Sela

TalkBack

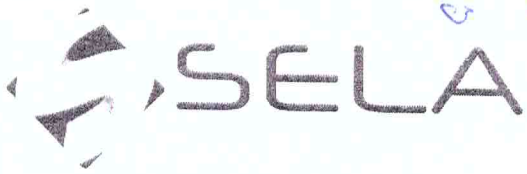
Chat and Backgammon

Messaging program

SRS

Document Release Notes

Ver	Date	Author	Description	Approved by	Approval date
1.0	Dec 23, 2008	Tzvi Chernofsky	Initial Draft	Roy Rachmany	
1.1	Jun 08, 2009	Roy Rachmany	Final Draft	Roy Rachmany	
1.2	July 15, 2009	Roy Rachmany	New Final Draft	Ofer Feldman	



2. Functional Requirements

TalkBack is a messaging system which provides users with the ability to chat with each other and/or play backgammon against each other. The system is comprised of clients who connect to a central application server, this allows them to see who is online, and to choose whom to talk to or play backgammon against.

2.1. *Presentation layer*

This section specifies the functionality of the GUI layer of the system by describing the functionality of the client. Detailed GUI layout is specified below in a separate section.

2.1.1. *Taskbar icon*

Upon launching the application, the client window will open, at the same time the icon of the program shall appear in the lower right-hand corner of the taskbar of the operating system (most commonly Microsoft Windows), as is with other messaging programs.

The appearance of the icon shall be to show the user is offline, until the client connects to the server. Then the icon shall change to any appearance as defined in section 7.1.

Mouse left-clicks on the icon should have no functionality. Right clicks shall have the functionality as detailed in section 7.1.

2.1.2. *Application main window*

The main window of the application should have the title and menu button as detailed in section 7.2 below. Upon launching the application, this window will have the format of the "sign-in screen", and upon connecting to the server – change to the "contact screen". It should remain this way until the user signs out, and then it should change back to the sign-in screen.

The title and the menu functions should stay the same on both screens. (Menu functions are detailed in section 7.2 below.)

2.1.2.1. *Sign-in screen*

The header should display a welcome message. Below this will be the sign-in area.

The sign in area includes fields for name and password and a button for submitting.

First time activation: When launching the application for the first time, the user must choose his sign-in name and password, enter them and connect to the service. Once the connection is completed, this screen will change to the client's main screen, which will be called "contact screen".

If the connection fails, the user should be presented with an appropriate message.

The application should remember the sign-in name for the next time the user signs in, and suggest it to him whenever the client is launched.

2.1.2.2. Contact screen

Here the user sees his list of contacts and their statuses – are they currently connected or not. In this version of the application, whoever signs-up to the service and successfully registers on the server is automatically included in the other users' contact lists.

When the user sign-up for the first time, the client launches this screen and shows a "connecting..." message, while it receives the list of contacts from the server. Now the list of contacts should appear on the user's contact list.
When an existing user signs-in, the screen should display the entire list of contacts, all in offline mode, and a "connecting..." message. When connection is complete, the status of those contacts that are online should change, and they should appear in the "online contact" section.

The user can choose to communicate with any online contact in one of the following ways: Chat or play Backgammon.

- Chat – the user double-clicks the contact name or highlights the contact and then presses the chat button. When doing so, a chat window will open separately, in which the user can chat with the contact.

- Play Backgammon – if the user highlights the contact name and presses the "game" button, a popup window with a "wait for connection..." message should appear, while waiting for the contact to accept the match proposal.
  If the contact confirms, a separate "game screen" will be launched, in which the game shall be played.
  In event the contact declines the request to play, the client of the initiating user shall display a "sorry, you request has been refused" message to the user.

In future version:

In event that no contact has been highlighted, then the server will attempt to find a random contact to play against (for this to happen, a different user must also be looking for a random player to play with).

- *If a user is found within the next 15 seconds, the application should launch the "game screen" and start the game against this contact.*

- *If no user is found to within the allotted time, the program will launch the "game screen" and notify the user he will be playing against a computerized opponent. The application will then make the moves for the opponent, according to an algorithm programmed into it.*

2.1.3. Chat screen

The screen should be divided into two panes: the upper part has the chat history, and the lower part is the editing area, where the user composes and sends his messages.

This screen is launched in two ways:

- The user selects a contact to chat with. In this case the window will be empty when launched.

- A contact initiates a chat with the user. In this case the window will appear with the contents of the contact's message in the history pane. The program should also emit some kind of sound to notify the user of the incoming message.

After the user composes a message and sends it, the message will disappear from the lower part and appear in the upper part of the screen.

If the message wasn't received successfully by the recipient, the application should show a "Your message wasn't sent successfully" warning beneath the sent message in the chat history pane.

All messages should appear - in the order they were sent - in the history pane.

Play Backgammon – if the user presses the "game" button in this screen, a popup window will be displayed, while waiting for the contact to confirm he's interested in playing. If the contact confirms, the "game screen" will be launched. If not, the user will be displayed with a message, and a button to close this popup.
Obviously, the only possibility here is to play against the contact the user is chatting with, and not a random contact.

2.1.4. *Game screen*

This screen is launched only after the contact has confirmed he wishes to play, or when the client will simulate a computerized player to play against the user.

This screen should show a standard backgammon board, as detailed below (in section 7.5). The user should see his name near the lower side of the board, and the contact's name near the upper side. Buttons are available for rolling dice or resigning the game. There should also be a button for undoing the last move.

The rules of backgammon are known, and a tutorial can be found in the menu of the main window, under: Help > How to play Backgammon.

The program should offer each side to roll one die to see who begins. The one who gets the higher number goes first (in case of a tie, each player rolls again).

All other turns begin with the conclusion of the other player's turn. From the moment each player's turn begins, he has 2 minutes to roll the dice and make his moves. If he doesn't move within this time, he forfeits the game and the other player is declared winner. In event this happens, the program should display a popup window, notifying the user he hasn't made a move in time and he lost the game.

The controls won't be active during the other player's turn. Only when the opponent completed his turn do the controls become active again.

When the player has rolled the dice, the program will display the dice and each number beneath them. As each move is used, the die representing it will disappear. (In case of doubles, which gives the player four moves - two times each die, then each die should become dimmer after the first move is made, and disappear after the second move.)

The player controls the playing pieces of the color assigned to him by the computer. The color is that of the five pieces located on the left-most column of his "home" area (the lower right-hand side of the

board). The player cannot move the opponent's pieces. (When moving one of his pieces to "knock off" an opposing piece, it will automatically be placed on the bar in the middle of the board.)

The player makes a move by clicking on a game piece to "pick it up", which will change the mouse pointer to be that piece. The player then clicks on the column he wishes to move it to, in order to "put it down" at that location. When removing pieces from the board, the user should click the screen to the right of his "home" area to drop the piece there. If the piece is successfully put down, the mouse pointer should return to normal.

If the user attempts an invalid move, there are two possibilities:

- If the user may not pick up the desired piece at the moment, then clicking the mouse on it will do nothing.
- If the user may pick up the piece but may not put it down at the desired location, then clicking the mouse at that location will cause a "beep" to be emitted, and the piece will remain "picked up."

A player can undo a move provided he didn't complete his turn. This means that he can undo his first move (or first, second and third in case of doubles). If the user completed his turn, the undo button shouldn't work, as should none of the other controls, as play now goes over to the other player.

If the player decides to resign the game, he may, on his turn, press the "resign" button. The program will open a pop-up window and ask the user to confirm his resignation. If he confirms, the other player will be notified he is the winner, and the game will end. If the player regrets his resignation and cancels, the game will continue as usual.

The first player to remove all his pieces from the board is the winner. When the game is over, the program should show a popup window with the winner's name and the number of points he won (1 for regular win, 2 for gammon, 3 for backgammon).

When a game ends, for one of these 3 reasons:

a) One player has removed all his pieces from the board.
b) A player resigned the game to the other player
c) A player didn't play his turn for two minutes

The program should display a popup to both players and notify the winner of his win, and the loser of his loss. When the user(s) close(s) this (these) popup window(s), the game screen(s) of the user(s) should close as well.

In future version:
In case of a computerized opponent, the client uses the game logic programmed into it to make appropriate moves in the game, and does not involve the server at all.

2.2. *Application layer*

2.2.1. *Client requests and reactions to responses*

The TalkBack client sends the server requests when the user wishes to do any action against the server, such as sign on or sign off, send messages to contacts or play backgammon against them.

The client receives responses from the server and must know how to act accordingly.

2.2.1.1. Sign-up request

When the user sign-in for the first time, the client launches this screen and at the same time sends a "create user" request to the server. The response from the server should include a list of all users who have already signed up to the service, and they should appear on the user's contact list.

2.2.1.2. Sign-in / sign-out request

When an existing user signs-in, the client should send a "sign on" request to the server. Now the client waits for the response. All contacts should appear as offline until the response is received.
The response should include all usernames of online contacts. The application should then show all these users as online.

When signing off from the contact screen, the client should send the server a "sign off" request.

2.2.1.3. Game request

If the user highlights the contact name and presses the "game" button, a "game request" including the contact's username will be sent to the server, to push a request message to that contact.
The user should be presented with a popup window with a "wait for connection..." message, while the client waits for the "game response" from the server.
If the contact confirms, a separate "game screen" will be launched, in which the game shall be played.
In event the contact declines the request to play, the client of the initiating user shall display a "sorry, you request has been refused" message to the user.

In future version:

In event that no contact has been highlighted, the following should happen:

The client sends a "game request" with no username specified in the request. The server will try to find a user to play against within 15 seconds.

- If a user is found within this time, the client will receive the "game response" with the username of the contact found. The client should then launch the "game screen" and start the game.
- If no user is found to within the allotted time, the "game response" from the server won't include any username. The client understands from this that no opponent has been found, and it must simulate a computerized player to play against the user. Following this the "game screen" will be launched.

If the user initiates a game from within a chat window, the client will send the name of that contact in the request. There is no possibility of playing against a random contact here.

2.2.1.4. Message request

After composing a message and pressing "send" or the "enter" key, the client should send a "message request" to the server, and wait for a "message response".

The response should include an indication whether the message was received successfully by the contact. If it wasn't, the client should notify the user with an alert displayed in the chat history.

2.2.1.5. *In-game requests*

If the user is playing against a real opponent, the client will send each move to the server and wait for a confirmation that the move has been displayed on the opponent's client. The client will then wait for a message from the server which includes the opponent's roll and for another message which includes the moves made. Then the client sends the server a message to confirm it displayed the moves received.

This continues until the game ends. Then the client should display the winner and not send anymore requests to the server concerning this game.

2.2.2. *Push messages from the server*

The client sometimes will receive push messages from the server, without initiating a request. This happens on several occasions: When a contact connects or disconnects, initiates a chat with this user or requests to play a game of backgammon against him.

- The client will receive an "online message" from the server when a contact connects. The client should then mark the contact as "online" in the main window ("contact screen") as described in section 7.3.
- The client will receive an "offline message" from the server when a contact disconnects. The client should then mark the contact as "offline" in the main window ("contact screen") as described in section 7.3.
- If the client receives a message which includes a contact's username and message content, it should understand that someone initiated a chat with the user. The client checks to see if a chat window with this user is already open.
    - If not, the client shall automatically launch the chat window and display the message received from the server.
    - If a chat window with this specific user is already open, the client should display the message in the history pane of the window, below the last message which is already there.

  In either case, the client should also emit some kind of sound to notify the user of the incoming message. Then it sends the server a confirmation that it displayed the received message.
- If the client receives a game request from another user, it should open a popup window and display a message to the user regarding the game invitation. The user can accept or decline.
    - If the user accepts, an "acceptance message" is sent to the server. Then the client waits for the server to send a "launch game" response. Following this the client should launch the game screen.

- If the user declines, the client should close the popup window and send the server a "refusal message".

2.2.3. Server requirements

2.2.3.1. Overview

The TalkBack server's purpose is to provide connections between clients of the system. The server receives requests from the clients and sends responses accordingly, provided that the request was in the correct format. The server also "pushes" messages to a client, when a different client sends requests to communicate with this client.

2.2.3.2. Sign on / sign off responses

When the server receives a "sign on" request from any client, it will do the following:

- Generate a "session code" for this client and send it back in the response. This session code will be used by the client for the rest of this session, and must be included in every request sent to the server. (That is how the server knows what client is sending it the request.)
- Place this client in the list (also known as a "map") of "online clients", and send back in the response the list of all other users who are online.
- Send all other online clients a message, notifying them that this user is online.

When the server receives a "sign off" request, it should do the following:

- Discard the session code and remove the user from the "map" of online clients.
- Send all online clients a message, notifying them that this user is offline.

2.2.3.3. Chat response

When receiving a "chat request" from a client, the server will check who the message is directed to, and see if that user is online. If so, the server will push the message to that client, causing it to show the message.

The server should wait for a confirmation from the receiving client, and then send the initiating client a response. The response should tell the client whether the message was received successfully at its destination. A negative response should be given in one of these cases:

- If the recipient is offline now.
- If the message couldn't be sent successfully to the user, for any other reason.

2.2.3.4. Backgammon request

2.2.3.4.1. Specific contact request

When the server receives a request from a client for a backgammon game, it must check if the request includes a username (against whom the user wants to play).

If so, the server checks if this user is online, meaning his client is connected to the server.

- If he's connected, the server pushes a message to the specified client, telling it to display a message to the user, to accept the game challenge. If the user accepts, his client sends an

"acceptance message" to the server. The server then checks that the initiating user hasn't cancelled the request, and if so, it then sends both clients a "launch game" response, which causes the game screen to be launched on both of them. Now the server will handle the game moves between the clients.
If the user declines, the server should receive a "refusal message" from his client, and send the requesting client a response that the game request has been declined.

- If the user is not connected, the server should send an error response to the client.

2.2.3.4.2. *Random contact (or computer) request (In future version)*

If the request does not include a username, then the server must search for a contact to play against or offer the requesting user to play against a computerized player. This is done in the following way:

- The server will place this user in a "pending game" list, and wait for 15 seconds. If within the allotted time another user gets added to this list, the server shall immediately initiate a game between these users, and launch the "game screen" on each of their clients. Then the server shall clear the "pending game" list.
- If there is a user already in the "pending game" list, the server will immediately initiate a game between these two users, and clear the pending list.
- If no user is found to match to the pending user within the allotted time, the server will send a response to the client that no user has been found, and now the client must launch the game and play the moves for the computerized opponent. Now the server must clear the "pending users" list.

2.3. *Reports*

TBD.

3. System Architecture

3.1. Overview

TBD

3.2. System Architecture Diagram

4. TalkBack Server Requirements

4.1. *Overview*

PF SQL

TalkBack will include an Access DB on which all system data will reside. This section will specify the main DB tables.

Users	
UserName	string
Password	string

5. Non-functional Requirements

5.1. *Data Management*

5.2. *Performance*

Any client activity that requires a response from the application servers will receive the response within no more than 5 seconds for ADSL or cable connections and no more than 10 seconds for dial-up connections.

5.3. *Security*

TalkBack will be based on the WCF architecture which supports security aspects.

6. Technical Implementation Aspects

6.1. System Configuration

This section describes the SW & HW configuration components, which comprise the TalkBack system, including the client application operated by the users and the various servers –application and DB.

6.1.1. Client Platform

The client application serves the users for communicating with other users, by chat and/or playing games against them. The client application can be installed on a machine with minimal HW & SW requirements:

- PC running MS-Windows XP
- CPU – Any Pentium 4
- RAM – at least 256 MB

6.1.2. Application Server Configuration

The application servers will be Windows server servers of Microsoft. System configuration will reside in XML formatted INI files; the configuration describes the deployment descriptors such as: DB location, security definitions, etc.

6.1.3. DB Server

The DB server will run Access.

6.2. Development Platform

The development will be based on .Net architecture. The application layer running on the application servers will be comprised of a set of .Net business components – these components will perform:

- The application logic
- DB interface

7. GUI Requirements Specification

This section specifies the screens layout and GUI requirements of the various GUI objects of each screen.

7.1. *Taskbar icon*

The taskbar icon shall appear like this when online:

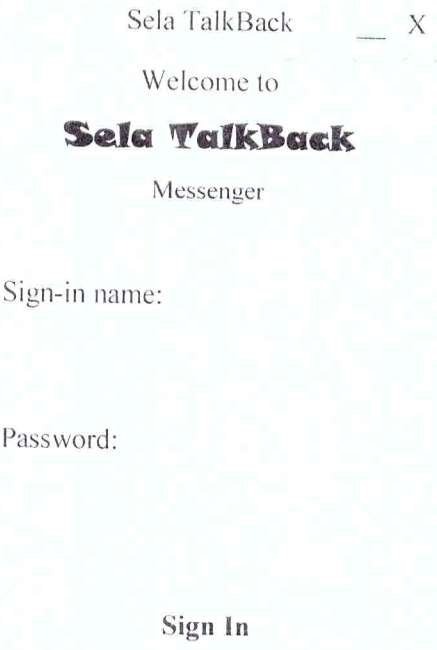
And this way when offline:

When hovering with the mouse over it, the following tooltip should appear: "Sela TalkBack messenger - <Status>," where status is either "online" or "offline" depending on the state of the user's application.

Right clicking the mouse on the icon should open the following menu:

Option	Functionality
Sign-in	Enabled only if the client is currently signed out. When selected, the client will sign in and the icon should change to the online appearance.
Sign-out	Enabled only if the client is currently signed in. When selected, the client will sign out and the icon should change to the offline appearance.
Open	Always enabled. When selected, the application window will appear on the screen.
Exit	Always enabled. When selected, the client will sign-out and the application should exit. The icon should disappear from the taskbar.

7.2. Sign-in screen



The title of this screen is: "Sela TalkBack".

The top of the screen displays the following header: "Welcome to Sela TalkBack Messenger".

Below this is the sign-in area.

#	Object	Title	Type	Values	Functionality

Register

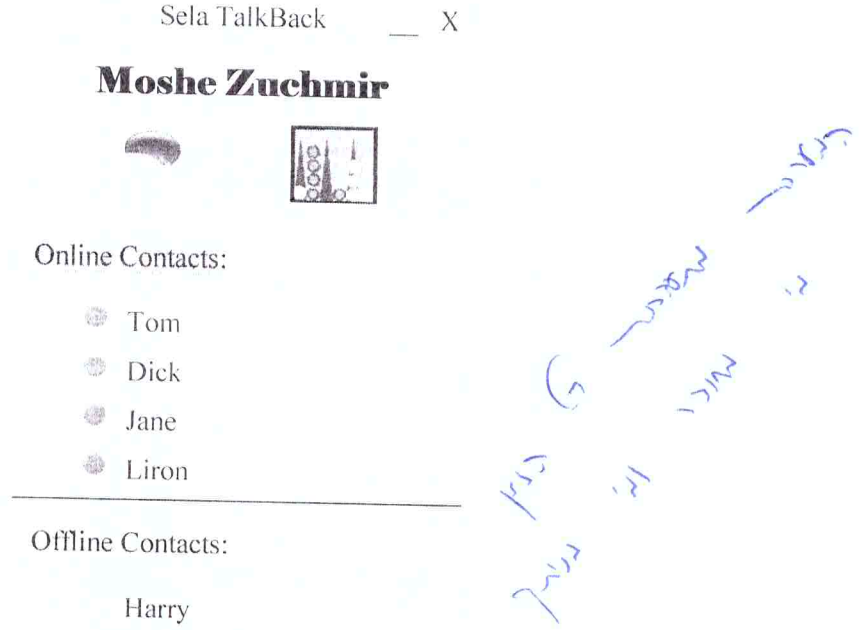
#	Object	Title	Type	Values	Functionality
1	Window header	Sela TalkBack	Typical windows window title	-	The title bar shall have the standard windows minimize and close buttons, as well as a down-pointing arrow for bringing up the menu.
2	Menu button	Downward pointing arrow	Button	–	Always enabled. Upon selection, the menu will pop up under the arrow. (See menu options below.)
3	Sign-in border	Welcome to Sela TalkBack messenger	Border	-	-
4	Sign-in	Sign-in name	Text box	None for a new user. For an existing user, the username may appear highlighted.	Always enabled. Displayed the characters entered by the user.
5	Password	Password	Password box	None	Always enabled. Displays the characters entered as dots.
6	User sign-in	Sign in	Button	-	Enabled only if user entered name and password. Upon selection, the client will attempt to connect to the server. If successful, the screen will change to the "contact screen". If sign-in failed, a popup window will appear with the message: "Cannot connect now. Try again later."

Menu options:

Option	Functionality

Option	Functionality
Connect / disconnect	"Connect" appears if the client is currently signed out. When selected, the client will sign in and the window should change to the "contact screen". "Disconnect" appears if the client is currently signed in. When selected, the client will sign out and the window should change back to the "sign-in screen."
Actions	Enabled only if the user is signed-in. When selected, the actions menu will open to the right of it. It should have these options: **Send Message** – enabled only if a contact is highlighted in the list. Selecting this should be like pressing the chat button in the contact screen. **Play Backgammon** – always enabled. Selecting this should be like pressing the game button in the contact screen.
Help	When selected the help menu will open to the right of it. It should have these options: **How to play** – launches a tutorial on how to play backgammon. **About** – shows info about the program.

7.3. *Contact screen*

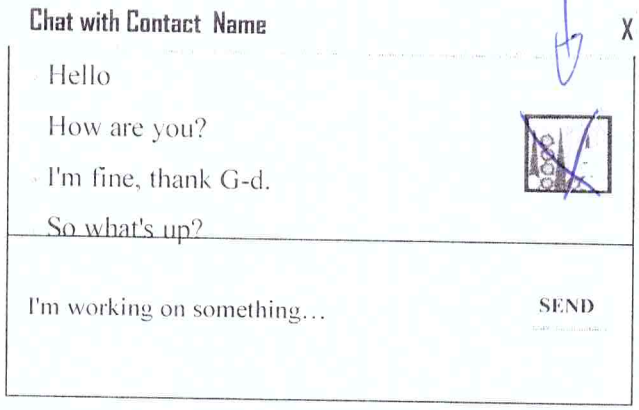


This screen will retain the title of the sign-in screen: "Sela TalkBack".

#	Object	Title	Type	Values	Functionality
1	Window header	Sela TalkBack	Typical windows window title	-	-
2	User border	<Username>	Border	The current username should appear here	-
3	Chat button	Chat icon	Button	-	Enabled only if one of the contact names in the contact lists below are highlighted. Upon selection, the "chat screen" will open, initiating a chat with the selected contact.

#	Object	Title	Type	Values	Functionality
4	Game button	Backgammon icon	Button	-	Always enabled. Upon selection, the application will attempt to start a game of backgammon against a selected contact (if a contact's name is highlighted) or against a computerized player (if no contact is highlighted).
5	Online Contacts area	Online contacts	List	Displays those contacts that are online with a green circle next to each name.	A contact name can be clicked. When clicked, the name will become highlighted and the chat button will become enabled.
6	Offline Contacts area	Offline contacts	List	Displays those contacts that are offline with an orange circle next to each name.	-

7.4. Chat screen

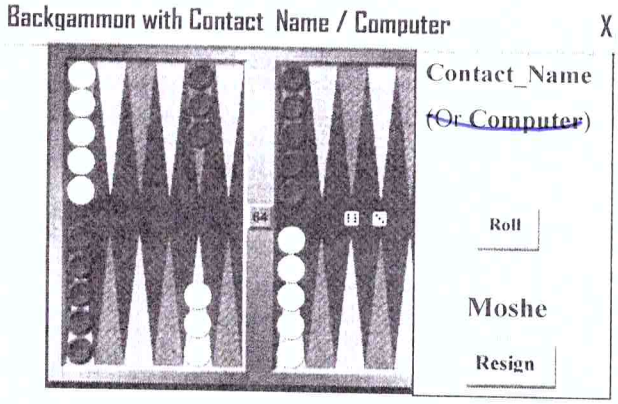


The title of this screen will be "Chat with <name of contact>."

#	Object	Title	Type	Values	Functionality
1	Window header	Chat with <name of contact>	Typical windows window title	The contact's name should appear here.	-
2	Chat history area	-	Border	-	Read only. Displays the messages sent and received by the user, in order they were sent.
3	Message composing area	-	Text area box	-	Here the user can write messages to send to his contact.
4	Send button	Send	Button	-	Enabled only if there are any characters entered in the message composing area. When selected, the message will get sent to the contact, be erased from the composing area and appear in the chat history. Note: pressing "enter" works in the same way.

#	Object	Title	Type	Values	Functionality
5	Game button	Game icon	Button	-	Always enabled. When selected, a popup will open, telling the user his contact is being asked to confirm the game request. If the contact confirms, the popup closes and the game window is launched. If the contact declines, the popup will notify the user.

7.5. *Game screen*



- In case of a game against a contact, the title of this screen will be "Backgammon with <name of contact>."
- In case of playing against the computer, the title should be "Backgammon against the computer."

#	Object	Title	Type	Values	Functionality
1	Window header	Backgammon with <name of contact> / the computer	Typical windows window title	If playing against contact, the contact's name should appear here.	-
2	Game border	-	Border		The game board appears here, with the pieces set up as in the beginning of a backgammon game (see illustration). Playing pieces can be picked up and put down as defined in the game screen section.

#	Object	Title	Type	Values	Functionality
3	Game pieces	-	Icons	-	Only the pieces of the color the user is playing can be clicked. Enabled only when it is the user's turn. When clicked, the mouse pointer will change to the piece. Can be moved only to places legal in the game.
4	Action border	-	Border		Appears to the right of the game board. Should have player's name on the bottom, and opponents name on the top. Also has buttons for rolling and resigning.
5	Roll button	Roll	Button	-	Enabled only at the beginning of a game or when opponent completed his turn. Upon selection: • At the beginning of the game, one die should appear rolled on the right part of the game board. • During the game, two dice should appear rolled on the right part of the board.
6	Resign button	Resign	Button	-	Enabled only when the game has already started and it is the player's turn. Upon selection, a popup window will open, asking the user to confirm his resignation. If the user confirms, a different popup will notify him he lost the game. If he cancels, the popup window will close.