Documentation: Use Cases

*We made UI designs for all the screens of our project and assembled them in a file named **ui.pdf**. Such file may be really handy when reading the following documentation, we deeply encourage readers to check it out while doing so.*

Login screen

Create user

Parameters:
- username (string)

Preconditions:
- There is no user logged in.

Flow:
1. User writes the desired username in a text field.
2. User clicks the “OK” button.
3. User is added to the database with default values.
4. User is automatically logged in and redirected to the main dashboard.

Exceptions:
- If the provided username is already registered in the database, an exception will be thrown and the client will notify the user.

Login

Parameters:

Preconditions:
- There is no user logged in.

Flow:
1. User clicks his corresponding profile in the profile list.
2. User is logged in and redirected to the dashboard.

Exceptions:

Navigation Bar

Dashboard (from navigation bar)

Parameters:

Preconditions:

- User is logged in.

Flow:

1. User selects the “DASHBOARD” tab in the menu bar.
2. The “DASHBOARD” screen is loaded.

Exceptions:

Kakuro List

Parameters:

Preconditions:

- User is logged in.

Flow:

1. User selects the “KAKURO LIST” tab in the menu bar.
2. The “KAKURO LIST” screen is loaded with “EASY” as the selected difficulty.

Exceptions:

My Kakuros

Parameters:

Preconditions:

- User is logged in.

Flow:

3. User selects the “MY KAKUROS” tab in the menu bar.
4. The “MY KAKUROS” screen is loaded.

Exceptions:

Statistics

Parameters:

Preconditions:

- User is logged in.

Flow:

5. User selects the “STATISTICS” tab in the menu bar.
6. The “STATISTICS” screen is loaded.

Exceptions:

Logout

Parameters:

Preconditions:

- User is logged in.

Flow:

1. User selects the “LOGOUT” button in the menu bar.
2. User is logged out and redirected to the “LOGIN” screen

Exceptions:

Dashboard

New Game (select Difficulty)

Parameters:
Preconditions:

- User is logged in.

Flow:

1. User presses one of the difficulty buttons in the “NEW GAME” section of the dashboard.
2. The “KAKURO LIST” screen is loaded and, once loaded, the selected tab is the difficulty chosen by the user.

Exceptions:

New Game (import Kakuro)

Parameters:

- file (string)

Preconditions:

- User is logged in.

Flow:

1. User presses the “IMPORT KAKURO” button in the “NEW GAME” section of the dashboard.
2. User provides a file by selecting it in a file explorer select file dialog.
3. The provided Kakuro is verified.
4. The provided Kakuro is added to the database as “USER_CREATED”.
5. User is redirected to the play kakuro screen and starts playing the imported Kakuro.

Exceptions:

- If the provided Kakuro is not valid, an exception will be thrown and the client will notify the user.

Generate Kakuro (by dimensions and difficulty)

Parameters:

- rowSize (int)
- colSize (int)
- difficulty (int) -> 0: EASY, 1: MEDIUM, 2: HARD, 3: EXTREME

Preconditions:

- User is logged in.

Flow:

1. The Kakuro generator generates a Kakuro with the provided dimensions and difficulty with a random seed.
2. User is sent to the play Kakuro screen for the generated Kakuro.

Exceptions:

Generate Kakuro (by seed)

Parameters:
- seed (int)

Preconditions:
- User is logged in.

Flow:

3. The Kakuro generator generates a Kakuro using the provided seed.
4. User is sent to the play Kakuro screen for the generated Kakuro.

Exceptions:
- If the seed is out of bounds, an exception will be thrown and the client will notify the user.

Create Kakuro (new Kakuro from size)

Parameters:
- rowSize (int)
- colSize (int)

Preconditions:
- User is logged in.

Flow:

1. The provided dimensions for the Kakuro that the user aims to create.
    a. If the dimensions are too big, a dialog will pop up warning the user that the UI may not be properly displayed.
2. The “KAKURO CREATOR” screen is loaded with an empty board (all cells are white with no defined value) with the desired dimensions.

Exceptions:
- If the dimensions are not valid, an exception will be thrown and the client will notify the user.

Create Kakuro (load WIP Kakuro)

Parameters:
- file (string)

Preconditions:
- User is logged in.

Flow:

1. User presses the “IMPORT” button in the “CREATE KAKURO -> HAND MADE” section of the dashboard.
2. User provides a file by selecting it in a file explorer select file dialog.

3. The format of the provided Kakuro is verified. We don’t verify the entire Kakuro here because since it is supposed to be a WIP Kakuro it may not be valid yet, hence, we verify that the content of the file is a Kakuro written in the standard format but we don’t verify the correctness of the board.
4. The “KAKURO CREATOR” screen is loaded with the provided Kakuro.

Exceptions:

- If the provided Kakuro is not valid, an exception will be thrown and the client will notify the user.

Resume game

Parameters:

Preconditions:

- User is logged in.

Flow:

1. User presses the “RESUME GAME” button for one of the in-progress Kakuros listen in the “HISTORY” section of the dashboard.
2. User is redirected to the “PLAY KAKURO” screen for such game and is able to keep playing.

Exceptions:

Kakuro List

List kakuros by Difficulty

Parameters:

Preconditions:

- User is logged in.

Flow:

1. User presses the “KAKURO LIST” button in the navigation bar.
2. The initial difficulty that will be loaded is “EASY” but user will be able to choose among “EASY”, “MEDIUM”, “HARD” and “BY USERS” by clicking to the corresponding tab.
3. The filtered list is displayed in the screen.
4. For each Kakuro, the user will be able to choose between these options:
    a. Export Kakuro.
    b. Play Kakuro (only if there is no game in progress for such Kakuro).
    c. Resume game (only if there is a game in progress for such Kakuro).

Exceptions:

Export Kakuro

Parameters:

- outputDirectory (string)

Preconditions:

- User is logged in.

Flow:

1. User selects the output directory by selecting it in a file explorer select directory dialog.
2. A file is created in the form of <kakuroID>.kak in the provided output directory containing the desired Kakuro in the standard format.

Exceptions:

- If the output directory cannot be reached, an exception will be thrown and the client will notify the user.

Play Kakuro

Parameters:

Preconditions:

- User is logged in.

Flow:

1. User presses the “PLAY GAME” button for one of the listed Kakuros.
2. User is redirected to the “PLAY KAKURO” screen and a new game for such Kakuro starts.

Exceptions:

Resume game

Parameters:

Preconditions:

- User is logged in.

Flow:

1. User presses the “RESUME GAME” button for one of the listed Kakuros.
2. User is redirected to the “PLAY KAKURO” screen and keeps playing the game in progress for such Kakuro.

Exceptions:

My Kakuros

This screen is analogous to the “List Kakuros” screen but instead of tabs for each difficulty and a different list for such, there are only the Kakuros created by the user that is currently logged in.

Play Kakuro

Play Kakuro

Parameters:
Preconditions:

- User is logged in.

Flow:

1. The Kakuro play screen is displayed, initially the user will see the empty state of the Kakuro.
2. At any given point the user will be able to exit and save progress (or not) for the game that they are playing.
3. While the user is playing they will be able to perform a sequence of actions:
    a. Undo movement: the last movement done by the user is undone and it is reflected in the board. If the user hasn't done any movement this button will be deactivated.
    b. Redo movement: the last reverted movement is reapplied and reflected in the board. Everytime the user makes a movement this button is deactivated since that movement will be counted as the most recent one.
    c. Clear board: the board is set to the initial stat and the movement list is erased.
    d. Click on a movement list item: when clicking on a movement list item, the board will be set as it was in such movement.
    e. Mark movement: the movement is marked in the movement list.
    f. Enable or disable help options: the user will be able to enable or disable the following help options:
        i. Show the values used in row/column in red in the number pad.
        ii. Show all combinations for the selected cell's row and column.
        iii. Auto erase notations when value is assigned to a cell in the row or column.
    g. Select cell: when the user clicks a cell we say that it is "selected", the selected help options will be displayed accordingly. The user will be able to do the following actions:
        i. Set value: user selects a number for the selected cell among the possible values and it is inserted in the board.
        ii. Clear value: the value for the selected cell is cleared.
        iii. Modify notations: the notations for the selected cell are updated as the user desires.
    h. Ask for a hint: if the current state of the board has no solution, it marks a movement in the movement list that provoked the board to not have a solution previous to which it was still solvable. If however, the current state of the board has a solution, it marks a cell in the board the value of which can be

inferred, in this case, if the user asked for a second hint, such cell would take the correct value.

i. Solve Kakuro: a solution for the Kakuro that is being played is displayed in the play screen. Current game is finished and user does not get any points from it. The solver does not take into account the movements already done by the user.

Exceptions:

- When the user makes an invalid movement, the conflicting cells will be marked in red to inform the user of their failure.

Kakuro Creator

Create Kakuro

Parameters:

Preconditions:

- User is logged in.

Flow:

1. In the Kakuro creator screen user will be able to execute a sequence of several actions:
    a. Select black cell edit tool: when using this tool, if the user clicks on a white cell, it will turn black, on the other hand, the user clicks on a black cell it will enter “value assignment mode”.
    b. Select white cell edit tool: when using this tool, if the user clicks on a black cell, it will turn white, on the other hand, if the user clicks on a white cell it will enter “value assignment mode”.
    c. Assign a value to a cell: the possible values for a selected cell are displayed in the screen, if the user chooses one of them, the chosen value is assigned to the cell. The user can clear the value of a specific cell pressing the “X” button located together with the possible values for such cell.
    d. Fill Kakuro: runs the generator from the current state of the board and reflects the result in the UI.
    e. Clear the entire board: cell values are erased from the board.
    f. Export board: a dialog is displayed so that the user can choose the output directory of the WIP kakuro.
    g. Validate and save: the Kakuro will be verified and stored in the database with the chosen name. The user will be redirected to the “MY KAKUROS” screen.
2. While the user is changing the board the client will be as helpful as possible noticing all the anomalies in order to provide a good UX.
3. At any given time the user would be able to leave the Kakuro creator without saving the result by either closing the program or by navigating to a different tab using the navigation bar buttons.

Exceptions:

- If the user tries to save a Kakuro and such Kakuro is not valid, an exception will be thrown and the client will notify the user.
- If the user tries to save a Kakuro and the name for such Kakuro is already used in the database, an exception will be thrown and the client will notify the user.