

**FACULTY OF COMPUTER SCIENCE AND
INFORMATION TECHNOLOGY
UNIVERSITI MALAYA**

**SEMESTER 2, SESSION 2023/2024
WIA1002 DATA STRUCTURE
GROUP PROJECT REPORT**

Occ	5		
Group Name	THINK LATER		
Topic	TOPIC 5: HACKING THE FUTURE		
Members	No.	Name	Matric No.
	1.	ANG LI JIA	23005237
	2.	BOO EE VONE	23005027
	3.	CHAN LIN NA	23004910
	4.	CHUA HUI YING NICOLE	23005225
	5.	TAN CHIEN LING	23004992
Lecturer	MOHAMAD HAZIM BIN MD HANIF		
Demo	1. MUHAMMAD FARIS BIN AHMAD FAIZ		
	2. AHMAD ILHAM BIN BAHARUDDIN		

1.0 Basic Requirements

In this section, you are required to state, clarify, and explain all basic requirements needed to complete the project. You may attach the sample outputs, but do not attach the screenshots of your program or source code. Please provide the accessible link for online sharing, i.e., GoogleDrive or Github. Note that, in your source code, please include comments to explain your code briefly for better understanding.

SOURCE CODE LINK:

[SOURCE CODE](#)

User Account

Explanation:

Users can view their own account after they logged in.

If a user logged in as a “Student”, they will be able to view their email, username, role, location coordinate, Parent(s), Friends and their current point.

If a user logged in as a “Parent”, they will be able to view their email, username, role, location coordinate, children and their past booking made.

If a user logged in as a “Educator”, they will be able to view their email, username, role, location coordinate, number of quizzes created and number of events created.

Sample Output:

The figure consists of three separate windows, each titled "User Account".

- Student Role (Top Left):** Shows fields for Email (ai@hotmail.com), Username (ai), Role (Student), Location Coordinate (X: -48, Y: 2), Parent(s) (firdaus_an), Friends (hz), and Current Points (7).
- Parent Role (Top Right):** Shows fields for Email (jazheng@gmail.com), Username (jz), Role (Parent), Location Coordinate (X: 108, Y: 172), Children (hz), and Past Booking Made (Child ID: hz, Destination: B).
- Educator Role (Bottom):** Shows fields for Email (In@edu.com), Username (Inedu), Role (Educator), Location Coordinate (X: 458, Y: -133), Number of Quizzes Created (0), and Number of Events Created (0).

Login / Registration Page

Login:

Users can choose their role to log in which is “Student”, “Parent” and “Educator”.

The user can choose to log in with a valid email or with a valid username.

Users must enter the correct password and username/email in order to login successfully.

If a user enters a username/email that is invalid or incorrect password, a message “Incorrect username or password” will prompt the user.

After the user logged in successfully, a message “You are successfully login as: (their role)” and they will bring it to the Main page.

Sample Output:



Registration:

Users can choose to register with the role “Student”, “Parent” or “Educator”.

Users must fill in their Full Name, Email, Username, Password, Confirm Password before they register and they must ensure the username does not been used before and also the Password and the Confirm Password field must be the same.

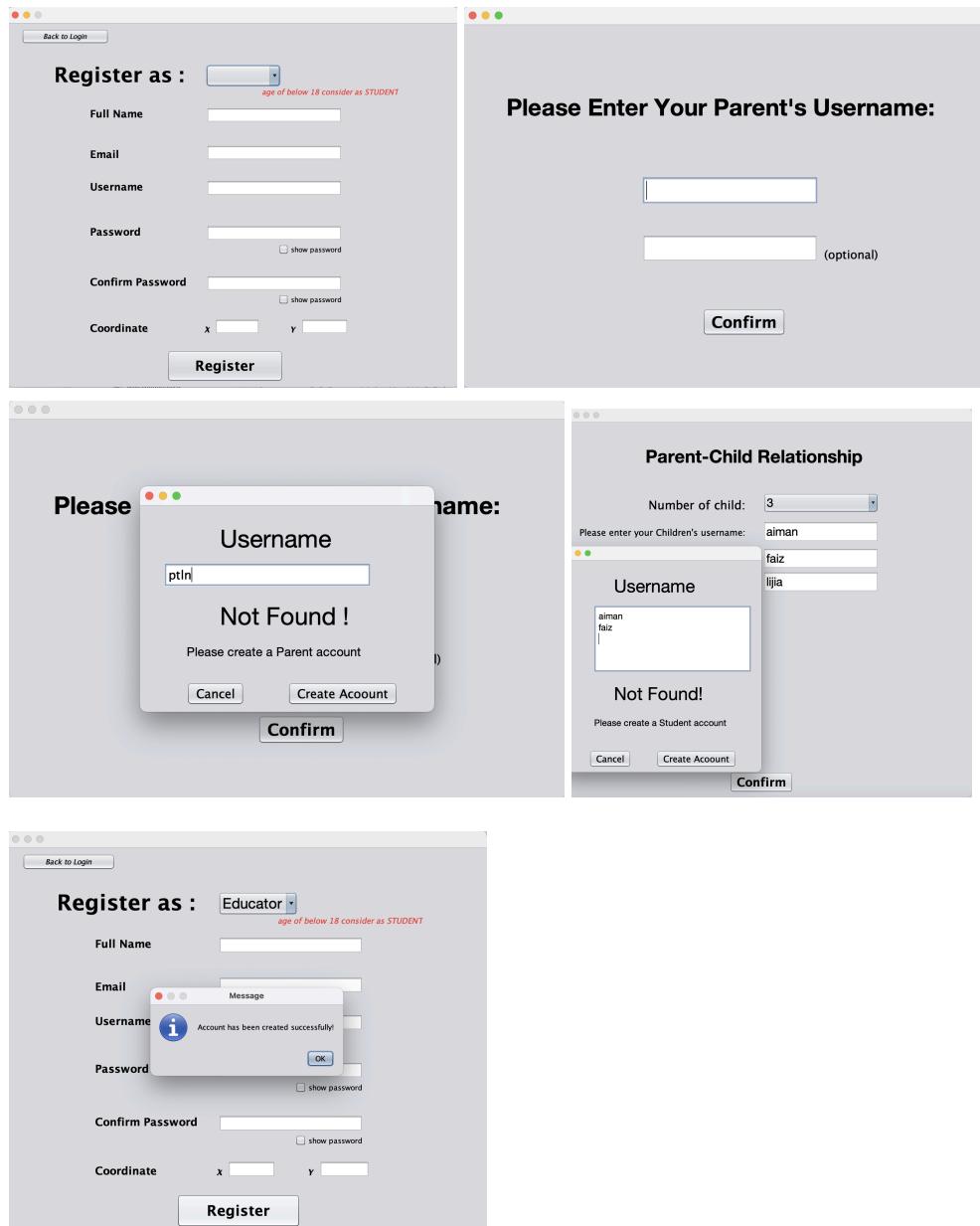
After the user pressed the “Register” button, both Coordinate X and Y were automatically generated for the user from a random number of any real values in the range of -500.0 and 500.0.

If the user registers as a “Educator”, a message “Account has been created successfully!” will prompt out to the user and the user will be brought to the login page to login after all requirements are fulfilled

If the user registers as a “Student”, they will bring to the next page which is the EnterParent page where they must confirm their parent’s name. Student can choose to enter only 1 parent’s name or both parent’s names on this page. They must enter a valid parent’s username in order to sign up successfully. If not, they can choose to create an account for their parents. After all requirements are fulfilled, a message “Account has been created successfully!” will prompt out to the user and the user will be brought to the login page to login.

If the user registers as a “Parent”, they will be brought to the next page which is the EnterChildren page where they must confirm their children’s name. Parents can choose their number of children in the combo box and the same number of textfield will pop out for users to enter their children’s username. They must enter a valid student’s username in order to sign up successfully. If not, they can choose to create an account for their children. After all requirements are fulfilled, a message “Account has been created successfully!” will prompt out to the user and the user will be brought to the login page to login.

Sample Output:



Data Storage

The Parent-Child Relationship will save in the text file named ParentChild.txt. The format of the file is <parent_username>, <child_username>.

For more effective data management sake, the other data is stored by using a database (MySQL).

To access to text files (txt) and SQL:



View Profile

In the main page, there is a “View Profile” button, users can view others profiles by entering a valid username after they pressed the “View Profile” button.

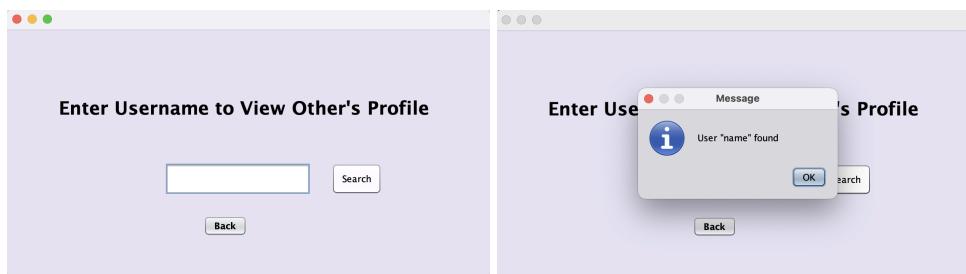
If a user searches for a username that does not exist, an error message “User does not exist” will prompt the user.

If a user searches for a username that exists and has a role of “Student”, they will bring to the next page which is “Student” page inorder to view others profile details. Users can see their Email, Username, Role, Location Coordinate, Friends and their Parent(s). There will also be a “Add Friends” button for users to send a friend request for the user which they are viewing.

If a user searches for a username that exists and has a role of “Parent”, they will bring to the next page which is “Parent” page inorder to view others profile details. Users can see their Email, Username, Role, Location Coordinate, Children and their Past Booking Made.

If a user searches for a username that exists and has a role of “Educator”, they will bring to the next page which is “Educator” page inorder to view others profile details. Users can see their Email, Username, Role, Location Coordinate, Number of Quizzes Created and Number of Events Created.

Sample Output:



Left Screenshot: Enter User's Profile

User does not exists

Right Screenshot: View Profile

Email: name@email.com
Username: name
Role: Student
Location Coordinate: X: 198 Y: -305
Friends: hz
Parent(s): qwe
Current Points: 22
Add Friends

Left Screenshot: View Profile

Email: in@edu.com
Username: inedu
Role: Educator
Location Coordinate: X: 458 Y: -133
Number of Quizzes Created: 0
Number of Events Created: 0

Right Screenshot: View Profile

Email: sad@gmail.com
Username: sad
Role: Parent
Location Coordinate: X: -392 Y: -286
Children: happy
Past Booking Made: No booking history found.

Access Management

Before a user logs in, they will first enter the home page where they can login or sign up. They are not allowed to access other pages before they successfully logged in to an account. After a user logged in, they can access those pages which are accessible according to their role.

Users with the role “Student” can access : Access Quiz, Add Friends and Leaderboard.

Users with the role “Parent” can access : Booking Page.

Users with the role “Educator” can access : Event Page and Quiz Page.

Event, Discussion Page and View Profile is accessible to all roles.

Sample Output:

Left Screenshot: STEM Home Page

Do you know what is SDG?
SDG 4 Quality Education
SDG 9 Industry, Innovation, and Infrastructure
SDG 17 Partnership for the Goals

Right Screenshot: STEM Home Page (Logged In)

Do you know what is SDG?
SDG 4 Quality Education
SDG 9 Industry, Innovation, and Infrastructure
SDG 17 Partnership for the Goals

Create Event & Quiz

Explanation:

Only the user with role “Educator” can access this function. From the Home Page, educator can access the Create Event or Create Quiz pages.

Event page comprises of:

- Create Event
- Edit Event
- Delete Event
- Display Event

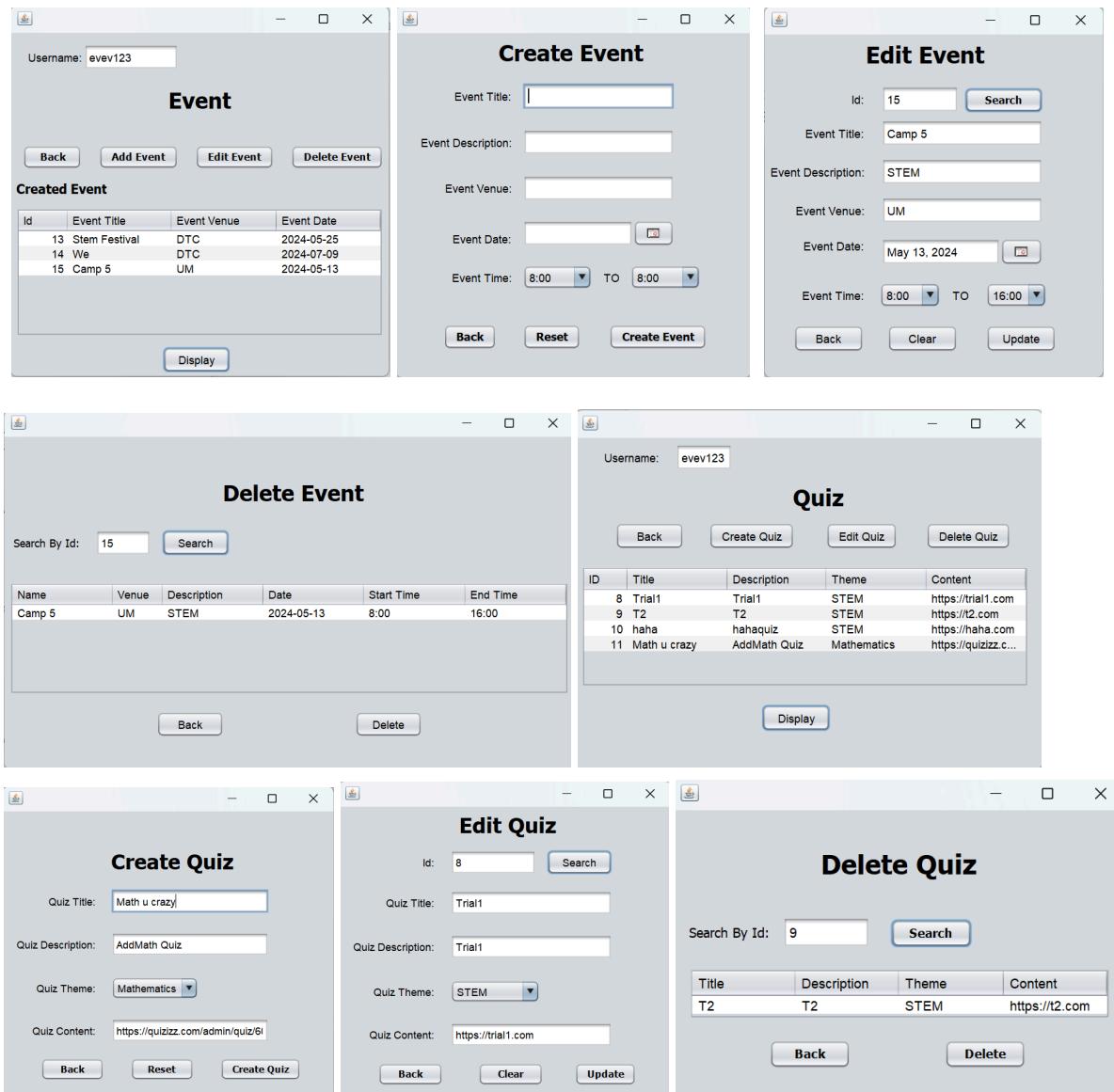
Quiz page comprises of:

- Create Quiz
- Edit Quiz
- Delete Quiz
- Display Quiz

By choosing the “Event” at the Home page, user will be directed to the Event page. From the Event page, the created event will be shown upon clicking the “Display” button. Educator can create an event in the Create Event Page by specifying the Event Title, Event Description, Event Venue, Event Date and Event Time by clicking the “Add Event” button. All columns must be filled in in order to create an event successfully. After creating an event, user can modify the event by searching the event’s ID (the ID is unique). Additionally, if user wants to delete a created event, click the “Delete Event” button and search the ID of the event to delete it. (Note that user can only edit or delete the event created by himself/herself)

By choosing the “Quiz” at the Home page, user will be directed to the Quiz page. If user click “display” button, only the quiz created by current login user will be displayed. To create a quiz, user can click “Add Quiz” button and fill in the Quiz Title, Quiz Description, Quiz Theme (Science/ Technology/ Engineering/ Mathematics) and Quiz content. As quizzing is not the main focus of this assignment, the Quiz Content is replaced with a Quizizz link. The created quiz can be modified by entering the “Edit Quiz” page, searching upon the ID (the ID is unique) of the quiz created and edit the content. User can also delete the quiz created by searching the ID of the quiz and click “Delete” button to remove the quiz. (Note that user can only edit or delete the quiz created by himself/herself)

Sample output:



View Event

All roles (Student, Parent and Educator) can view the events created in the Event page.

Explanation:

a. Data Structure:

In our graphical user interface (GUI), we use the 'DefaultTableModel' from Java's Swing framework to handle and display data in a table format. This model is flexible and allows us to easily add, remove, and update rows of data, which is essential for displaying current event and registration information. The 'DefaultTableModel' helps us sort the data by event dates and times, ensuring that live events (those happening today) and the three closest upcoming events are displayed prominently.

b. Logic:

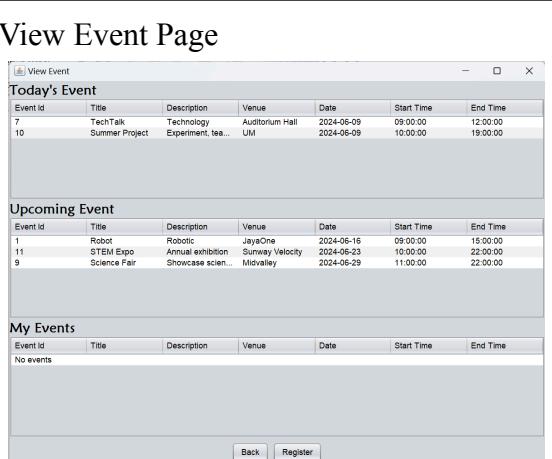
The logic ensures a smooth registration process for students. When a student wants to register for an event, the system first checks if the event is happening today and if it has already started. If so, registration is not allowed. It also checks if the student has already registered for the event or has other events on the same day, which would cause a scheduling conflict. If there are no conflicts, the student is registered for the event, and they earn 5 points. These points are then updated in the 'leaderboard', ensuring that the student's achievements are accurately recorded. The logic also prevents students from registering for multiple events on the same day, ensuring fair participation for all students. For displaying events, it just fetches from the database based on the date.

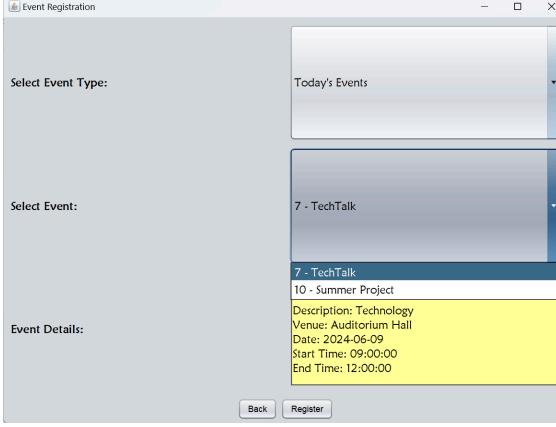
c. Database:

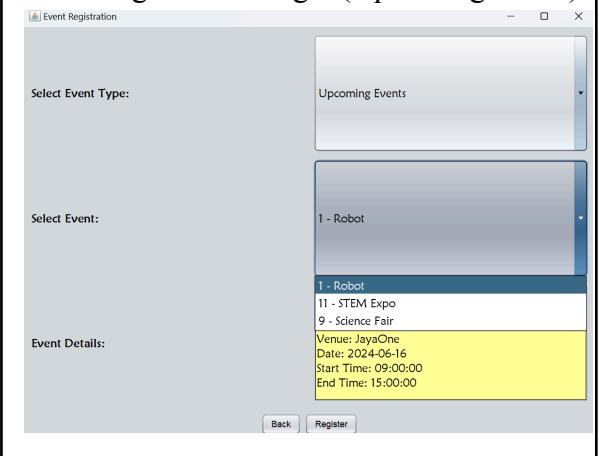
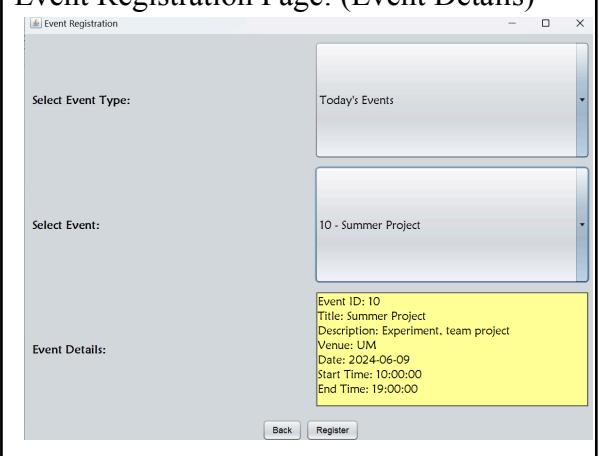
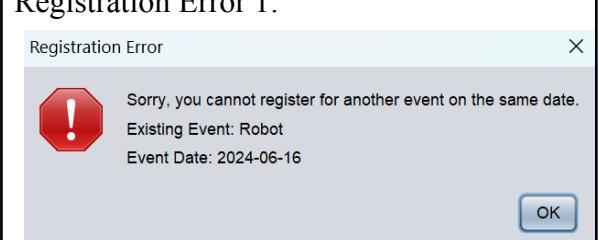
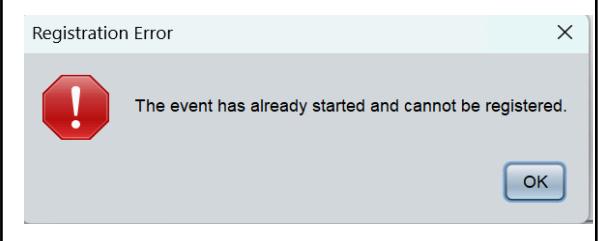
We have four main tables involved in this part: 'event_create', 'event_reg', 'leaderboard', and 'bookings'. The 'event_create' table contains details about each event, including its name, description, date, and time. It is used to display Today's Event and 3 closest Upcoming Events. The 'event_reg' table tracks which students have registered for which events, ensuring that no student registers for the same event more than once and only register for one event per day as well as not registering for the past events. It is used to show My Event. The 'leaderboard' table adds and records the 5 points each student earns from registering for an event, keeping track of their progress. The 'bookings' table checks if a parent has booked something for a student on a particular day, preventing the student from registering for an event on the same day.

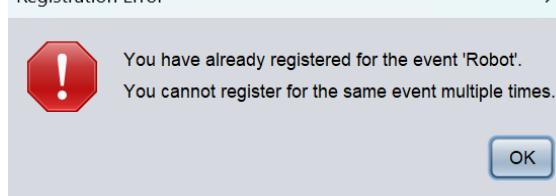
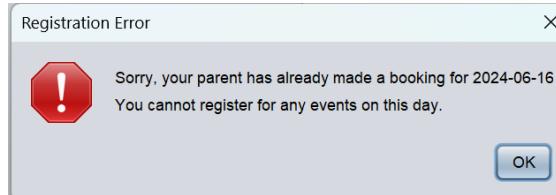
Sample output:

a. Role: Student (View event, Register event and get points)

 <p>View Event Page</p> <p>Today's Event</p> <table border="1"> <thead> <tr> <th>Event Id</th> <th>Title</th> <th>Description</th> <th>Venue</th> <th>Date</th> <th>Start Time</th> <th>End Time</th> </tr> </thead> <tbody> <tr> <td>7</td> <td>Tech Talk</td> <td>Technology Experiment, tea...</td> <td>Auditorium Hall</td> <td>2024-06-09</td> <td>09:00:00</td> <td>12:00:00</td> </tr> <tr> <td>10</td> <td>Summer Project</td> <td>Experiment, tea...</td> <td>UM</td> <td>2024-06-09</td> <td>10:00:00</td> <td>19:00:00</td> </tr> </tbody> </table> <p>Upcoming Event</p> <table border="1"> <thead> <tr> <th>Event Id</th> <th>Title</th> <th>Description</th> <th>Venue</th> <th>Date</th> <th>Start Time</th> <th>End Time</th> </tr> </thead> <tbody> <tr> <td>1</td> <td>Robot</td> <td>Robot</td> <td>JayaOne</td> <td>2024-06-16</td> <td>09:00:00</td> <td>15:00:00</td> </tr> <tr> <td>11</td> <td>STEM Expo</td> <td>Annual exhibition</td> <td>Sunway Velocity</td> <td>2024-06-23</td> <td>10:00:00</td> <td>22:00:00</td> </tr> <tr> <td>9</td> <td>Science Fair</td> <td>Showcase scienc...</td> <td>Mirvalley</td> <td>2024-06-29</td> <td>11:00:00</td> <td>22:00:00</td> </tr> </tbody> </table> <p>My Events</p> <table border="1"> <thead> <tr> <th>Event Id</th> <th>Title</th> <th>Description</th> <th>Venue</th> <th>Date</th> <th>Start Time</th> <th>End Time</th> </tr> </thead> <tbody> <tr> <td colspan="7">No events</td> </tr> </tbody> </table> <p style="text-align: center;">Back Register</p>	Event Id	Title	Description	Venue	Date	Start Time	End Time	7	Tech Talk	Technology Experiment, tea...	Auditorium Hall	2024-06-09	09:00:00	12:00:00	10	Summer Project	Experiment, tea...	UM	2024-06-09	10:00:00	19:00:00	Event Id	Title	Description	Venue	Date	Start Time	End Time	1	Robot	Robot	JayaOne	2024-06-16	09:00:00	15:00:00	11	STEM Expo	Annual exhibition	Sunway Velocity	2024-06-23	10:00:00	22:00:00	9	Science Fair	Showcase scienc...	Mirvalley	2024-06-29	11:00:00	22:00:00	Event Id	Title	Description	Venue	Date	Start Time	End Time	No events							<p>Students will see 3 tables, which are live events (we named it as Today's Event for easy understanding), closest 3 Upcoming Events and My Events table.</p> <ol style="list-style-type: none"> For Today's Event table, it will show all the events that are happening on the current day. If there are no events today, then it will show No events. For Upcoming Event table, let's say today is 2024-06-09, and the upcoming events are on: > 2024-06-16, 2024-06-23, 2024-06-29, 2024-06-30, 2024-07-09 <p>Thus, the first 3 which are the closest 3 upcoming events will be displayed.</p>
Event Id	Title	Description	Venue	Date	Start Time	End Time																																																										
7	Tech Talk	Technology Experiment, tea...	Auditorium Hall	2024-06-09	09:00:00	12:00:00																																																										
10	Summer Project	Experiment, tea...	UM	2024-06-09	10:00:00	19:00:00																																																										
Event Id	Title	Description	Venue	Date	Start Time	End Time																																																										
1	Robot	Robot	JayaOne	2024-06-16	09:00:00	15:00:00																																																										
11	STEM Expo	Annual exhibition	Sunway Velocity	2024-06-23	10:00:00	22:00:00																																																										
9	Science Fair	Showcase scienc...	Mirvalley	2024-06-29	11:00:00	22:00:00																																																										
Event Id	Title	Description	Venue	Date	Start Time	End Time																																																										
No events																																																																

	<p>If there are no events after today, then it will show No events.</p> <p>c. For My Events table, it is only visible to Students. It will show all the current logged-in Student's registered events. If the student did not register for any events, it will show No events.</p> <p>At the bottom of the view event page, it consists of 2 buttons which are the Back button to go back to the Main page and Register button which is only visible to students to register for any events displayed whether on Today's Event or Upcoming Event.</p>
<p>Event Registration Page: (Today's Event)</p>  <p>The screenshot shows the 'Event Registration' window. In the 'Select Event Type:' dropdown, 'Today's Events' is selected. In the 'Select Event:' dropdown, '7 - TechTalk' is selected. The 'Event Details:' section shows the following information for '7 - TechTalk':</p> <ul style="list-style-type: none"> Description: Technology Venue: Auditorium Hall Date: 2024-06-09 Start Time: 09:00:00 End Time: 12:00:00 <p>At the bottom are 'Back' and 'Register' buttons.</p>	<p>After Students press the Register button in view event page, it will show the Event Registration page. In this page, students can choose the event type whether the Today's Event or Upcoming Events.</p> <p>In this case, Student chooses Today's Event then in the select event drop down it will have all today's events similar to what have been displayed in the Today's Event table in the view event page. If there is no today's event then it will show nothing on the select event drop down and cannot register.</p> <p>The bottom of the registration page has a Back button to go back to the View Event page while the Register button to register for the selected event.</p>

<h3>Event Registration Page: (Upcoming Event)</h3> 	<p>If students choose the event type as Upcoming Events in the select event drop down it will show the closest 3 upcoming events similar to what have been displayed in the Upcoming Event table in the view event page. If there is no upcoming event then it will show nothing on the select event drop down and cannot register.</p> <p>The bottom of the registration page has a Back button to go back to the View Event page while the Register button to register for the selected event.</p>
<h3>Event Registration Page: (Event Details)</h3> 	<p>After the student selects the event they want, then it will show the details of the event including Event ID, Title of the event, Description to give more details of the events, Venue of the event, Date, Start time and End Time to give clear information of the event to the Student. So that students can decide which events suit them the most easily and clearly.</p>
<p>Registration Error 1:</p>  <p>Registration Error 2:</p> 	<p>After Student select desired event and click Register button, there might be errors occur:</p> <p>Error 1: It is technically impossible for students to attend an event that clashes with another event they have registered for. Since the event time clashing is prohibited, to make it easier, an error message will be displayed if the event to be registered is on the same day as his/her other registered events. Let's say if the Student already registered for an event on 2024-06-16, then it cannot register for another event on the same date. Thus, it will prompt out the Registration Error 1 message informing Student the error and their existing event they had registered for clarification.</p> <p>Error 2: If it is today's event and the registration time has already passed the event start time, then</p>

<p>Registration Error 3:</p> 	<p>it will prompt out the Registration Error 2 message. Let's say the event started at 9.00 am then Student wants to register at 11.00 am, it will prompt out the error message stating that the event had started and cannot register already.</p>
<p>Registration Error 4:</p> 	<p>Error 3: Every Student can only register for each event once. If Student registers for the same event twice it will prompt out the Registration Error 3 message. Let's say Student already registered for the "Robot" event, then he/she tends to register one more time, then it will show the error message stating that the student has already registered for the event.</p> <p>Error 4: If the Student's parent already made a booking on the date, then Student cannot register for the event on that date already. Otherwise it will prompt out the Registration Error 4 message with the booking date made by their parents.</p>
<p>Successful Registration:</p> 	<p>If no error occurs, then it will prompt the Registration Success message. Since young students can gain 5 points per event registered, thus every successful registration will add 5 points to the Students' current points to encourage them to be more engaged and interested in STEM. The message informs Students the successfully registered event name with their current updated points after registration with a friendly tone.</p>

View Event Page (after successful registration)

The screenshot shows a web application interface for viewing events. At the top, there is a header with the title "View Event Page (after successful registration)". Below the header, there are three tables:

- Today's Event**: Shows events for the current day. One event is listed: "TechTalk" at Auditorium Hall.
- Upcoming Event**: Shows events scheduled for the future. Two events are listed: "Robot" and "MedMath".
- My Events**: Shows events registered by the user. One event is listed: "Robot".

At the bottom of the page are two buttons: "Back" and "Register".

After students have successfully registered for an event, then it will be added and displayed in the My Events table so that students can know which event they have already registered and its details clearly and easily.

b. Role: Parents / Educator (View event only)

View Event Page:

The screenshot shows a web application interface for viewing events. At the top, there is a header with the title "View Event Page". Below the header, there are two tables:

- Today's Event**: Shows events for the current day. One event is listed: "TechTalk" at Auditorium Hall.
- Upcoming Event**: Shows events scheduled for the future. Three events are listed: "Robot", "MedMath", and "STEM Expo".

At the bottom of the page is a single "Back" button.

For parents and educators, both roles can only view the events and cannot register. Thus, only 2 tables are visible to them, which are the Today's Event and Upcoming Event. At the bottom of this page only has the Back button to go back to the Main page. The 2 tables shown are the same with what we mentioned in Students' View Event page, just these 2 roles do not have My Event table.

Attempt Quiz

Young Students can view the quizzes on the Quiz Page and attempt some to receive points.

Explanation:

a. Data Structure:

We use ArrayList for managing and organizing quizzes. When quizzes are fetched from the `quiz_create` table in the database, they are stored in an ArrayList named `allQuizzes`. Each element in `allQuizzes` represents a quiz and holds essential details such as quiz ID, title, theme, educator username and completion status, encapsulated within objects of the Quiz class. We also use a separate ArrayList called `filteredQuizzes` for handling filtered quizzes based on selected themes. When a student applies filters (such as selecting specific themes like Science or Mathematics),

the system dynamically populates 'filteredQuizzes' with quizzes that match the selected themes. This dynamic data structure ensures that only relevant quizzes are displayed to students, which lead to the theme filtering. It also allows for efficient data handling, enabling functionalities like sorting quizzes and displaying quiz summaries.

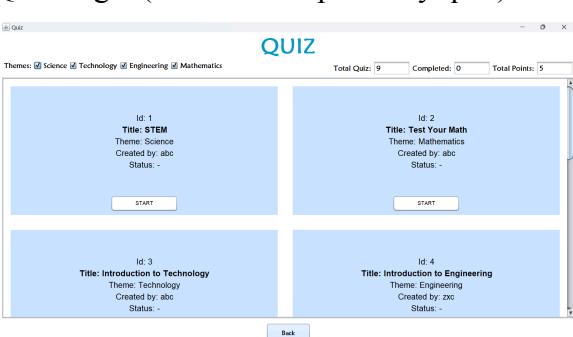
b. Logic:

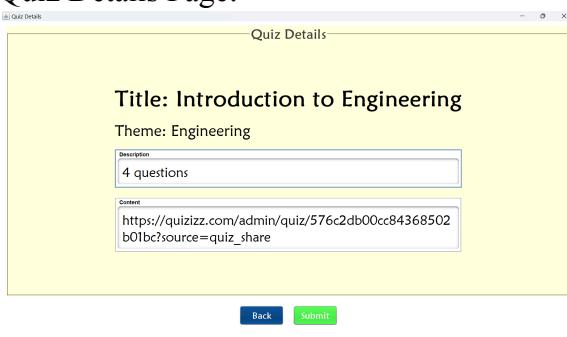
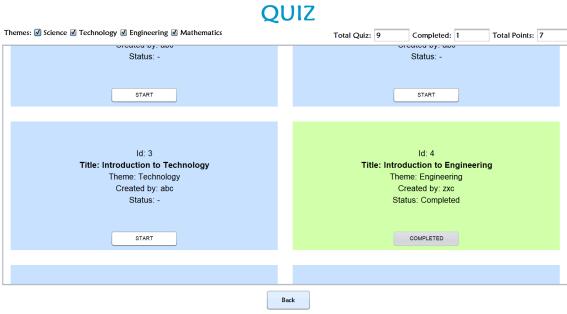
The logic implemented in the system revolves around quiz viewing, filtering, completion tracking, and point allocation. Young students can view quizzes on the Quiz Page and attempt them to earn points. The system supports filtering quizzes based on themes (Science, Technology, Engineering, Mathematics) selected by the student. By default, all themes are selected, showing all quizzes. However, students can choose 1 to all 4 themes as filters, displaying quizzes matching the selected themes. Upon completing a quiz, the system awards 2 marks to the student and marks the quiz as completed in the database. This logic ensures a streamlined and rewarding experience for students interacting with quizzes while maintaining data accuracy and progress tracking.

c. Database:

This part integrates with three primary database tables: 'quiz_create', 'quiz_status', and 'leaderboard'. The 'quiz_create' table stores all created quizzes, including their titles, themes, and content. It is used to display all the quizzes and sorted based on the quiz_id to be more organized. When a new student is registered, triggers within the database automatically associate all quizzes from 'quiz_create' with the student, ensuring comprehensive quiz availability for students. The 'quiz_status' table tracks the completion status of quizzes for each student, preventing duplication of marks by limiting awards to 2 marks per completed quiz. Additionally, the 'leaderboard' table manages students' points, incrementing by 2 points for each completed quiz. This database structure supports smooth data management, efficient awarding of points, and accurate tracking of quiz completion status.

Sample output:

<p>Quiz Page: (Haven't complete any quiz)</p>  <p>Themes: <input checked="" type="checkbox"/> Science <input type="checkbox"/> Technology <input type="checkbox"/> Engineering <input type="checkbox"/> Mathematics</p> <p>QUIZ</p> <p>Total Quiz: 9 Completed: 0 Total Points: 0</p>	<p>For every student, after clicking Access Quiz in Students' drop down in the Main page, it will show all the quizzes created with a big and obvious QUIZ title in the quiz page.</p> <p>Then, students can see the themes selection checkbox on the upper left hand side of the page. All the themes are selected by default, therefore quizzes for every theme will be displayed once students enter this page.</p> <p>At the upper right hand side there is a</p>
--	---

	<p>summary info, containing the number of quizzes, number of completed quizzes and total points of the current student so that the student can easily know their latest points without going back to the leaderboard or their account.</p> <p>For every quiz, it is shown in a box with its respective Id, Quiz Title, Theme, Educator who created it, status of the quiz whether already completed or not and a START button to go to the Quiz Details page to attempt the quizzes.</p> <p>At the bottom of the Quiz page there is a Back button to navigate back to the Main page.</p>
<p>Quiz Details Page:</p> 	<p>After students click the START button, they will enter the quiz details page. It will show the Title of the selected quiz with its theme, description and the content. For the content, we provide the clickable link for attempting the quizzes. After students have completed the quiz, then they can click the Submit button on the bottom of the page. There is also a Back button to navigate back to the Quiz page.</p>
<p>After submitting:</p> 	<p>After submitting the quiz, then it will prompt the completing message with the latest points as upon every completion of the quiz, the student will be awarded 2 points to encourage and increase interests of students in STEM. So it will show the current points after adding 2 points.</p>
<p>Quiz Page: (After some quiz completed)</p> 	<p>After the message, it will go back to the Quiz page, the submitted quiz will be marked as completed which changes the quiz box color to green. While the other quiz box will remain blue. The completed quiz will not be able to START again since the button has become completed and unclickable. This makes sure that students can only attempt the quiz once to ensure only 2 points will be awarded for each</p>

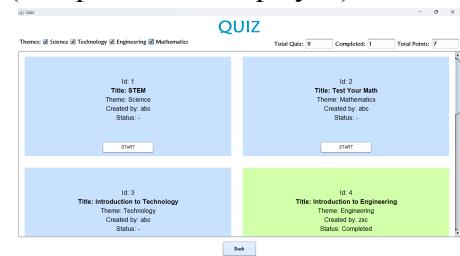
completed quiz. Then, the summary info will also be updated. The completed quiz will increase and the total points is the current points of the student.

Quiz Page: (Theme filtering)

The themes for the quizzes for now will only be Science, Technology, Engineering or Mathematics, and students can use it to filter the quizzes that they want to view. By default, all themes are selected, and all quizzes are displayed. Students are able to select 1 to all 4 themes as the filter.

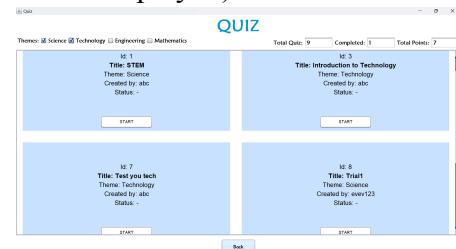
All 4 themes selected (default):

(All quizzes are displayed)



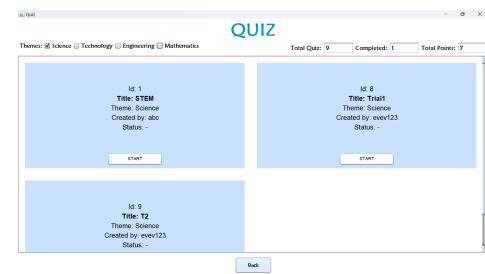
2 themes selected:

(Only Science and Technology theme quizzes are displayed)



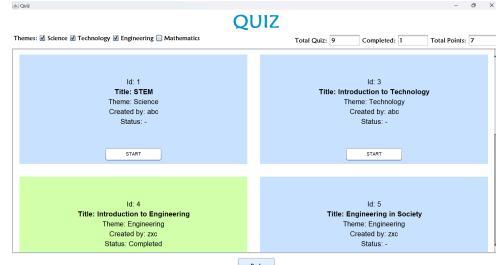
1 theme selected:

(Only Science theme quizzes are displayed)



3 themes selected:

(Only Science, Technology and Engineering theme quizzes are displayed)



Make Bookings

Explanation:

Parents have the ability to book or buy tickets for their children (Young Students) to participate in tours and visits to STEM-related companies, discovery centers, or museums. The destination suggestions are organized in ascending order based on the Euclidean distance between the user's registered coordinates and the coordinates of the target destination.

In this section, various data structures are used to efficiently handle destinations, user coordinates, and booking details within the booking page.

1. Maps:

- **destinations Map:** This map stores destination names as keys and their corresponding coordinates as values. It is used to efficiently associate each destination with its geographical coordinates. The *HashMap* implementation is chosen here for its fast

retrieval time based on destination names.

2. Arrays:

- **Array of Coordinates:** Within the *destinations* map, the coordinates of each destination are stored as an array of doubles (*double[]*). This array structure allows for the compact storage of X and Y coordinates in a single data structure. Each destination's coordinates can be accessed using its name as a key in the map.

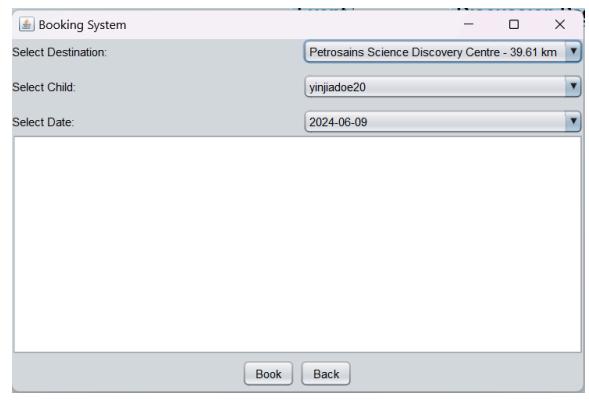
3. Lists:

- **Combo Box Items:** Both the *childrenComboBox* and *dateComboBox* utilize lists to populate their selectable items dynamically.
 - *childrenComboBox*: The list of children associated with the parent's account is retrieved from the database and populated into this combo box.
 - *dateComboBox*: The list of available dates for booking is generated based on the selected child's availability. These dates are loaded into the combo box to allow the user to select a booking date.
- **Formatted Destinations:** In the *getFormattedDestinations()* method, a list (*ArrayList*) of formatted destination strings is created. These strings contain both the destination name and the distance from the user's location, formatted for display purposes. This list is sorted based on the proximity of destinations to the user's coordinates before being returned to populate the destination combo box.

4. Database Interaction:

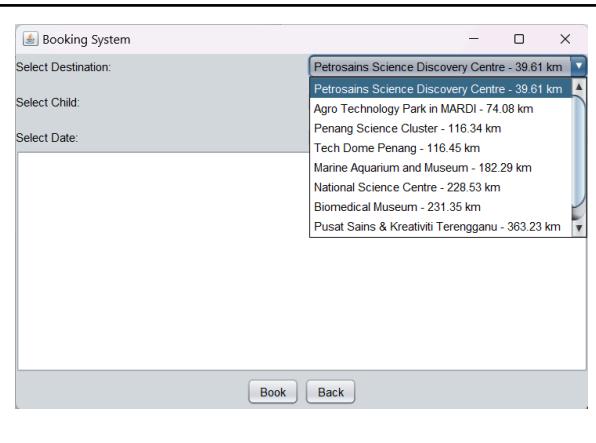
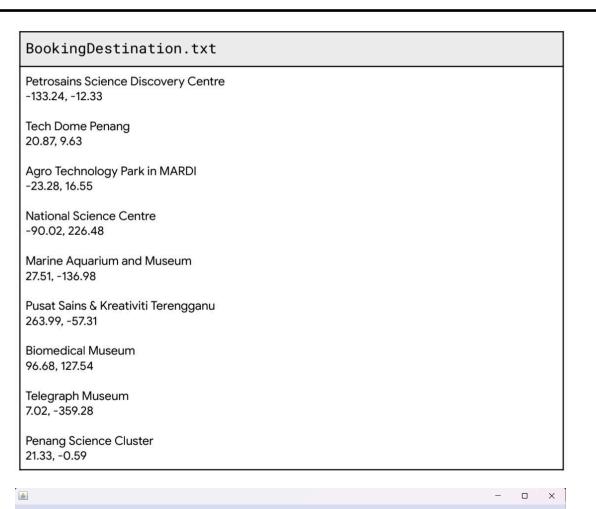
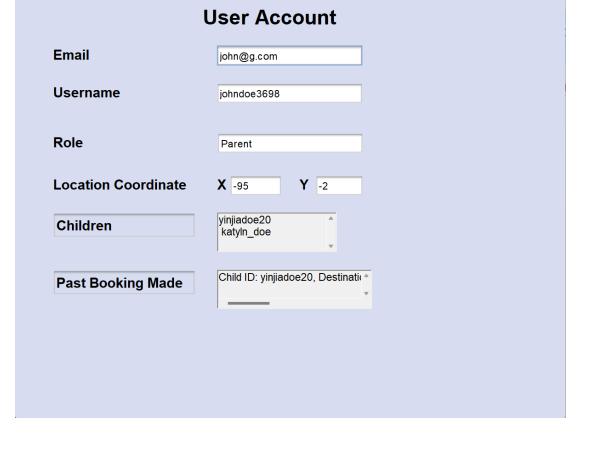
- The class interacts with a database using SQL queries to retrieve and store relevant data for the booking process. This involves querying for user coordinates, checking for event conflicts and existing bookings, retrieving available children and dates, and inserting new bookings into the database. The retrieved data is then utilized to populate the combo boxes and perform booking operations.

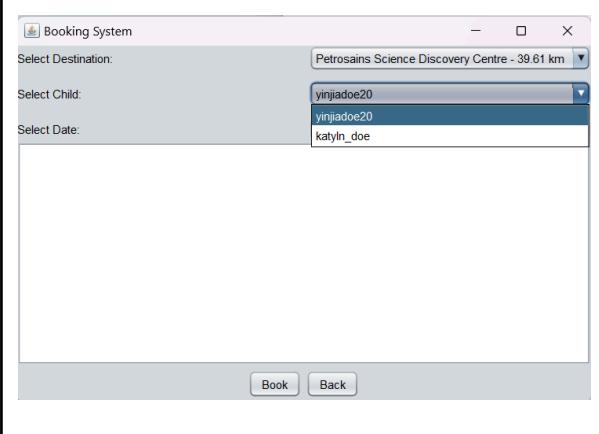
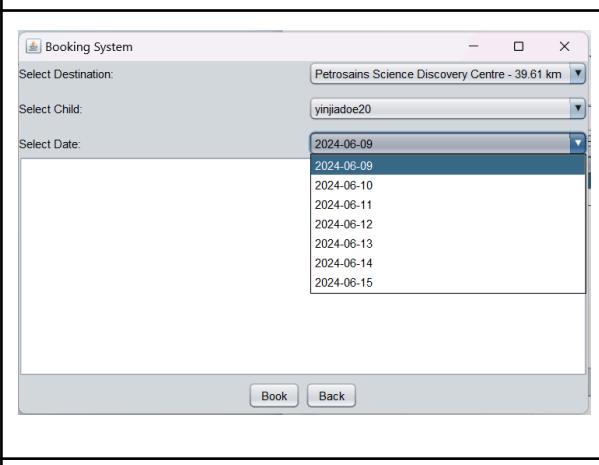
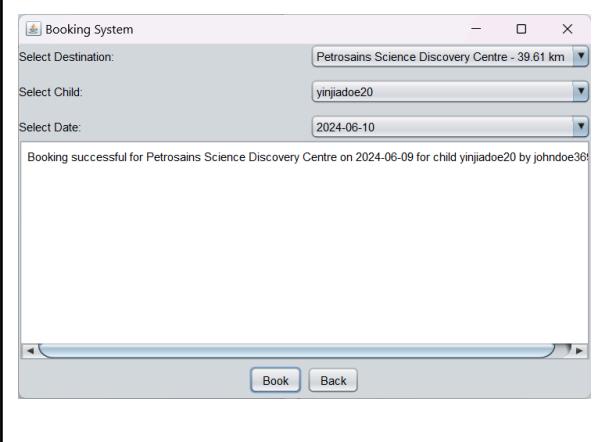
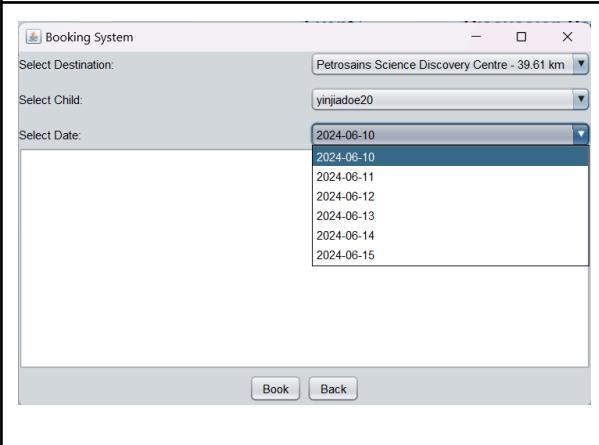
Sample output:

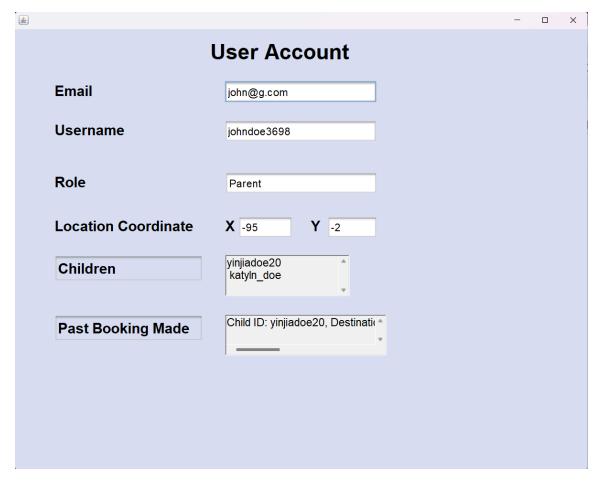


The screenshot shows a window titled "Booking System". It contains three dropdown menus: "Select Destination" (set to "Petrosains Science Discovery Centre - 39.61 km"), "Select Child" (set to "yinjiade20"), and "Select Date" (set to "2024-06-09"). At the bottom are two buttons: "Book" and "Back".

1. On the booking page, parents can select the destination, child, and date.

 <p>The screenshot shows a window titled "Booking System". It has three input fields: "Select Destination:", "Select Child:", and "Select Date:". Below these is a dropdown menu listing various locations with their distances from the user's coordinate:</p> <ul style="list-style-type: none"> Petrosains Science Discovery Centre - 39.61 km Petrosains Science Discovery Centre - 39.61 km Agro Technology Park in MARDI - 74.08 km Penang Science Cluster - 116.34 km Tech Dome Penang - 116.45 km Marine Aquarium and Museum - 182.29 km National Science Centre - 228.53 km Biomedical Museum - 231.35 km Pusat Sains & Kreativiti Terengganu - 363.23 km <p>At the bottom are "Book" and "Back" buttons.</p>	<p>2.</p> <ol style="list-style-type: none"> The destination suggestions are displayed in ascending order of the Euclidean distance between the user's coordinate upon registration and the coordinate of the target destination.
 <p>The screenshot shows a text file named "BookingDestination.txt" with the following content:</p> <pre> Petrosains Science Discovery Centre -133.24, -12.33 Tech Dome Penang 20.87, 9.63 Agro Technology Park in MARDI -23.28, 16.55 National Science Centre -90.02, 226.48 Marine Aquarium and Museum 27.51, -136.98 Pusat Sains & Kreativiti Terengganu 263.99, -57.31 Biomedical Museum 96.68, 127.54 Telegraph Museum 7.02, -359.28 Penang Science Cluster 21.33, -0.59 </pre>	<p>2.</p> <ol style="list-style-type: none"> For example, the distance between user's coordinate and Petrosains Science Discovery Centre is calculate by: $\sqrt{(-133.24 - 95)^2 + (-12.33 - (-2))^2}$ $= 39.61\text{km}$ <ol style="list-style-type: none"> The user coordinate is automatically generated randomly when the account is created. Users can view their coordinates in their profile.
 <p>The screenshot shows a "User Account" window with the following fields:</p> <ul style="list-style-type: none"> Email: john@g.com Username: johndoe3698 Role: Parent Location Coordinate: X: -95, Y: -2 Children: yinjiadoe20, katyln_doe Past Booking Made: Child ID: yinjiadoe20, Destination: [list] 	

	<p>3. Parents can choose one of their children to book a tour or visit.</p>
	<p>4. Parents can choose a date for the booking. The program only displays time slots up to one (1) week from the current day. Assuming the current day is 08/06/2024, here is an example output.</p>
	<p>5. A message indicating that the booking was successful.</p>
	<p>6. The booked date (2024-06-09) will no longer be displayed.</p>

 <p>The screenshot shows a window titled "User Account". It contains fields for "Email" (john@g.com), "Username" (johndoe3698), "Role" (Parent), "Location Coordinate" (X: -95, Y: -2), and a "Children" dropdown menu listing "yinjiadoe20" and "katyln_doe". A "Past Booking Made" section shows a list: "Child ID: yinjiadoe20, Destination: Petrosains Science Discovery Centre - 39.61 km".</p>	<p>8. All bookings made will be updated in the parent's profile.</p>
--	--

Parent-Child Relationship

During the registration process, “Student” must fill in one to two parents whereas “Parent” must fill in their children. The parent child relationship will save to a text file named “ParentChild.txt” by using the “ParentChildTextGenerator”. The format of the file is <parent_username>, <child_username>.

Global Leaderboard

The global leaderboard is a page that shows the ranking of each registered Young Student based on current points.

Explanation:

a. Data Structure:

We use a DefaultTableModel that is a part of Java's Swing framework. In our leaderboard application, we use the DefaultTableModel to hold and manage the data about students' rankings[RANK], usernames[USERNAME] and points[POINTS]. It provides methods to dynamically insert, delete and update rows and columns based on

our data as well as customize the appearance of our table, making it flexible, attractive and keeping our leaderboard up to date.

b. Sorting Logic:

When we bring data from our database into the DefaultTableModel, we arrange it in a particular order, which is highest points first and the current points received taking priority. If two students have the same points. In this case, the student who earned those points earlier will be placed higher up in the leaderboard to maintain fairness.

c. Database:

We use the populateLeaderboard method as our bridge to the database. It connects to the database using SQL queries, which are special commands that help us get and organize data. On the other hand, to make sure every student gets onto the leaderboard, we've set up a trigger. This trigger activates whenever a new student (with the role "student") is added to our system. It automatically puts them into the leaderboard. Moreover, in our database, we use a field called pointLastUpdated. This field helps us track the date and time when a student earned their points. By using this information, we ensure that students who earned their points earlier are ranked higher in the leaderboard when the same points are earned by two different students.

Sample output:

Leaderboard Page:

RANK	USERNAME	POINTS
1	e	24
2	name	22
3	a	19
4	lilia	17
5	hiji	12
6	Adantan09	10
7	xh	9
8	hz	9
9	happy	7
10	yuyu	7
11	test	7
12	ali	7
13	asd	5
14	murid	2
15	ahmedfirdaus07	0
16	Laura_tan	0

Upon opening the leaderboard from the Students' drop down in the Main page, you'll see a "LEADERBOARD" title at the top in gold color. Below, a table indicates each student's Rank, Username, and Points. The usernames include all registered Young Students, and the points indicate their current points.

Each row represents a student. The rows are sorted by points in descending order, with an additional rule: if two students have the same points, the one who achieved them first is ranked higher (first-come, first-serve basis).

The current logged-in student's row is highlighted in yellow for easy identification, while others have a white background.

	A "Back" button at the bottom allows navigation back to the Main page.
--	--

Send & Manage Friend Request

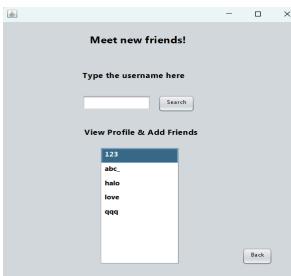
Explanation:

Students can add friends by sending friend requests to other students that are not part of their friend list upon clicking into their profile, they cannot add educator and parent as friends. At the same time, educator and parent cannot add anyone. This is achieved by only displaying the "add friend" button on a student's profile if the user who viewed the profile is also a student. also a studeny. Young Students can also manage the incoming friend requests that they receive. They can either accept the request and add the person into their friend list, or reject the request and remove the request from the request list.

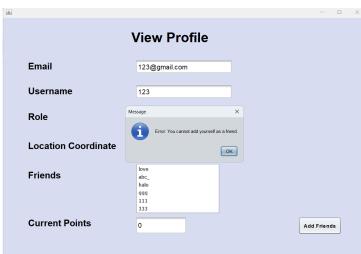
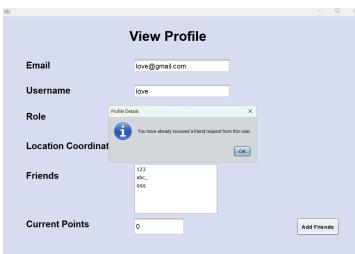
Required: Add friend, friend request list (Accept and decline friend request) and friend list.

Sample output:

Add friends:

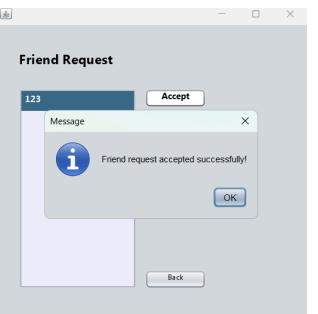
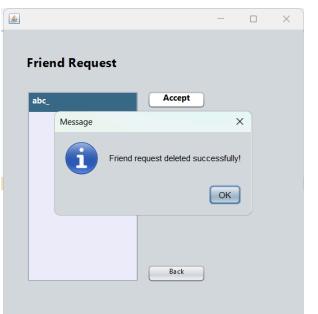
(i) Add friend when searching and viewing the profile in main page	(ii) Add friend by searching or browsing at the student list (can view the details)
	

There are some scenarios when adding friends.

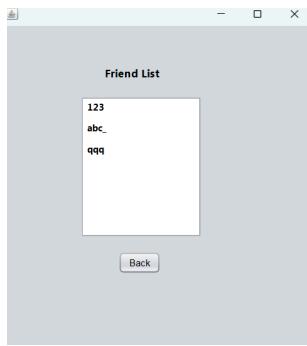
(i) Cannot as himself as friend	(ii) Received friend request before, check the manage friend request.
	

<p>(iii) They are not friends, no friend request sent or friend request received. Hence, the student can add him as a friend.</p>	<p>(iv) Send friend requests before, the student can choose to delete friend requests.</p>
 	 
<p>(v) They are friends, cannot add friends anymore.</p>	

Friend Request List:

Show the request	Can accept request	Can decline request
		

Show the friend list



2.0 Extra Features

In this section, you are required to state, clarify, and explain all extra features proposed to complete the project.

Graphical User Interface

The GUI (Graphical User Interface) of our project is built using Java Swing, a user interface toolkit that allows us to create visually appealing page. With Swing, we design interactive elements like windows, buttons, text fields, and menus that users can interact with. Our design is focused on simplicity and functionality, ensuring that users can easily navigate through different sections of our application. For example, when users log in or register, they are presented with a clean and intuitive interface where they can input their credentials and access various features based on their role. We utilize features like message dialogs to provide feedback to users, enhancing their overall experience. The use of Swing's components and layouts helps us create a cohesive and user-friendly interface that meets the needs of our project.

Database

MySQL is a type of database management system (DBMS) that we're utilizing for our project. This encompasses a wide array of data, such as user credentials (emails, usernames, passwords), designated roles (like Educators, Parents, and Young Students), relationships between parents and children, and details about various activities like events, quizzes, tour bookings, and friend requests. We have 10 different tables for storing different data for different fields. We use MySQL due to its ability to efficiently organize and manage this vast amount of data. It uses structured tables to store information in a logical and easily accessible manner, allowing us to retrieve specific data quickly when needed. For instance, when a user logs in, MySQL helps fetch their profile details, check their role permissions, and display relevant content based on their access rights.

Link for mySQL code:

SQL

Password Hashing

All of the passwords will be encrypted by using bcrypt, which is a password-hashing function. bcrypt works by combining hashing and a technique known as salting, which is specifically developed to make stored passwords more safe.

Hashing: Bcrypt processes a user's password using a sophisticated mathematical function. This function converts the password to a fixed-length string of characters that appear random and meaningless. The hashed value is what is kept in the database, not the original password. Because the hashing function is one-way, reversing the hash will not produce the original password.

Salting: To improve security, bcrypt incorporates a random number called a salt. This salt is unique to each password and is attached to it before hashing. The combined value (password + salt) is then passed to the hashing function.

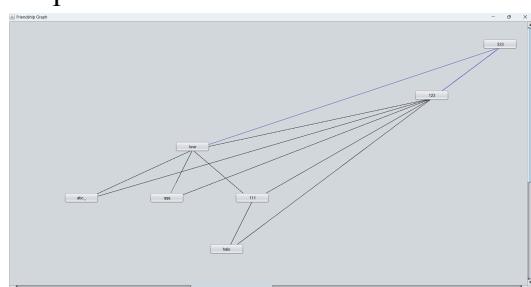
Friendship Found: Where Connection Begin

Explanation:

Student friend list is shown in a graph using visualization and data structure like graph. The student is able to view the connection as 3 layers. First is himself, second is his friends and third is his friends' friends. By doing so, he has a clearer image on who he does not add as friend yet and can add them as friend to expand his social circles. The add friend method will be the same as mentioned in basic functionality, it will check whether they are friends or not, receive friend requests or sent friend request.

Sample output:

Graph visualization:



 Himself.	 His friend.	 His friend's friend where he can send the friend request.

3.0 Contribution Acknowledgement

State the contribution for each member in this section. You are encouraged to specify the contributions in tabular form.

Task			
No	Description	PIC	Contributions
1	Create User Class	Chan Lin Na	<i>Develop the User class with required fields (Email, Username, Password, Role, Parent(s), Children, Location Coordinate, Current Points).</i>
2	Login / Registration Page	Chan Lin Na	<i>Implement sign-up and login pages for users to register and log into their accounts. Automatically generate Location Coordinates.</i>
3	Data Storage	All	<i>Implement data storage to save parent-child relationship in ParentChild.txt. Ensure data persists after program termination.</i>
4	View Profile Page	Chan Lin Na	<i>Create a profile page displaying user data (excluding password). Display additional data based on user role (Educators, Parents, Young Students).</i>
5	Access Management	Chan Lin Na	<i>Implement role-based access management to restrict access to specific pages based on user role.</i>
6	Create Event & Quiz Pages	Boo Ee Vone	<i>Enable Educators to create events and quizzes with specified fields. Populate with at least 10 events and 10 quizzes.</i>
7	View Event Page	Ang Li Jia	<i>Display live events and the closest 3 upcoming events. Allow Young Students to register for events and gain points. Handle event time clashes.</i>

8	<i>Attempt Quiz Page</i>	<i>Ang Li Jia</i>	<i>Enable Young Students to view and attempt quizzes filtered by theme. Award points for completed quizzes.</i>
9	<i>Booking Page</i>	<i>Tan Chien Ling</i>	<i>Allow Parents to book visits to STEM-related destinations for their children. Display destinations sorted by Euclidean distance. Handle booking time clashes.</i>
10	<i>Parent-Child Relationship</i>	<i>Tan Chien Ling</i>	<i>Implement the parent-child relationship in user profiles based on provided data. Display parents for Young Students and children for Parents.</i>
11	<i>Global Leaderboard</i>	<i>Ang Li Jia</i>	<i>Create a leaderboard displaying Young Students sorted by current points in descending order. Handle tie cases based on pointLastUpdated state.</i>
12	<i>Send & Manage Friend Requests</i>	<i>Chua Hui Ying Nicole</i>	<i>Enable Young Students to send, accept, and reject friend requests. Display friend lists and manage incoming friend requests.</i>
13	<i>Develop Graphical User Interface (Extra Feature)</i>	<i>All</i>	<i>Create a user-friendly GUI for the application using Java Swing.</i>
14	<i>Implement Database Integration (Extra Feature)</i>	<i>All</i>	<i>Integrate a database, MySQL, for persistent data storage.</i>

15	<i>Implement Password Hashing (Extra Feature)</i>	<i>Chan Lin</i>	<i>Implement password hashing or encryption to ensure user data security.</i>
16	<i>Visualize Friend Connections (Extra Feature)</i>	<i>Chua Hui Ying Nicole</i>	<i>Implement graphical visualization of friend connections among Young Students to help discover new friends.</i>

4.0 Conclusion

Provide a concise overview of the project, highlighting the methodology employed and the knowledge gained and implemented.

To keep up with the rapidly-advancing world, the importance of exposure to the concept and importance of STEM (Science, Technology, Engineering, and Mathematics) knowledge to the young generation is essential. To achieve this, we create a simple gamification, where we focus on making the learning process fascinating by integrating quizzes and event registrations related to STEM. This approach not only engages students but also connects them to three critical Sustainable Development Goals (SDGs): SDG 4 (Quality Education), SDG 9 (Industry, Innovation, and Infrastructure), and SDG 17 (Partnerships for the Goals).

The methodology of the project involves defining clear user roles and interactions: young students are targeted for quizzes and event participation, earning points to foster engagement; parents manage their children's STEM activities and make bookings for tours and visits; educators create content, quizzes, and events to enrich the platform. The features include user accounts and access management, ensuring each user has a personalized account with role-specific access. A quizzes and points system incentivizes continuous learning by allowing students to earn points for answering STEM-related questions. Event registration enables students to sign up for tours to STEM-related companies and museums, enhancing experiential learning. A booking system allows parents to schedule visits for their children to various STEM sites, while a leaderboard and social features track student points and manage friend requests to encourage social learning.

In implementing these features, specific data structures are employed to optimize performance and functionality. For example, handling friend relationships is managed using a graph data structure, where nodes represent users and edges represent friendships. This structure efficiently supports operations like finding mutual friends or suggesting new connections.

In terms of data handling, the project ensures secure storage of user data and activities, guaranteeing persistent data across sessions. Location-based features automatically generate user coordinates to provide proximity-based event and booking suggestions. The knowledge gained and implemented through this project highlights the integration of STEM knowledge, where quizzes and events deepen students' understanding of STEM concepts and relate them directly to real-world applications.

This platform fosters critical thinking, problem-solving, and innovation, aligning with SDG 4 by providing quality education through engaging methods. For SDG 9, it links STEM education to industrial innovation via industry tours and technological insights, inspiring students to future developments. SDG 17 is addressed by fostering partnerships among educators, parents, and STEM organizations, promoting collaboration and shared knowledge. This gamified approach makes learning enjoyable and instills a sense of purpose, guiding students toward sustainable development goals.