions table is updated with a ‘loss’ and the player is sent back to the Game Forum GUI Screen.

If the dealer’s cards are higher than 21 the player wins, the utbl\_Games and utbl\_PlayerBankroll tables are updated, the utbl\_transactions table is updated with a ‘win’ and the player is sent back to the Game Forum GUI Screen.

**usp\_SlotMachine**

Receives as input the player username from the Slot Machine Game Form GUI Screen.

Randomly selects 3 symbols from the utbl\_SymbolTable. If the symbols are equal to each other, the player wins the game, utbl\_Games and utbl\_PlayerBankroll tables are updated, the utbl\_transactions table is updated with a ‘win’ and the player is sent back to the Game Forum GUI Screen.

If the symbols are different, the player loses, the utbl\_Games table is updated, the utbl\_transactions table is updated with a ‘loss’ and the player is sent back to the Game Forum GUI Screen.

**usp\_CardTableFiller**

This procedure fills the utbl\_CardTable table with 4 sets of consecutive numbers. This procedure is called before every blackjack game.

**usp\_SymbolTableFiller**

This procedure fills the utbl\_SymbolTable table with 3 sets of 6 unique symbols. This procedure is called upon set up of the database.

**usp\_CompanyDefinitions**

This procedure allows for insert, update or delete of the utbl\_CompanyDefinitions table with values for the company constant definition keys. This procedure is called from the Company Management GUI Screen to manage the table without the help of a DBA.

**usp\_create\_NewPeopleUser**

Procedure to loop through the whole Security.utbl\_CasinoManagers table and create a DB user for each manager to enable Row Level Security on the utbl\_Games table.

**udf\_securitypredicate**

A function called from the SECURITY POLICY GamesPolicyFilter to add the security policy on the utbl\_Games table to allow on the manager of the game to see the appropriate game results.

**usp\_Feedback**

This procedure is called from the Player FeedBack GUI Screen where the player sends their feedback by mail to the casino admin.

**udf\_Bankroll**

Calculates the player bankroll by getting the sum of transactionAmount from the utbl\_transactions table where the transactionTypes are ‘deposit’, ‘withdrawal’, ‘bet’, ‘win’ and ‘bonus’ and calculating: bankroll = deposit - withdrawal - bet + win + bonus

**usp\_insertTransactions**

Inserts a new transaction to the utbl\_transactions table on the event of ‘Bonus’, ‘Bet’, ‘Withdrawal’, ‘Loss’, ‘Win’

**udf\_updateGame**

This function is called whenever a game has been played and updates the utbl\_Games table on win, loss and round number.

**usp\_CompanyDefinitions**

Allows for company management to insert, update or delete values and keys in the utbl\_CompanyDefinitions table without the need of a DBA.

**usp\_logout**

Allows for the player to initiate logout from the system

# Maintenance

* Cleaning the Admin.utbl\_ApplicationLog table is scheduled for 6:00 AM daily
* DB Statistics updates are scheduled for 7:00 AM daily
* DB Integrity checks are scheduled for 8:00 AM daily
* Index rebuild are scheduled for 12:00 AM (midnight) daily
* Full backup will be done daily at 5:00 am
* Differential backup will be done hourly
* Transaction log backup will be done every minute

# DBandTablesCreation

# This file creates the DB, schemas and all tables.

The DB is created with a flexible filename for the log and data files.

After table creation the master key for certificate and encryption is created on the utbl\_creditCard table is created.

# Insert Scripts

Holds the insert scripts for the reference tables, utbl\_CasinoManagers and utbl\_CompanyDefinitions tables.

# Installation Scripts

This file includes all procedures and set up scripts for the application and jobs needed as a prerequisite before the running of the application procedures.

* Set the DB to read\_commited\_snapshot and set transaction isolation level to level snapshot to enable for maximum concurrency.
* Creation of profile and account for email sending.
* Set Row Level Security on the utbl\_Games table for the game managers from the utbl\_CasinoManagers table. The usp\_create\_ManagerUser procedure users from the table. The creation of security predicate udf\_securitypredicate and security policy GamesPolicyFilter.
* usp\_betBonus procedure for the **Bet Bonus for Casino job** to check if a player has made a bet larger than the minBetAmntForBonus value in the utbl\_CompanyDefinitions table. If so, the player will receive a bonus of the value of betBonus from the utbl\_CompanyDefinitions table.
* usp\_connectionsCheck procedure for the **Number of Connections > adminAmount** checks how many players are connected. If the amount is higher than the value for the numConnectionsAlert value in the utbl\_CompanyDefinitions table. If so, the administartor will receive a mail to the value of adminMailAddress from the utbl\_CompanyDefinitions table.
* usp\_noNewLogins procedure for the **No logins in last 10 mins job** checks since when the last player login time was. If the amount of minutes is higher or equal to the value for the numMinsNoLoginsAlert value in the utbl\_CompanyDefinitions table. If so, the administartor will receive a mail to the value of adminMailAddress from the utbl\_CompanyDefinitions table.
* Enables server audit for the admin schema.
* usp\_createFullBackup procedure for the **fullBackupCasino Job** creates a full backup with name, destination and description taken from the values in utbl\_CompanyDefinitions table.
* usp\_createDiffBackup procedure for the **diffBackupCasino Job** creates a differential backup with name, destination and description taken from the values in utbl\_CompanyDefinitions table.
* usp\_createLogBackup procedure for the **logBackupCasino Job** creates a transaction log backup with name, destination and description taken from the values in utbl\_CompanyDefinitions table.
* Creation of TestAdmin\_Casino login to test masking on utbl\_Players table on lastname, firstname and email and the masking of the columns.
* Creation of a linked server for oracle.
* Creation of usp\_playerActivity **playerActivityCheck job** procedure for the for checking if players were inactive in more than the amount stated in the value of @adminActivePlayer in the utbl\_CompanyDefinitions table.

Alerts

No Full Backup in 24 Hours - gets raised when full backup fails \isn’t performed in 24 hours.