Umm Al-Qura University
Computer & Info. Systems College
Computer Science department

Advanced Programming
3rd Semester 2022-2023
Course Project

نبراس
متعة التعلم

# Nebras Application

## Project Number 1

| | |
|---|---|
| Jenan Mustafa | ID 442015713 |
| Ruba Balubaid | ID 442017269 |
| Dana Algamdi | ID 442008509 |
| Farah Alhasani | ID 442013345 |
| Ahlam Almatrafi | ID 442002618 |

Advanced Programming Course

Instructor: Dr. Aisha Alsiyami

2022- 2023

## 1- Project Title

Nebras Application.

## 2- Project Idea and Project Aims

The idea of our application is a fun and educational game for kids it aims to increase kid's knowledge about several interesting topics. it will contain 3 categories (Math, geography, and religion) each category will have a different question and each question is considered a level. So, when the user wins the first level, level 2 will automatically be accessible, and more points will be added to the user's record. Also, the user's process is saved along with his/her account in the game's database.

## 3- Project Functionalities

Mention all the functionalities or services which are provided by your application in details.

1. Sign up: by entering a valid email address, username, and password. Then all this information will be stored in the player table (Database).

2. Login: in this service, the user must enter his username and password to be checked and compared with the stored data in the database after signing up.

3.Edit info: this service allows the user to edit his/her email address and password if needed.

4. Settings:
- Show Account Information: display the information of the logged-in account (Username, Email Address, Password, Points count).
- Who we are? : introduction to our project's idea.
- Log out: for logging out of the current account and return to the main interface where you can sign up for a new account or log in again.

5. Follow up on the player's process and open new levels depending on that process.

6. I/O file writes in the registration file all of the sign-up information for each user.

# 4- Project Design and Implementation

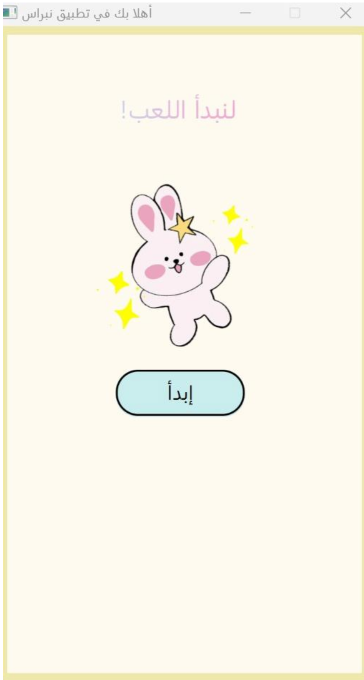
## A. Graphical User Interface

Write paragraph(s) that answers all of the following questions:

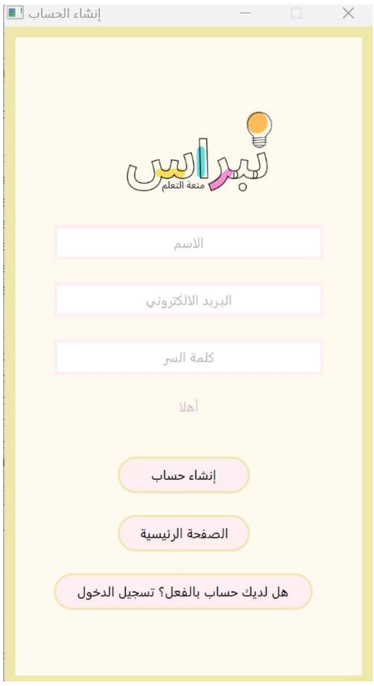
What have been designed? How many interfaces? (add screen shots).
What is the purpose of each interface?
Explain in details the usage flow of your application.

Our team designed 13 different interfaces. Starting with the welcome, sign-up, and log-in scenes. Moving along to the player's information, edit information, the game levels (5 levels), settings, and the who we are scene. Our first scene is the welcome scene that appears once the user opens the application. It shows a short welcome message and a start button to begin the game. After clicking the start button the user will access the sign-up interface and may choose to log in or sign up for a new account. If he/she wishes to create a new account he/she must enter a valid username, email address, and password (password must contain at least 8 characters). After that, the user must log in to check and compare the entered data with the stored data in the database (only username and password are asked for when logging into an existing account). If the user wishes to change his/her password or email address for any reason that is possible by the edit info scene. Next, the player will be able to access the game successfully and the levels interface will appear where the user can start playing level 1 and earn points to open levels 2,3,4 until level 5. The user could also access the settings interface for checking his/her profile, who are we page, go back to the level 3 page, or log out of the current account. First, the player's profile will display the player's email, password, and number of points depending on how many levels have been played. Second, is the Who Are We page which shows a short description of our application's idea. Moreover, our levels are made of 2 religious questions, 2 mathematical questions, and 1 geographical question for kids to answer. Finally.

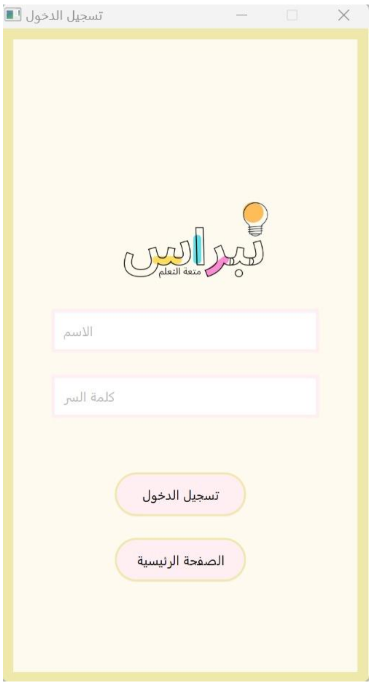
## 1) Welcome Interface



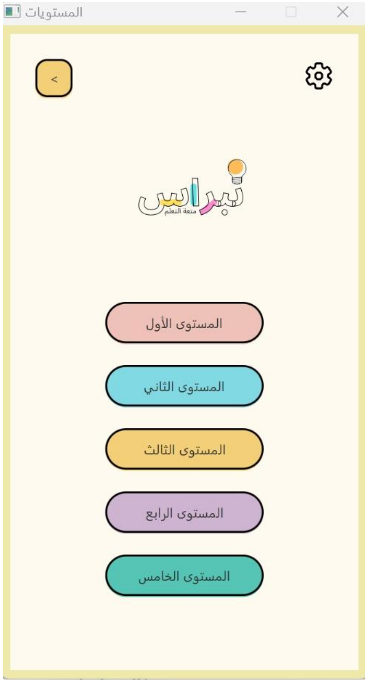
## 2) Sign Up Interface



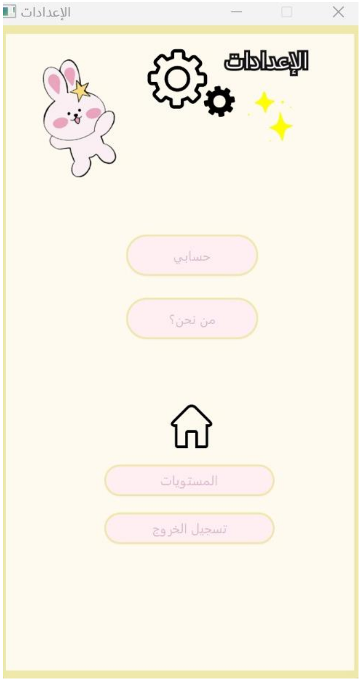
## 3) Log-in Interface



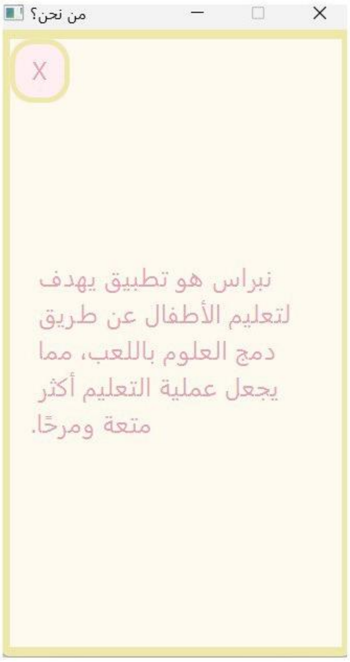
## 4) Levels Interface



### 5) Settings Interface

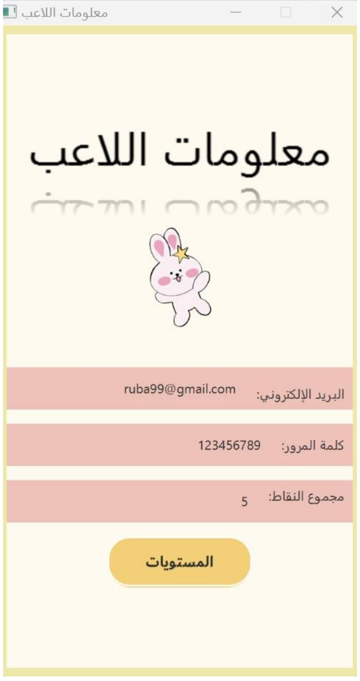


### 6) Who are we Interface



نبراس هو تطبيق يهدف
لتعليم الأطفال عن طريق
دمج العلوم باللعب، مما
يجعل عملية التعليم أكثر
متعة ومرحًا.

### 7) Plyers Info Interface



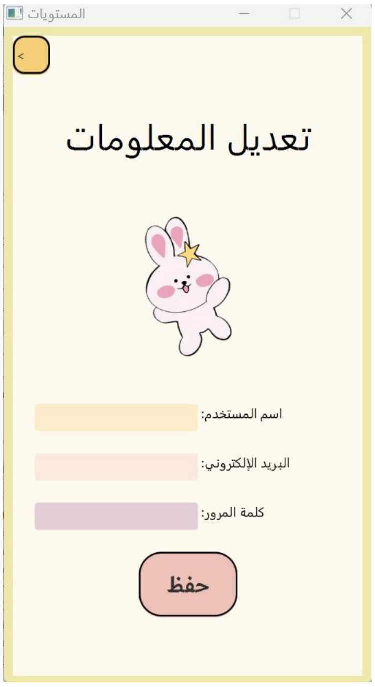
معلومات اللاعب

البريد الإلكتروني: ruba99@gmail.com

كلمة المرور: 123456789

مجموع النقاط: 5

### 8) Edit Info Interface



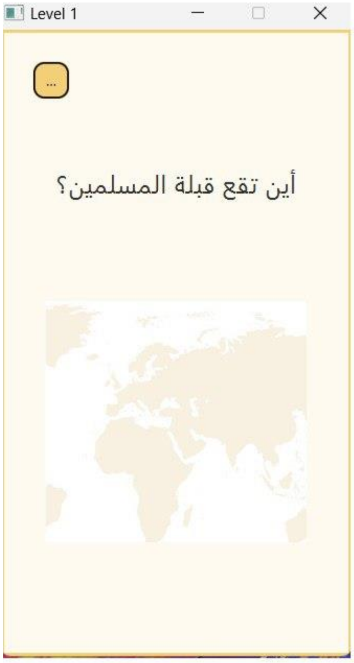
تعديل المعلومات

اسم المستخدم:

البريد الإلكتروني:

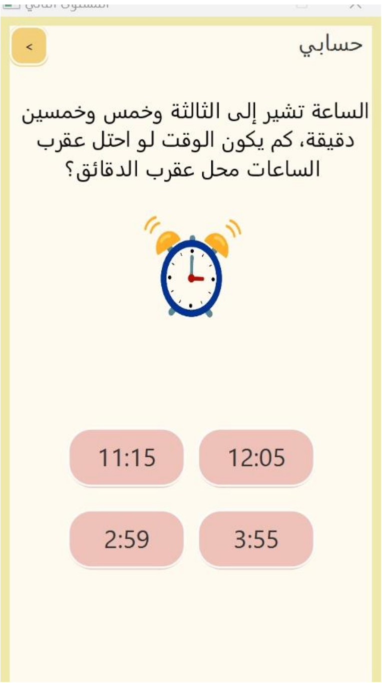
كلمة المرور:

### 9) Level 1 Interface



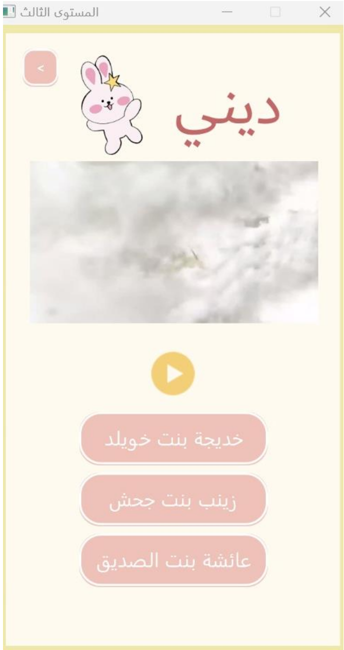
أين تقع قبلة المسلمين؟

### 10) Level 2 Interface

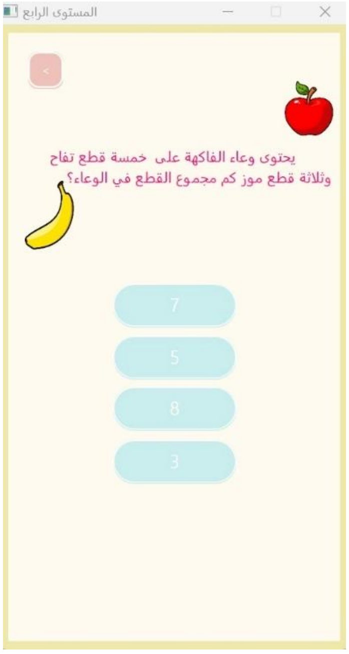


الساعة تشير إلى الثالثة وخمس وخمسين دقيقة، كم يكون الوقت لو احتل عقرب الساعات محل عقرب الدقائق؟

### 11) Level 3 Interface

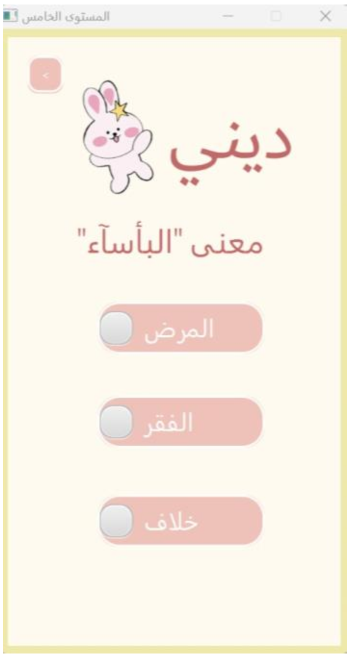


### 12) Level 4 Interface



يحتوى وعاء الفاكهة على خمسة قطع تفاح وثلاثة قطع موز كم مجموع القطع في الوعاء؟

### 13) Level 5 Interface



معنى "البأساء"

## B. Event-Driven Programming

Write paragraph(s) that answers all of the following questions.

How many events?

What types of events have been implemented in your Application?

Mention and explain all the event sources in your application.

Our project contains 51 events made of 39 setOnActions, 11 setOnMousePressed, and 1 setOnKeyPressed action. Here are some of the included events:

- Clicking on the startButton in the welcome scene changes the scene to the registration scene.
- Pressing the Enter key in the welcome scene also changes the scene to the registration scene.

- Clicking on the topButton in the playerInfo scene changes the scene to the levels scene.
- Clicking on the btnRegi button in the registration scene checks and creates a new player if the checks pass.

- Clicking on the btnMain button in the registration scene changes the scene to the welcome scene.
- Clicking on the btnLogin button in the welcome scene changes the scene to the login scene.

- Clicking on the btnLoginScene button in the login scene checks and changes the scene to the levels scene if the checks pass.
- Clicking on the btnMainlogIn button in the login scene changes the scene to the welcome scene.

- Clicking on the btnProfile button in the settings scene changes the scene to the playerInfo scene and retrieves player information from the database.
- Clicking on the btnwho button in the settings scene changes the scene to the "who we are" scene.

- Clicking on the settingHomeBtn button in the settings scene changes the scene to the levels scene.
- Clicking on the btnLogout button in the settings scene changes the scene to the welcome scene.

- Clicking on the btnClose button in the "who we are" scene changes the scene back to the settings scene.

There are many events implemented in this application. Most of them are related to button clicks or mouse presses on various GUI components such as buttons, text fields, and labels. The events implemented in this code can be grouped into several categories. The first category includes events related to changing the scene to a different screen. For example, clicking on a "start" button or pressing the enter key on the welcome screen switches to the registration screen. Clicking on a "login" button switches to the login screen. Clicking on a "back" button goes back to the previous screen. These events are handled by event handlers that contain code to change the scene and set the title of the window.

The second category includes events related to validating user input and creating or updating database records. For example, clicking on the "register" button triggers validation checks and creates a new player if the checks pass. Clicking on the "login" button triggers validation checks and switches to the levels screen if the checks pass. These events are handled by event handlers that contain code to retrieve user input, validate it, and interact with the database as needed.

The event handlers in this code also interact with the database using Hibernate. They retrieve player state information from the database and update it based on the user's actions in the game. For example, when the user selects the correct answer in a level, the code increments the player's star count in the database. These database interactions are important for persisting the user's progress and allowing them to resume their game later.

The third category includes events related to gameplay. For example, clicking on objects or buttons in the game screen triggers actions such as playing sound effects, displaying messages, and updating player state. These events are handled by event handlers that contain code to execute the necessary actions based on the user's input. In summary, the event sources in this code are the buttons, mouse presses, keyboard input on the graphical user interface components, and the event handlers interact with the database using Hibernate to persist the user's progress.

## C. Java Database Programming

Write paragraph(s) that answers all of the following questions.

Explain the need of database in your application and how you utilized from each table you have created in your database.

How many tables? What are the columns?

Explain how you connect your object model to your relational model?.

Explain the relationship between your database and your GUI, i.e mention all the mappings between the columns and UI components.

The database is an essential component of this application as it allows the user's progress to be saved and retrieved at a later time. The database is used to store player information, including their username, password, and their progress through the levels of the game. This information is used by the application to display the appropriate level to the user when they log in and to update their progress as they complete each level. There are two tables in the database created for this application: the "player" table, and the "player_state" table. The "player" table contains the columns "player_name", "player_email", and "player_password", which are used to store the player's login data. The "player_state" table has columns "name_of_player", and "number_of_stars", which are used to store the player's progress through the game. To connect the object model to the relational model, the application uses Hibernate, which is a Java-based ORM (Object-Relational Mapping) framework. The Hibernate framework maps the Java objects to the corresponding database tables, allowing the application to interact with the database. The application defines Java classes that represent the data stored in each table, and Hibernate maps these classes to the corresponding database tables. The relationship between the database and the GUI is established through mappings between the columns in the database tables and the UI components. So, when the user logs in, the application retrieves their progress from the "player_state" table and displays the appropriate level to the user on the screen. When the user completes a level, the application updates the "player_state" table to reflect the player's progress and the number of stars earned. In summary, we wrote classes using Java as a model and used JavaFX code as a view, and at the end of the code we added the controller.

## D.Model-View Controller

Write paragraph(s) that answers all of the following questions.

Explain how you have implemented the MVC pattern?

Did you use any tool to design your application using this pattern? Which tool?

We only used JavaFX code to design our application interfaces.

### E. Extra

Write paragraph(s) that answer all of the following questions.

If you did any extra work, explain what is it? Why you did it? How it is related to your application?

We've done some extra work using Animation, Multimedia, and I/O file, to add to our experience by applying the concepts we learned in the course to our project. First, we made an I/O file that writes in a file named registration the sign-up information for each user. Second, we used multimedia for one of the levels where we added a video for the player to watch and answer the question. Finally, we added effects and animation on the rabbit and the stars images in the first interface (Welcome Scene). It's related to our project because we wanted to make interfaces that will catch kids' attention and by applying multimedia and effects/animation we assured that kids will enjoy playing this game.

## 5- The distribution of the work

In this section, mention the distribution of the work between the team. Be honest and feel free to show your tasks individually. Each member (if you want) can calculate the spent time in this project (days, hours or even minutes).

| Member | Task(s) |
|---|---|
| 1- Jenan | Player info scene - Edit info scene<br>level 2 scene - Report |
| 2- Ruba | Level 1 scene - Levels scene<br>Edit info scene -Sign Up scene<br>Login scene - Database |
| 3- Dana | Settings scene - Who are we scene<br>Sign up scene – Login scene<br>I/O file |
| 4- Farah | Welcome scene - Level 4 scene<br>Animation |
| 5- Ahlam | Level 5 scene - Level 3 scene<br>Report |

## 6- What we learnt

Write paragraph(s) that answers the following questions.

What you have learned from the project?
What is the advantages you gained from designing this project?
Did you feel the teamwork was very useful in programming project?
Are you satisfied with your application?
Did you feel you will be able to work in a large scale project later?

Through our work on the "Nabras" project, we gained many skills related to game design using JavaFX. We acquired experience in analysing errors and overcoming the challenges that we faced during the design process. Additionally, we developed skills in time management, task scheduling, and teamwork. Effective teamwork streamlined the development process, increased productivity, and improved the overall quality of the final program. Furthermore, teamwork allowed us to learn from one another, share knowledge, and build on each other's strengths to achieve the best possible outcome. While we are satisfied with the "Nabras" application, we aspire to improve it, as there may be gaps and errors that need addressing. Working on large-scale projects requires experience and a readiness to learn and adapt to new challenges, and smaller projects can provide a valuable experience over time to prepare for larger-scale projects in the future.

## 7- Project difficulties

Write paragraph(s) that answers the following questions.

Mention any difficulties you have faced in this project if any (Resources difficulties, technical difficulties, programming difficulties or teamwork difficulties)

We didn’t face any resource difficulties, but we encountered technical challenges during the project. We had issues with different versions of NetBeans, such as version 8.0.2, which did not support some of the features present in version 8.2. When exchanging code among team members, it was incompatible with different versions. Additionally, Java version 17 did not support the Hibernate file, which was a significant obstacle, and it took a lot of time to resolve such issues. Regarding programming difficulties, we faced challenges in inserting some files, such as videos. However, the solution was simple. We also faced a problem with connecting the database on some devices. Regarding teamwork difficulties, there were challenges in coordinating meeting times, but cooperation and communication among team members helped overcome these challenges.

## 8- Project Declaration

☑We confirm that the work of this project was solely undertaken by ourselves and that no help was provided from other sources as those allowed. As well as we confirm that we completely aware of the violation consequences of the academic integrity.

Read and check the previous declaration.

List all your IDs

ID 442015713
ID 442017269
ID 442008509
ID 442013345
ID 442002618