

Project Report
Student Management System Maintenance and Enhancement
For
Software Maintenance and Configuration (SE3216)

Submitted by

Aghared Kadu	443007057
Retal Malki	44410160
Jana Alyamani	444006930
Dana Fawaz	444003813
Jana Alotaibi	444007448

Supervisor:

Dr. Enas Bugis



DEPARTMENT OF SOFTWARE ENGINEERING
COLLEGE OF INFORMATION SYSTEM
UMM AL-QURA UNIVERSITY

FEB 5, 2025

Documentation:

1. Explain how the bug was fixed and how it was resolved.

Problem: Duplicate Student IDs

In the original code, there was a problem where the program allowed students to have the same ID. This could cause confusion when searching, updating, or removing a student because there would be multiple students with the same ID.

How Did We Fix It?

We fixed the issue by adding a check before adding a new student. Now, the program checks if the ID already exists and stops the user from entering a duplicate.

Instead of writing the check inside `addStudent()`, we created a separate method called `isDuplicateID()`. This method helps keep our code organized and easy to read.

Steps to Fix the Bug:

1. Created a helper method (`isDuplicateID`)

- This method goes through the student list and checks if an ID is already used.
- If it finds a match, it returns `true` (meaning the ID exists).
- If there is no match, it returns `false` (meaning the ID is unique).

2. Updated `addStudent` to use `isDuplicateID`

- Before adding a student, the program calls `isDuplicateID` to check if the ID is taken.
- If the ID exists, the program shows an error message and does not add the student.
- If the ID is unique, the program adds the student normally.

2. Describe the newly added feature and its implementation.

Problem: Remove Student by ID

In the original code, there was no feature to remove a student by their ID. This made it difficult to manage the student list, especially when there was a need to delete outdated or incorrect records.

How Did We Fix It?

We implemented a solution by adding a new feature to remove a student using their unique ID. This was achieved by creating a helper method to check if the student exists and a dedicated method to handle the removal process. Additionally, we updated the main menu to include this new functionality.

Steps to Fix the Bug:

1. Created a Helper Method (studentExists):

- This method iterates through the student list and checks if a student with the given ID exists.
- If a match is found, it returns true (indicating the student exists).
- If no match is found, it returns false.

2. Added a New Option in the Main Menu:

- Updated the main menu to include a new option: "4. Remove Student by ID."
- This allows users to access the removal feature easily.

3. Implemented the removeStudentById Method:

- The method prompts the user to enter the student ID they want to remove.
- It calls the studentExists method to check if the student exists:
- If the student exists:
- The method removes the student from the list using removeIf.
- A success message is displayed to confirm the removal.

- If the student does not exist:
- An error message is displayed, informing the user that the student was not found.

4. Provided User Feedback:

- Displayed clear and concise messages to inform the user whether the operation was successful or if the student was not found.

3. Detail the refactoring changes made and their impact.

1. Overview of Refactoring

The **Student Management System** has been refactored to improve:

- Code readability
- Maintainability
- Scalability
- Best coding practices

2. Refactored `displayStudents()` → `showStudentList()`

Change:

- The method `displayStudents()` was renamed to `showStudentList()` for better readability.
- Improved text alignment using `printf()` to display student details in a table-like format.

Impact:

- Easier to read student details.
- Better formatting for large datasets.
- Aligns with best practices in method naming (verb + noun structure).

3. Renamed Methods for Better Clarity

Change:

Old Name	New Name (Best Practice)
addStudent()	addNewStudent()
displayStudents()	showStudentList()
searchStudentById()	findStudentById()
removeStudentById()	deleteStudentById()

Impact:

- Methods now clearly describe their functionality.
- Easier for developers to understand and use.
- Follows standard coding conventions.

4. Added Meaningful Comments and Documentation

Change:

- Added Doxygen-style comments for all classes and methods.
- Used @brief, @details, @param, and @return for better documentation generation.

Impact:

- Developers can quickly understand what each function does.
- Easier to generate automated documentation using Doxygen.
- Improves maintainability and readability.

5. Code Modularization – Using Multiple Classes

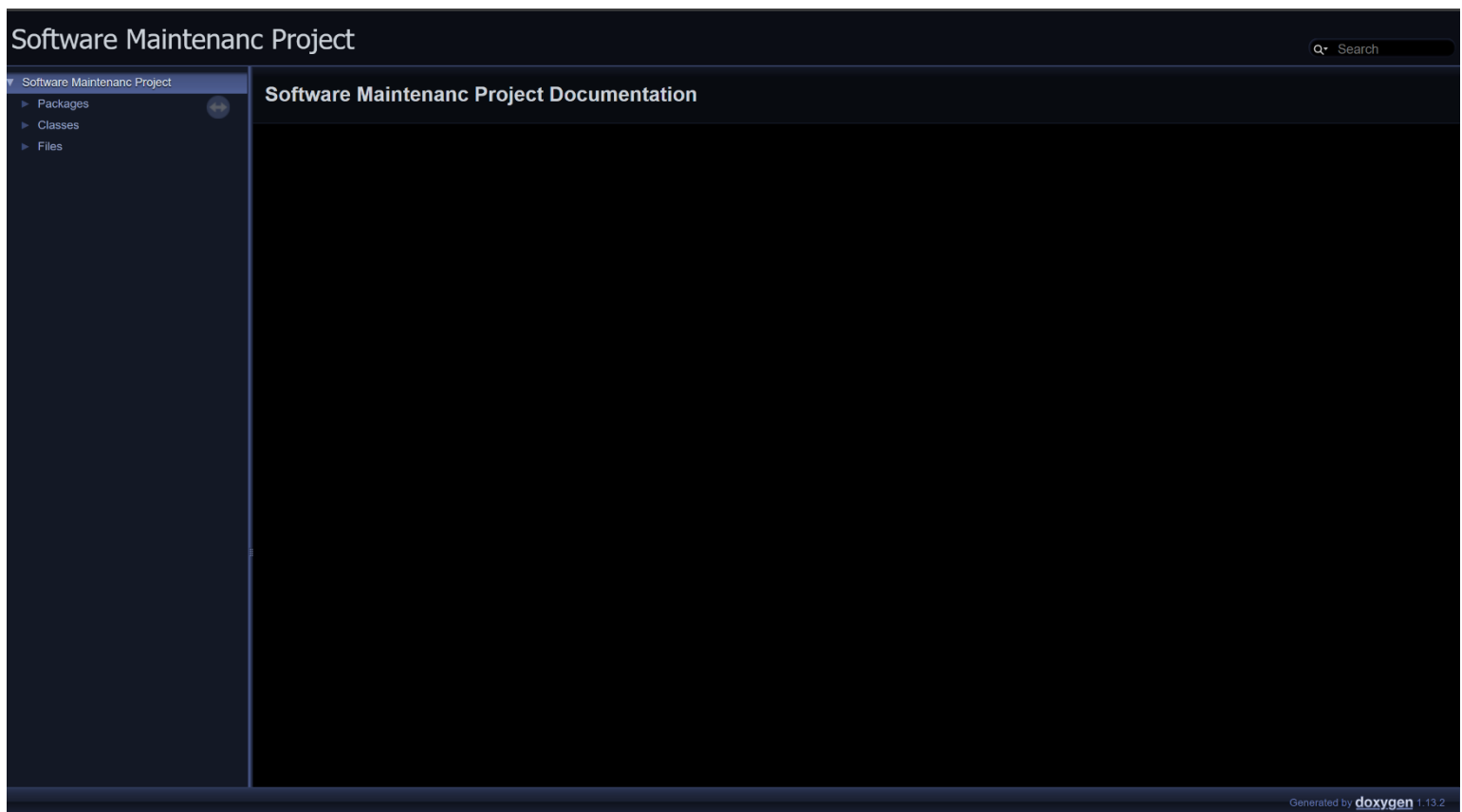
Change:

- Separation of concerns:
 - Created Student.java → Represents a student entity
 - Created StudentManager.java → Handles student operations (Add, Search, Remove)
 - StudentManagementSystem.java now only handles UI and user input

Impact:

- Better organization – each class has a single responsibility.
- Easier to scale – Can now add features like file/database storage.
- Encapsulation – Student logic is separate from UI logic.

Documentation on Doxygen



Software Maintenan Project

Q Search

Software Maintenan Project

Package List

softwaremaintenancproject

Classes

Files

Package softwaremaintenancproject

Classes

class Student

Represents a student with ID, name, and age. More...

class StudentManagementSystem

Manages the UI and user input for the student system. More...

class StudentManager

Handles student operations (add, remove, search, display). More...

softwaremaintenancproject

Generated by doxygen 1.13.2

Software Maintenan Project

Q Search

Software Maintenan Project

Package List

Classes

Class List

Class Index

Class Members

Files

Class List

Here are the classes, structs, unions and interfaces with brief descriptions:

softwaremaintenancproject

Student

Represents a student with ID, name, and age

StudentManagementSystem

Manages the UI and user input for the student system

StudentManager

Handles student operations (add, remove, search, display)

Generated by doxygen 1.13.2

Software Maintenanc Project

Q Search

Software Maintenanc Project

Package

Classes

Class List

softwaremaintenancproject

Student

StudentManagementSystem

StudentManager

Class Index

Class Members

Files

softwaremaintenancproject.Student Class Reference

Public Member Functions | List of all members

Represents a student with ID, name, and age. More...

Public Member Functions

Student (int id, String name, int age)

< Student's age

void displayStudent ()

Displays student details.

Detailed Description

Represents a student with ID, name, and age.

Definition at line 12 of file Student.java.

Constructor & Destructor Documentation

Student()

softwaremaintenancproject.Student.Student (int id, String name, int age)

< Student's age

Constructs a new Student object.

softwaremaintenancproject

Student

Generated by doxygen 1.13.2

Software Maintenanc Project

Q Search

Software Maintenanc Project

Package

Classes

Class List

softwaremaintenancproject

Student

displayStudent

StudentManagementSystem

StudentManager

Class Index

Class Members

Files

Student()

softwaremaintenancproject.Student.Student (int id, String name, int age)

< Student's age

Constructs a new Student object.

Parameters

id

The student's unique ID.

name

The student's name.

age

The student's age.

Definition at line 23 of file Student.java.

Member Function Documentation

displayStudent()

void softwaremaintenancproject.Student.displayStudent ()

Displays student details.

Definition at line 32 of file Student.java.

The documentation for this class was generated from the following file:

src/softwaremaintenancproject/Student.java

softwaremaintenancproject

Student

Generated by doxygen 1.13.2

Software Maintenan Project

Q Search

Software Maintenan Project

Package

Classes

Class List

softwaremaintenancproject

Student

StudentManagementSystem

StudentManager

Class Index

Class Members

Files

softwaremaintenancproject.StudentManagementSystem

Static Public Member Functions | List of all members

Manages the UI and user input for the student system. More...

Static Public Member Functions

static void main (String[] args)

Detailed Description

Manages the UI and user input for the student system.
Definition at line 14 of file StudentManagementSystem.java.

Member Function Documentation

main()

static void softwaremaintenancproject.StudentManagementSystem.main (String[] args)

Definition at line 15 of file StudentManagementSystem.java.

The documentation for this class was generated from the following file:

- src/softwaremaintenancproject/StudentManagementSystem.java

softwaremaintenancproject

StudentManagementSystem

Generated by doxygen 1.13.2

Software Maintenan Project

Q Search

Software Maintenan Project

Package

Classes

Class List

softwaremaintenancproject

Student

StudentManagementSystem

StudentManager

Class Index

Class Members

Files

softwaremaintenancproject.StudentManager

Public Member Functions | List of all members

Handles student operations (add, remove, search, display). More...

Public Member Functions

boolean isDuplicateID (int id)
Checks if a student ID already exists.

boolean addNewStudent (Student student)
Adds a new student to the system.

void showStudentList ()
Displays the list of students in a formatted table.

Student findStudentById (int id)
Searches for a student by ID.

boolean deleteStudentById (int id)
Removes a student from the system by ID.

Detailed Description

Handles student operations (add, remove, search, display).
Definition at line 14 of file StudentManager.java.

Member Function Documentation

addNewStudent()

softwaremaintenancproject

StudentManager

Generated by doxygen 1.13.2

Software Maintenanc Project

Q Search

Software Maintenanc Project

Packages

Classes

Class List

softwaremaintenancproject

Student

StudentManagementSystem

StudentManager

addNewStudent

deleteStudentById

findStudentById

isDuplicateID

showStudentList

Class Index

Class Members

Files

◆ addNewStudent()

boolean softwaremaintenancproject.StudentManager.addNewStudent (Student student)

Adds a new student to the system.

Ensures unique ID validation before adding the student.

Parameters

student Student object to be added.

Returns

True if added successfully, false if ID is exists.

Definition at line 32 of file StudentManager.java.

◆ deleteStudentById()

boolean softwaremaintenancproject.StudentManager.deleteStudentById (int id)

Removes a student from the system by ID.

Parameters

id The student ID to remove.

Returns

True if removed successfully, false if not found.

Definition at line 68 of file StudentManager.java.

◆ findStudentById()

softwaremaintenancproject

StudentManager

Generated by doxygen 1.13.2

Software Maintenanc Project

Q Search

Software Maintenanc Project

Packages

Classes

Class List

softwaremaintenancproject

Student

StudentManagementSystem

StudentManager

addNewStudent

deleteStudentById

findStudentById

isDuplicateID

showStudentList

Class Index

Class Members

Files

◆ findStudentById()

Student softwaremaintenancproject.StudentManager.findStudentById (int id)

Searches for a student by ID.

Parameters

id The student ID to search.

Returns

The Student object if found, otherwise null.

Definition at line 59 of file StudentManager.java.

◆ isDuplicateID()

boolean softwaremaintenancproject.StudentManager.isDuplicateID (int id)

Checks if a student ID already exists.

Parameters

id The student ID to check.

Returns

True if the ID exists, false otherwise.

Definition at line 22 of file StudentManager.java.

◆ showStudentList()

void softwaremaintenancproject.StudentManager.showStudentList ()

softwaremaintenancproject

StudentManager

Generated by doxygen 1.13.2

Software Maintenanc Project

Q Search

Software Maintenanc Project

➤ Packages

▼ Classes

➤ Class List

Class Index

▼ Class Members

All

Functions

➤ Files

Here is a list of all class members with links to the classes they belong to:

- `addNewStudent()` : `softwaremaintenancproject.StudentManager`
- `deleteStudentById()` : `softwaremaintenancproject.StudentManager`
- `displayStudent()` : `softwaremaintenancproject.Student`
- `findStudentById()` : `softwaremaintenancproject.StudentManager`
- `isDuplicateID()` : `softwaremaintenancproject.StudentManager`
- `main()` : `softwaremaintenancproject.StudentManagementSystem`
- `showStudentList()` : `softwaremaintenancproject.StudentManager`
- `Student()` : `softwaremaintenancproject.Student`

Generated by **doxygen** 1.13.2

Software Maintenanc Project

Q Search

Software Maintenanc Project

➤ Packages

▼ Classes

▼ Files

➤ File List

File List

Here is a list of all files with brief descriptions:

[detail level 1 2 3]

▼ src

▼ softwaremaintenancproject

📄 Student.java

📄 StudentManagementSystem.java

📄 StudentManager.java

Generated by **doxygen** 1.13.2

Testing Results:

Adding students (including handling duplicates).

```
Output - SoftwareMaintenanceProject (run) x
run:
--- Student Management System ---
1. Add Student
2. Display Students
3. Search Student by ID
4. Remove Student by ID
5. Exit
Choose an option: 1
Enter Student ID: 4
Enter Student Name: Aghared
Enter Student Age: 22
Student added successfully!

--- Student Management System ---
1. Add Student
2. Display Students
3. Search Student by ID
4. Remove Student by ID
5. Exit
Choose an option: 1
Enter Student ID: 4
Enter Student Name: Jana
Enter Student Age: 21
Error: Student ID already exists.

--- Student Management System ---
1. Add Student
2. Display Students
3. Search Student by ID
4. Remove Student by ID
5. Exit
Choose an option:
```

Removing students.

```
--- Student Management System ---
1. Add Student
2. Display Students
3. Search Student by ID
4. Remove Student by ID
5. Exit
Choose an option: 4
Enter Student ID to remove: 4
Student removed successfully.

--- Student Management System ---
1. Add Student
2. Display Students
3. Search Student by ID
4. Remove Student by ID
5. Exit
Choose an option:
```

Displaying the updated student list.

```
--- Student Management System ---
1. Add Student
2. Display Students
3. Search Student by ID
4. Remove Student by ID
5. Exit
Choose an option: 2
No students found.

--- Student Management System ---
1. Add Student
2. Display Students
3. Search Student by ID
4. Remove Student by ID
5. Exit
Choose an option:
```

After adding some students:

```
--- Student Management System ---
1. Add Student
2. Display Students
3. Search Student by ID
4. Remove Student by ID
5. Exit
Choose an option: 2

--- Student List ---
ID          Name          Age
-----
2           Aghared           22
3           Retal             20
1           Jana              21
4           Dana              21
```

UML in Understand

