Logo: Cardiff Metropolitain University

School of Technologies

### Assessment

### Brief

|  |  |
| --- | --- |
| Module Code | Module Title |
| CSE4006 | Object Oriented Programming |
| Academic Year | Semester |
| 2024 | 2 |
| Module Leader email | |
| upeka@icbtcampus.edu.lk | |

Content

[Assessment Details 2](#_Toc67665314)

[Submission Details 3](#_Toc67665315)

[Assessment Criteria 4](#_Toc67665316)

[Further Information 6](#_Toc67665317)

[Who can answer questions about my assessment? 6](#_Toc67665318)

[Referencing 6](#_Toc67665319)

[Submission problems 6](#_Toc67665320)

[Unfair academic practice 7](#_Toc67665321)

[How is my work graded? 7](#_Toc67665322)

# Assessment Details

|  |  |  |
| --- | --- | --- |
| Assessment title | Abr. | Weighting |
| **Connect Me** | WRIT1 | 100% |
| Pass marks are 40% for undergraduate work and 50% for postgraduate work unless stated otherwise. | | |

|  |  |
| --- | --- |
| Task/assessment brief: | |
| **“Connect Me”**is a Mobile service company, handling their customer care employee details manually. They want to automate the process so that the HR manager and the Assistant can manage employee details easily. They have different departments (Operations, Quality assurance, Training, etc.) with different types of designations (Supervisor, Team captain, Quality Manager, Quality Monitor, Training Manager, etc.)  **User levels and functionalities are follows**  **HR Manager**   1. Add new departments and designations 2. Add new Employees and allocate them to available departments and designations 3. Search Employee details based on department, designations and name etc. 4. Create a new account (HR Assistant)   **HR Assistant**   1. Search Employee details based on department and name etc.   ***You are required to apply OOP concepts for the above scenario. Data need to be saved and retrieved from a File***  **Part A: Report**  **Task 1**. Provide design solution (UML diagrams) for the above mention Scenario. Provide clear explanation for all the diagrams mention below. (Provide assumption if necessary) **(30 marks) (LO2)**   * Use case Diagram * Class Diagram * Sequence Diagram   **Task 2:** Develop suitable system for the above scenario based on the design. Required to use Object Oriented concepts (Object, Class, Abstraction, Inheritance, Encapsulation and Polymorphism) for the development. Document the main functionalities and Object-Oriented concepts applied with proper explanation and source code.  **(Marks 20) (LO1, LO3)**  **Task 3:** Provide a user manual for the developed solution **(Marks 10) (LO3)**  **Guidelines for the report format**  Paper A4  Margins 1.5” left, 1” right, top and bottom  Page numbers – bottom, right  Line spacing 1.5  Font  Headings 14pt, Bold  Normal 12pt  Font face- Times New Roman  **Part B: Demonstration**  **Task 4:** System demonstration. System should work according to the expected functionalities. Should be able to demonstrate Object Oriented concepts (Object, Class, Abstraction, Inheritance, Encapsulation, and Polymorphism) applied to the given scenario**. (Marks 40) (LO1, LO3)** | |
| Word count (or equivalent): | 3000 |
| This a reflection of the effort required for the assessment. Word counts will normally include source code, any text, tables, calculations, figures, subtitles and citations. Reference lists and contents of appendices are excluded from the word count. Contents of appendices are not usually considered when determining your final assessment grade. | |

|  |
| --- |
| Academic or technical terms explained: |
|  |

# Submission Details

|  |  |  |  |
| --- | --- | --- | --- |
| Submission Deadline: | This will be provided on the Moodle submission point. | Estimated Feedback  Return Date | This will normally be 20 working days after initial submission. |
| Submission  Time: | By 2.00pm on the deadline day. |  | |
| Moodle/Turnitin: | **Any assessments submitted after the deadline will not be marked and will be recorded as a non-attempt unless you have had an extension request agreed or have approved mitigating circumstances. See the School Moodle pages for more information on extensions and mitigating circumstances.** | | |
| File Format: | The assessment must be submitted as a pdf document (save the document as a pdf in your software) and submit through the Turnitin submission point in Moodle.  **Your assessment should be titled with your:**  **student ID number, module code and assessment ID,**  **e.g. st12345678 CSE5013 WRIT1** | | |
| Feedback | Feedback for the assessment will be provided electronically via Moodle. Feedback will be provided with comments on your strengths and the areas which you can improve. View the [guidance](https://learn.cardiffmet.ac.uk/mod/glossary/showentry.php?courseid=8107&eid=9581&displayformat=dictionary) on how to access your feedback.  All marks are provisional and are subject to [quality assurance processes](https://outlookuwicac.sharepoint.com/:b:/s/QED/Ec3kYQQeEHdKrCbo_tJnr2kBomIiiLINmPebUgvTUljq9Q?e=a0G2z5) and confirmation at the programme Examination Board. | | |

# Assessment Criteria

|  |
| --- |
| Learning outcomes assessed |
| LO1: Explain the fundamentals of Object-Oriented Programming concepts  LO2: Design Object-Oriented based applications  LO3: Develop Object-Oriented applications |
| Other skills/attributes developed  This includes elements of the Cardiff Met EDGE (Ethical, Digital, Global and Entrepreneurial skills) and other attributes developed in students through the completion of the module and assessment. These will also be highlighted in the module guidance, which should be read by all students completing the module. Assessments are not just a way of auditing student knowledge. They are a process which provides additional learning and development through the preparation for and completion of the assessment. |
| |  |  | | --- | --- | | ETHICAL | Consider the impact of the code: Think about potential biases in algorithms or how your code might be misused.  Write secure code: Prevent data breaches and ensure user privacy.  Document your code clearly: This helps others understand and maintain it ethically. | | DIGITAL | Use OOP to design modular and reusable code: This promotes efficient development and information literacy.  Leverage OOP to build scalable applications: This allows your code to adapt to changing needs, demonstrating problem-solving with technology. | | GLOBAL | Develop software that can be localized for different cultures: Use inheritance and polymorphism in OOP to create a foundation that adapts to different languages or user interfaces. | | ENTREPRENEURIAL | Design OOP structures that are flexible and adaptable: This allows for future innovation and opportunity recognition.  OOP promotes code reusability: This saves resources and allows for quicker development cycles, a key aspect of entrepreneurship | |

|  |
| --- |
| Marking/Assessment Criteria |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Task** | **Poor**  **< 40** | **Satisfactory**  **40 - 55** | **Good**  **55 -69** | **Excellent**  **70 -100** |
| Task 1 | Poor use of Object-Oriented Design Methodology  Use case Diagram   * Poor or no identification of use cases * Poor or no Identification of Actors and associations   Class Diagram   * Poor or no identification of associated methods, with correct signatures and attributes in each class * Poor or no identification of relationships   Sequence Diagram   * Poor or no identification of set of use cases as sequence diagrams. | Proper use of Object-Oriented Design Methodology  Use case Diagram   * Has identify few of correct use cases * Has identify few correct Actors and associations   Class Diagram   * Identification of associated methods, with correct signatures and attributes in each class * Correct identification of relationships   Sequence Diagram   * Has identified set of use cases (about 3) as sequence diagrams. | Use case Diagram   * Accurate use of <<include>> <<extend>> stereo types in use case diagram   Class Diagram   * Clear identification of private, public access modifiers & it is visible in the class diagram   Sequence Diagram   * Appropriate use of lifelines, messages and objects in proposed sequence diagrams   Correct use of UML notations with minor mistakes  Evaluation   * Student has given basic description about the design and given a reasonable justification * Effective judgements have been made about the content and levels of information to be included | Excellent Design   * Highly detailed diagram * Use of OO concepts clearly visible * Backed by relevant assumptions * Excellent use of UML notation   Evaluation   * Good justification of the design * Judge validity of results * Use critical reflection to evaluate the work and justify with valid explanations   Fluency (Of design)  Evidence of critical analysis on different perspectives covering how all UML diagrams support in designing |
| Task 2 | No proper standard coding  Insufficient code evidence with no or poor explanation  Poor use of Object-oriented concepts no or poor explanation  Poor implementation of the design | Proper standard coding  Sufficient code evidence with proper explanation  Basic use of Object-oriented concepts (at least 3 concepts) with basic explanation  Proper implementation of the design with minor mistakes | Proper use of Object-oriented concepts (at least 4 concepts) with proper explanation  Proper implementation of the design with no mistakes | Excellent use of Object-oriented concepts (all the concepts) with proper explanation |
| Task 3 | Poor structure no clear explanation and screenshots  Content is not specific and sufficient | Has provide basic explanation with screenshots  Content is specific and sufficient | Has provide average explanation with screenshots | Excellent, clear and descriptive explanation with screenshots |
| Task 4 | System with syntax and logical errors  Poor or not able to explain and demonstrate implementation of the design Poor or not able to explain and demonstrate Object oriented concepts used in the system | Error free system  Partially able explain, demonstrate and justify the implementation of the design Partially able to explain and demonstrate Object oriented concepts used in the system | Able to explain, demonstrate and justify the implementation of the design  Able to explain and demonstrate Object oriented concepts used in the system | Has demonstrate excellent level of understanding of Object-oriented concepts |

# Further Information

## Who can answer questions about my assessment?

Questions about the assessment should be directed to the staff member who has set the task/assessment brief. This will usually be the Module Leader. They will be happy to answer any queries you have.

Staff members can often provide feedback on an assignment plan but cannot review any drafts of your work prior to submission. The only exception to this rule is for Dissertation Supervisors to provide feedback on a draft of your dissertation.

## Referencing and independent learning

Please ensure you reference a range of credible sources, with due attention to the academic literature in the area. The time spent on research and reading from good quality sources will be reflected in the quality of your submitted work.

Remember that what you get out of university depends on what you put in. Your teaching sessions typically represent between 10% and 30% of the time you are expected to study for your degree. A 20-credit module represents 200 hours of study time. The rest of your time should be taken up by self-directed study.

Unless stated otherwise you must use the **HARVARD** referencing system. Further guidance on referencing can be found in the Study Smart area on Moodle and at [www.citethemrightonline.com](http://www.citethemrightonline.com) (use your university login details to access the site). Correct referencing is an easy way to improve your marks and essential in achieving higher grades on most assessments.

## Technical submission problems

It is strongly advised that you submit your work at least 24 hours before the deadline to allow time to resolve any last minute problems you might have. If you are having issues with IT or Turnitin you should contact the IT Helpdesk on (+44) 2920 417000. You may require evidence of the Helpdesk call if you are trying to demonstrate that a fault with Moodle or Turnitin was the cause of a late submission.

## Extensions and mitigating circumstances

Short extensions on assessment deadlines can be requested in specific circumstances. If you are encountering particular hardship which has been affecting your studies, then you may be able to apply for mitigating circumstances. This can give the teachers on your programme more scope to adapt the assessment requirements to support your needs. Extensions and mitigating circumstances policies and procedures are regularly updated. You should refer to your degree programme or school Moodle pages for information on extensions and mitigating circumstances.

## Unfair academic practice

Cardiff Met takes issues of unfair practice **extremely seriously.** The University has procedures and penalties for dealing with unfair academic practice. These are explained in full in the University's Unfair Practice regulations and procedures under [Volume 1, Section 8](https://www.cardiffmet.ac.uk/registry/academichandbook/Pages/Ah1_08.aspx) of the Academic Handbook. The Module Leader reserves the right to interview students regarding any aspect of their work submitted for assessment.

Types of Unfair Practice, include:

**Plagiarism,** which can be defined as using without acknowledgement another person’s words or ideas and submitting them for assessment as though it were one’s own work, for instance by copying, translating from one language to another or unacknowledged paraphrasing. Further examples include:

* Use of any quotation(s) from the published or unpublished work of other persons, whether published in textbooks, articles, the Web, or in any other format, where quotations have not been clearly identified as such by being placed in quotation marks and acknowledged.
* Use of another person’s words or ideas that have been slightly changed or paraphrased to make it look different from the original.
* Summarising another person’s ideas, judgments, diagrams, figures, or computer programmes without reference to that person in the text and the source in a bibliography/reference list.
* Use of assessment writing services, essay banks and/or any other similar agencies (NB. Students are commonly being blackmailed after using essay mills).
* Use of unacknowledged material downloaded from the Internet.
* Re-use of one’s own material except as authorised by your degree programme.

**Collusion**, which can be defined as when work that that has been undertaken with others is submitted and passed off as solely the work of one person. Modules will clearly identify where joint preparation and joint submission are permitted, in all other cases they are not.

**Fabrication of data**, making false claims to have carried out experiments, observations, interviews or other forms of data collection and analysis, or acting dishonestly in any other way.

## How is my work graded?

Assessment grading is subject to thorough quality control processes. You can view a summary of these processes on the [Assessment Explained Infographic](https://outlookuwicac.sharepoint.com/sites/QED/Shared%20Documents/Forms/Front%20Page.aspx?id=%2Fsites%2FQED%2FShared%20Documents%2Fstudent%20guide%20%2D%20Is%20my%20mark%20fair%2Epdf&parent=%2Fsites%2FQED%2FShared%20Documents&p=true&originalPath=aHR0cHM6Ly9vdXRsb29rdXdpY2FjLnNoYXJlcG9pbnQuY29tLzpiOi9zL1FFRC9FYzNrWVFRZUVIZEtyQ2JvX3RKbnIya0JvbUlpaUxJTm1QZWJVZ3ZUVWxqcTlRP3J0aW1lPXFLb08zblB3MkVn).

Grading of work at each level of Cardiff Met degree courses is benchmarked against a set of general requirements set out in [Volume 1, Section 4.3](https://www.cardiffmet.ac.uk/registry/academichandbook/Documents/AH1_04_03.pdf) of our Academic Handbook. A simplified version of these Grade Band Descriptors (GBDs) with short videos explaining some of the academic terminology used can be accessed via the [Facilitation of Learning](https://outlookuwicac.sharepoint.com/sites/QED/SitePages/Facilitation-of-Learning.aspx) resource page.

We would strongly recommend looking at the [Study Smart](https://learn.cardiffmet.ac.uk/course/view.php?id=1416) area of Moodle to find out more about assessments and key academic skills which can have a significant impact on your grades. Always check your work thoroughly before submission.



|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Module Title | | | | | | **Module Number** | | **JACS Subject Code(s) and % of each subject** | | | | | **ASC Category(ies)** | | |
| Object Oriented Programming | | | | | | CSE4006 | | I322 | | | | | 6 | | |
| **Level (3 to 8)** | | **Credits** | | **ECTS Credit** | | **Module Value (1=20 credits)** | | | | **% Taught in Welsh** | | | | **Module Type** | |
| 4 | | 15 | | 7.5 | | 0.75 | | | | 0% | | | | Taught | |
| **Teaching Period (Term/Semester)** | | | | | | | **Pre-requisites** | | | | | | | | |
| Semester 2 | | | | | | | CSE4002, CSE4004 | | | | | | | | |
| **Module Leader** | | | | | **School(s)** | | | | | | | Campus | | | |
| Upekha Wijesinhge | | | | | Cardiff School of Technologies | | | | | | | ICBT | | | |
| Assessment Methods | | | | | | | | | | | | | | | |
| Assessment Code and  Method | | | Duration/Length of  Assessment Method | | | | Weighting of Assessment | | | | **Threshold** | | | Approximate Date  of Submission | |
| WRIT1-Coursework | | | 3000 words equivalent | | | | 100% | | | | 1 | | | End Semester | |
| Rationale for Assessment and Opportunity for Feedback – This field is optional. | | | | | | | | | | | | | | | |
| Writ1-Assignment is providing opportunity assess the students understanding of concepts of Object-oriented programming. Also assess the ability to design and develop Object oriented program. | | | | | | | | | | | | | | | |
| **Aim(s)** | | | | | | | | | | | | | | | |
| Through this module student can gain sound understand about the Object-Oriented program with their technical concepts and how that can be applied for real time development industry. | | | | | | | | | | | | | | | |
| **Learning Outcomes** | | | | | | | | | | | | | | | |
| On successful completion of this module, students should be able to:   * Explain the fundamentals of Object-Oriented Programming concepts * Design Object-Oriented based applications * Develop Object-Oriented applications | | | | | | | | | | | | | | | |
| **Learning and Teaching Delivery Methods** | | | | | | | | | | | | | | | |
| **Method** | **Rationale** | | | | | | | | **Type of Contact (scheduled/ guided independent study/placement)** | | | | | | **Total hours** |
| Lecture | Introduction to module - essential information & guidance for students - module guidelines and identification of the learning outcomes associated with the module | | | | | | | | SCHEDULED | | | | | | 30 |
| Workshops / Labs | To allow exploration of all aspects of module content (knowledge, understanding, skills & other attributes) in an interactive group setting | | | | | | | | SCHEDULED | | | | | | 30 |
| Student Centred Learning hours | Research on the related technologies. Development and testing of the applications and preparation for Viva. | | | | | | | | NON-CONTACT | | | | | | 90 |
| Total |  | | | | | | | |  | | | | | | 150 |
| **Indicative Content** | | | | | | | | | | | | | | | |
| Basic Principles of object-oriented programming:   * + Class, Objects   + Features: Abstraction, inheritance, polymorphism, encapsulation.   Object Oriented Concepts using UML:   * + USE Case diagram, Class Diagram, Sequence diagram   Programming basics:   * + Variables and Data Types     - Primitive data types, Derived data types   + Declare and initialize variables   + Operators: E.g.: Arithmetic operators, Relational operators, Logical operators, Increment/Decrement operators, Assignment operators, Bitwise Operators, ternary operator, instance of Operator   + Decision making structures: if, if...else, nested if, switch statement   + Loop Control structures: while, do. while, for     - Loop Control Statements: break statement, continue statement   + Creating and Using Arrays: Declare, instantiate, initialize and use of one-dimensional and multi-dimensional array.   Methods: Create methods, calling methods, Passing Parameters by Value and by Reference, Return types  Constructors: Constructors, Calling Constructors, Parameterized Constructor, this keyword  Object oriented programming implementation:   * + Inheritance:   Types of Inheritance: Single, Multilevel, Hierarchical, Multiple inheritance (Explain the reason not supporting by java)  Keywords: Super, extends, Instance of, Final class,   * + Polymorphism: Method Overloading, Method Overriding   + Abstraction: Abstract class, Interfaces   + Encapsulation:     - access modifiers: private, default, protected, public     - non-access modifiers.: static, abstract     - Get and Set Method   Handling Exceptions:   * + Catching and Handling Exceptions (try, catch blocks), Catching Multiple Exceptions, Throws/Throw Keywords   Threads:   * + Thread class, Runnable interface   GUI Application: Swing and AWT  File handling and I/O: Stream classes, Reading and Writing Files  Java collection framework: ArrayList, Vector, LinkedList, List, Queue, Deque | | | | | | | | | | | | | | | |
| **Required Reading** | | | | | | | | | | | | | | | |
| Murach, J., 2017. *Murach's Java Programming*. [S.L.]: Mike Murach & Associates. | | | | | | | | | | | | | | | |
| Recommended Reading | | | | | | | | | | | | | | | |
| [Kathy, S](https://www.amazon.com/Kathy-Sierra/e/B001H6U55G/ref=dp_byline_cont_book_1)., 2005. Head First Java, 2nd Edition. California: O'Reilly Media. Preece J., Rogers, Y. & Sharp, H. (2015), Interaction Design. 4th ed. New York: Wiley.  Burd, B., 2012. Beginning Programming with Java for Dummies. 3rd ed. John Wiley & Sons.  Tutorialspoint, 2020. Java Tutorial. [ online ] Available at:< <https://www.tutorialspoint.com/java/index.htm> > **[**Accessed 5th April 2019] | | | | | | | | | | | | | | | |
| **Access to Specialist Requirements** | | | | | | | | | | | | | | | |
| J2SE ,NetBeans | | | | | | | | | | | | | | | |

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **CARDIFF SCHOOL OF** **TECHNOLOGY: VERIFICATION OF INTERNAL MODERATION OF ASSESSMENT** | | | | | | | | | | | | | | **IV2** |
| **Module Number:**  **CSE4006** | | **Module Name:**  **Object Oriented Programming** | | | | | | **Module Leader: Upeka Wijesinghe** | | | |  | **Year/Term:**  ***2024*** | |
| **MODULE ASSESSMENT STRATEGY** | | | | | **BRIEFING INCLUDES:** | **X** | | **IV’s COMMENTS / RECOMMENDATIONS** | | | | **MODULE LEADER’S ACTION TAKEN** | | |
| WRIT1-Coursework | | | ***100%*** | | **Description of Task/Problem/Topic** |  | | **All leaning outcome covered by task list.** | | | |  | | |
|  | | |  | | **Hand in Details** |  | |
|  | | |  | | **Guidance Notes** |  | |
|  | | |  | | **Learning Outcomes** |  | |
| **Final Grade Weighting** | | | ***100%*** | | **Assessment Criteria** |  | |
| **ASSESSMENT DETAILS** | | | | | **Mark/Grade Criteria** |  | |
|  | | | | | **Marking Scheme** |  | |
| **Feedback Sheet** |  | |
| **Assessment type:**  **Report and a workshop** | | | | | **Referencing Requirements Given** |  | |
| **Nominated IV:chatura** | | | | | | | | **IV Signature: Chatura** | | | | | **Date: 2-1-25** | |
| **CARDIFF METROPOLITAN LINK TUTOR COMMENTS:** | | | | | | | | | **EXTERNAL EXAMINER COMMENTS:** | | | | | |
| **X** | **Please check appropriate box below** | | | | | | | | **X** | **Please check appropriate box below** | | | | |
|  | **I confirm that I have considered the above draft assignment/exam and I am happy to approve the content.**  **Assessment can now be forwarded to the External Examiner for approval.** | | | | | | | |  | **I confirm that I have considered the above draft assignment/exam and I am happy to approve the content.**  **Assessment can now be released to the students.** | | | | |
|  | **I confirm that I have considered the above draft assignment/exam and I am happy to approve the content subject to the above amendments.**  **Assessment can be forwarded to the External Examiner once these changes have been implemented and verified.** | | | | | | | |  | **I confirm that I have considered the above draft assignment/exam and I am happy to approve the content subject to the above amendments.**  **Assessment can be released to students once these changes have been implemented and verified.** | | | | |
|  | **I confirm that I have considered the above draft assignment/exam and suggest the above amendments.**  **I would like to see the final amended version before I confirm approval.** | | | | | | | |  | **I confirm that I have considered the above draft assignment/exam and suggest the above amendments.**  **I would like to see the final amended version before I confirm approval.** | | | | |
| **Cardiff Metropolitan Link Tutor:** | | | | **Link tutor signature:** | | | **Date:** | | **External Examiner:** | | **EE Signature:** | | **Date:** | |

**Automation of 'Connect Me' Using Object-Oriented Programming**

**Beruwalage Krishan Maduwantha**

**GL/HDCSE/CMU/11/03**

**Object Oriented Programming**

**CSE4006**

**Mr. Aruna Indika**

**ICBT Campus**

**2025**

|  |  |  |
| --- | --- | --- |
| **Qualification** | | **Module Number and Title** |
| Level 4 | | CSE4006 - Object Oriented Programming |
| **Student Name & No.** | | **Assessor** |
| Krishan maduwantha beruwalage  GL/HDCSE/CMU/11/03 | | Mr. Aruna Indika |
| **Hand out date** | | **Submission Date** |
| 03/04/2025 | | 04/04/2025 |
| **Assessment type**  Course work | **Duration/Length of Assessment Type**  3000 word(Coursework) | **Weighting of Assessment**  100% (Individual work) |

Assignment cover sheet

I, Krishan Maduwantha Beruwalage and GL/HDCSE/CMU/11/03 <name of the student and registration number>, certify that the work submitted for this assignment is my own and research sources are fully acknowledged.

**Learner declaration**

|  |  |  |  |
| --- | --- | --- | --- |
| **Marks Awarded** | | | |
| First assessor | |  | |
| IV marks | |  | |
| Agreed grade | |  | |
| Signature of the assessor |  | Date |  |

**Feedback Form**

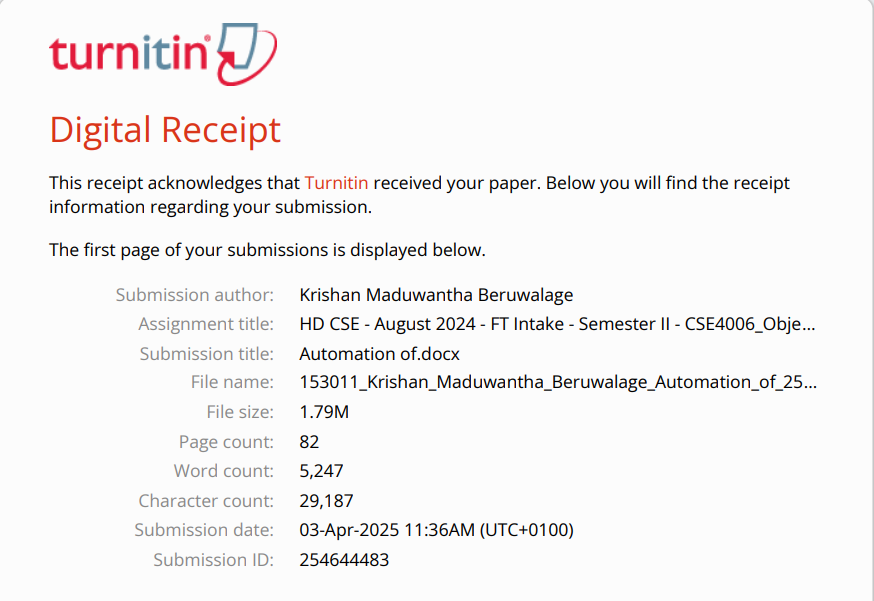
**International College of Business & Technology**

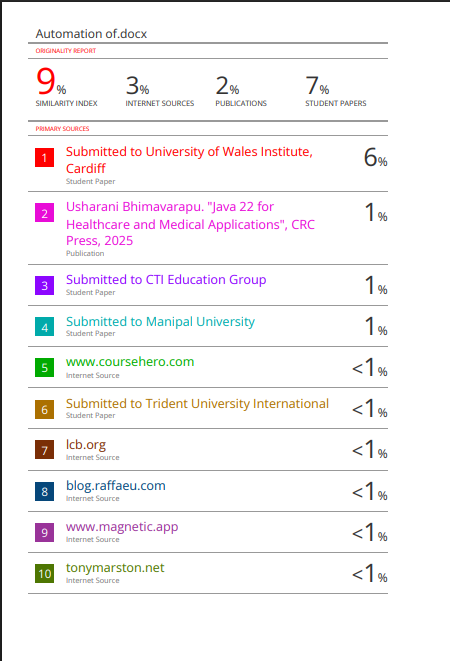
**Module:** Object Oriented Programming

**Student ID: GL/HDCSE/CMU/11/03**

**Assignment: Writ1**

|  |  |  |  |
| --- | --- | --- | --- |
| **Assessor Feedback** | | | |
|  | | | |
| General Comment |  | | |
| **Assessor Name** | | **Signature** | **Date** |
|  | |  |  |





Contents

[Acknowledgment 34](#_Toc194588418)

[Executive summary 35](#_Toc194588419)

[Tables and Figures 36](#_Toc194588420)

[Introduction 39](#_Toc194588421)

[Task 1: Design Solution (UML Diagrams) 41](#_Toc194588422)

[1.1. Use Case Diagrams for "Connect Me" HR Management System and HR Assistant System 41](#_Toc194588423)

[1.2 Use Cases for HR Manager: 42](#_Toc194588424)

[1.2.1 Login & Logout 42](#_Toc194588425)

[1.2.2 Add Departments & Designations 42](#_Toc194588426)

[1.2.3 Manage Employees 42](#_Toc194588427)

[1.2.4 Search Employee 43](#_Toc194588428)

[1.2.5 Create New Account (HR Assistant) 43](#_Toc194588429)

[1.3 Use cases for HR Assistant: 43](#_Toc194588430)

[1.3.1 Login & Logout 43](#_Toc194588431)

[1.3.2 Search Employee 44](#_Toc194588432)

[1.4 HR Manager System Use Case Diagrams 45](#_Toc194588433)

[1.4.1 Login & Logout 45](#_Toc194588434)

[1.4.2 Manage Departments & Designations 46](#_Toc194588435)

[1.4.3 Manage Employees 47](#_Toc194588436)

[1.4.4 Search Employee 48](#_Toc194588437)

[1.4.5 Create HR Assistant Account 49](#_Toc194588438)

[1.5 HR Assistant System Use Case Diagrams 50](#_Toc194588439)

[1.5.1 Login & Logout 50](#_Toc194588440)

[1.5.2 Search Employee 51](#_Toc194588441)

[1.6 Justification of Use Case Diagrams 52](#_Toc194588442)

[1.7 Class Diagram 53](#_Toc194588443)

[1.7.1 Classes & Attributes 53](#_Toc194588444)

[1.7.2 Relationships: 56](#_Toc194588445)

[1.8 Class diagram for the “Connect Me” company 57](#_Toc194588446)

[1.9 Justification for the Class Diagram 58](#_Toc194588447)

[1.10 Sequence Diagrams 58](#_Toc194588448)

[1.10.1 Sequence Diagrams for HR Manager 59](#_Toc194588449)

[1.10.2 Sequence Diagrams for HR Assistant 67](#_Toc194588450)

[1.11 Justification for Sequence Diagrams 80](#_Toc194588451)

[Task 02 - Object-Oriented Concepts in Connect-me 81](#_Toc194588452)

[2.1 Introduction 81](#_Toc194588453)

[2.2 Classes and Objects 82](#_Toc194588454)

[2.3 Encapsulation 84](#_Toc194588455)

[2.4 Abstraction 85](#_Toc194588456)

[2.5 Inheritance 87](#_Toc194588457)

[2.6 Polymorphism 88](#_Toc194588458)

[2.7 Array 89](#_Toc194588459)

[1. String Array for Department and Designation (Add\_Department\_and\_Designation.java) 90](#_Toc194588460)

[2. Splitting a String into an Array (Create\_New\_Account.java & Manage\_Employee.java) 91](#_Toc194588461)

[2.8 File Handling in Java 92](#_Toc194588462)

[2.8.1 Writing and Reading Employee Data from a File 92](#_Toc194588463)

[2.8.2 Example Output 95](#_Toc194588464)

[2.9 Error Handling in Java 95](#_Toc194588465)

[Task 03 - User Manual for 'Connect Me' Project 97](#_Toc194588466)

[3.1. Overview 97](#_Toc194588467)

[3.2 System Requirements 97](#_Toc194588468)

[3.3 Installation & Setup 97](#_Toc194588469)

[3.4 Features & Usage 99](#_Toc194588470)

[3.4.1 Login Page 99](#_Toc194588471)

[99](#_Toc194588472)

[3.4.2 Sign in Page 100](#_Toc194588473)

[3.4.3 HR Manager Dashboard 101](#_Toc194588474)

[3.4.4 HR Assistant Dashboard 102](#_Toc194588475)

[3.4.5 Add Departments and Designations 103](#_Toc194588476)

[3.4.6 Manage Employee 104](#_Toc194588477)

[3.4.7 Create a new Account 105](#_Toc194588478)

[3.4.8 Search Employee 106](#_Toc194588479)

[3.4.9 File Storage 107](#_Toc194588480)

[3.5 Troubleshooting 108](#_Toc194588481)

[Conclusion 109](#_Toc194588482)

[Recommendation 111](#_Toc194588483)

[References 112](#_Toc194588484)

# Acknowledgment

First, I would like to thank my parents and ICBT University for contributing to this high national diploma. Also, I would like to thank our lecturer, Mr. Aruna Indika, for his valuable guidance and support throughout Object Oriented Programming (CSE4006). His dedication and encouragement contributed to the success of this lesson.

# Executive summary

“Connect Me” is a mobile service company that currently handles customer care employee details manually. To streamline operations, they wish to automate their HR processes. This system will allow HR Managers and HR Assistants to efficiently manage employee details, departments, and designations. The solution will be built using Object-Oriented Programming (OOP) principles, with functionalities like employee management, department creation, and account creation for HR Assistants. The data will be stored in a file system for easy retrieval and modification.

# Tables and Figures

[Figure 1- Login & Logout (HR Manager) (Use Case diagram) 20](#_Toc194588551)

[Figure 2 - Manage Departments & Designations (HR Manager) (Use Case diagram) 21](#_Toc194588552)

[Figure 3 - Manage Employees (HR Manager) (Use Case diagram) 22](#_Toc194588553)

[Figure 4 - Search Employee (HR Manager) (Use Case diagram) 23](#_Toc194588554)

[Figure 5 – Create New Account (HR Assistant) (HR Manager) (Use Case diagram) 24](#_Toc194588555)

[Figure 6 - Login & Logout (HR Assistant) (Use Case diagram) 25](#_Toc194588556)

[Figure 7 - Search Employee (HR Assistant) (Use Case diagram) 26](#_Toc194588557)

[Figure 8 - Class diagram 32](#_Toc194588558)

[Figure 9 - Login Process HR Manager (sequence diagram) 44](#_Toc194588559)

[Figure 10 - HR Manager Manages Departments and Designations (sequence diagram) 45](#_Toc194588560)

[Figure 11 - HR Manager Managing Employees (sequence diagram) 46](#_Toc194588561)

[Figure 12 - HR Manager Searching for Employees (sequence diagram) 47](#_Toc194588562)

[Figure 13 - HR Manager Creating a New Account (HR Assistant) (sequence diagram) 48](#_Toc194588563)

[Figure 14 - HR Manager Logout (sequence diagram) 49](#_Toc194588564)

[Figure 15 - Login Process HR Assistant (sequence diagram) 50](#_Toc194588565)

[Figure 16 - HR Assistant Searching for Employees (sequence diagram) 51](#_Toc194588566)

[Figure 17 - HR Assistant Logout (sequence diagram) 52](#_Toc194588567)

[Figure 18 - Class in the program 55](#_Toc194588568)

[Figure 19 - Class in the program 56](#_Toc194588569)

[Figure 20 - Encapsulation 57](#_Toc194588570)

[Figure 21 – Abstraction 58](#_Toc194588571)

[Figure 22 – Inheritance 59](#_Toc194588572)

[Figure 23 – Polymorphism 60](#_Toc194588573)

[Figure 24 - String Array for Department and Designation 62](#_Toc194588574)

[Figure 25 - Splitting a String into an Array 63](#_Toc194588575)

[Figure 26 - File path 64](#_Toc194588576)

[Figure 27 – BufferedWriter 64](#_Toc194588577)

[Figure 28 – BufferedReader 65](#_Toc194588578)

[Figure 29 - Example Output 66](#_Toc194588579)

[Figure 30 - Error Handling 67](#_Toc194588580)

[Figure 31 - Java Development Kit (JDK) installed 68](#_Toc194588581)

[Figure 32 - required text files 69](#_Toc194588582)

[Figure 33 - Login Page 70](#_Toc194588583)

[Figure 34 - Sign in Page 71](#_Toc194588584)

[Figure 35 - HR Manager Dashboard 72](#_Toc194588585)

[Figure 36 - HR Assistant Dashboard 73](#_Toc194588586)

[Figure 37 - Add Departments and Designations 74](#_Toc194588587)

[Figure 38 - Manage Employee 75](#_Toc194588588)

[Figure 39 - Create a new Account 76](#_Toc194588589)

[Figure 40 - Search Employee 77](#_Toc194588590)

[Figure 41 - File Storage 78](#_Toc194588591)

# 

# Introduction

"Connect Me" aims to enhance its HR management by automating employee information handling. Currently, the organization relies on manual processes, which are time-consuming and prone to errors. To address these challenges, an Object-Oriented Programming (OOP)-based HR automation system is being designed and developed.

This system will streamline HR tasks for HR Managers and HR Assistants, enabling them to manage employee details efficiently. Employees will be categorized based on their respective **departments** and **designations**, ensuring structured data organization. Additionally, the system will allow HR Managers to create and manage accounts for HR Assistants. For data persistence and easy retrieval, all employee records will be stored in files.

Departments in "Connect Me":

* Operations
* Quality Assurance
* Training
* Customer Service
* IT
* Human Resources (HR)

Designations in "Connect Me":

* Supervisor
* Team Captain
* Quality Manager
* Quality Monitor
* Training Manager
* IT Manager

By implementing this HR automation system, "Connect Me" seeks to improve efficiency, accuracy, and accessibility in managing employee information, ultimately contributing to smoother HR operations.

# Task 1: Design Solution (UML Diagrams)

## ****1.1. Use Case Diagrams for "Connect Me" HR Management System and HR Assistant System****

**Purpose:**

The **Use Case Diagram** identifies the key functionalities of the system and how the actors (**HR Manager and HR Assistant**) interact with it.

**Explanation:**

The diagram includes **two primary actors:**

* **HR Manager**: Has full control over employee management, department and designation management, and account creation for HR Assistants.
* **HR Assistant**: Has limited access and can only search employee details.

**HR Manager** handles:

* Adding, updating, and deleting **departments** and **designations**.
* Managing employees (adding, assigning designations, updating, and removing).
* Searching for employees.
* Creating HR Assistant accounts.

**HR Assistant** can only:

* Search employees by department and name.

## ****1.2 Use Cases for HR Manager:****

### ****1.2.1 Login & Logout****

* Login
* Validate credentials (**includes:** Login)
* Load HR Manager Dashboard (**includes:** Login)
* Logout
* End Session (**includes:** Logout)

### ****1.2.2 Add Departments & Designations****

* Add New Department (**includes:** validation)
* Add New Designation
* Validate data

### ****1.2.3 Manage Employees****

* Add New Employee (**includes:** validation)
* Allocate Employee to Department (**includes:** Add New Employee)
* Assign Designation
* Update Employee Details
* Remove Employee
* Validate data

### ****1.2.4 Search Employee****

* Search by Department
* Search by Name

### ****1.2.5 Create New Account (HR Assistant)****

* Add New **Account** HR Assistant
* Assign Login Credentials (**includes:** Add New **Account** (HR Assistant))
* Remove Account

## ****1.3 Use cases for HR Assistant:****

### ****1.3.1 Login & Logout****

* Login
* Validate credentials (**includes:** Login)
* Load HR Assistant Dashboard (**includes:** Login)
* Logout
* End Session (**includes:** Logout)

### 

### ****1.3.2 Search Employee****

* Search by Department
* Search by Name

## 1.4 HR Manager System Use Case Diagrams

### 1.4.1 ****Login & Logout****

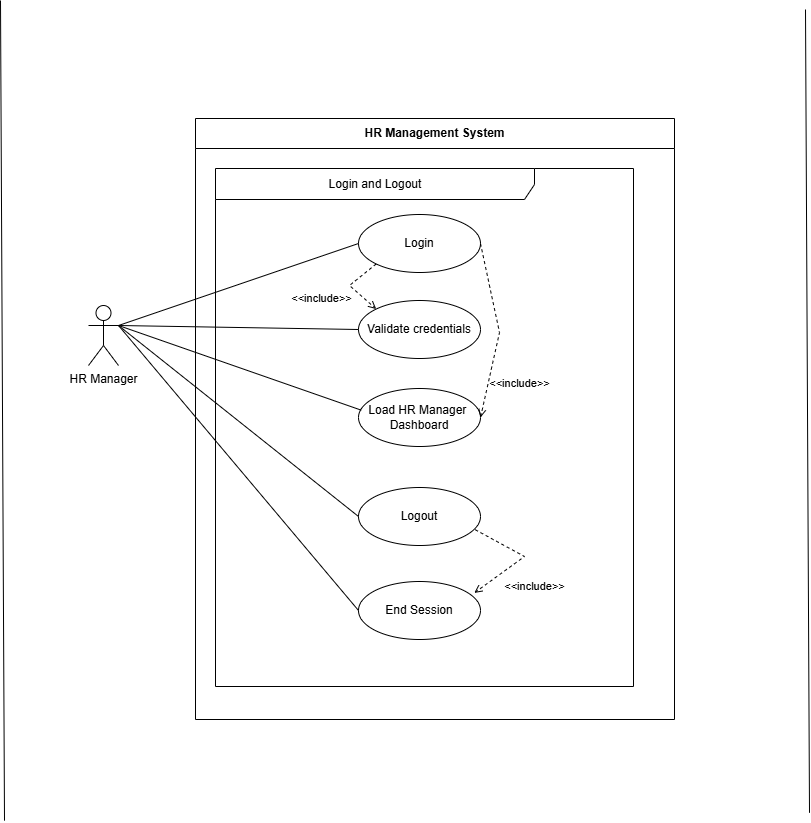


Figure 1- Login & Logout (HR Manager) (Use Case diagram)

### 1.4.2 ****Manage Departments & Designations****

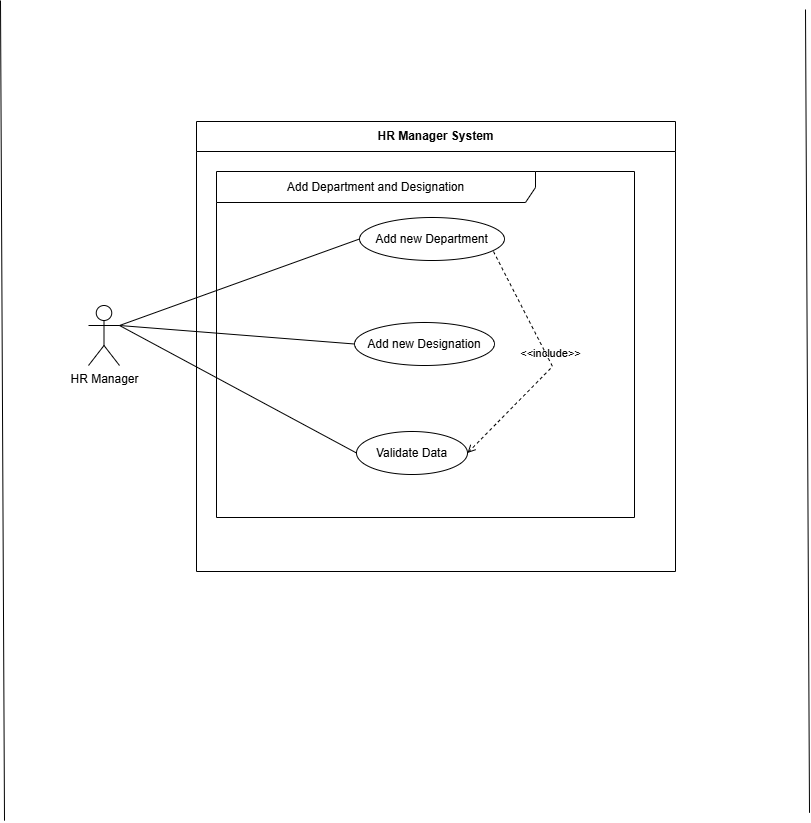


Figure 2 - Manage Departments & Designations (HR Manager) (Use Case diagram)

### 1.4.3 ****Manage Employees****

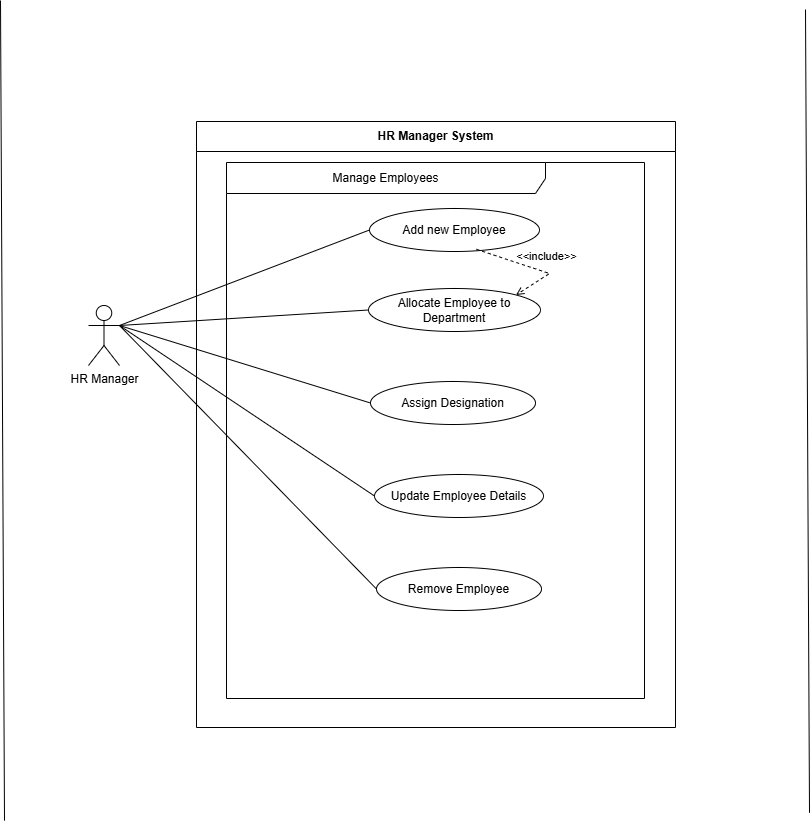


Figure 3 - Manage Employees (HR Manager) (Use Case diagram)

### 1.4.4 ****Search Employee****

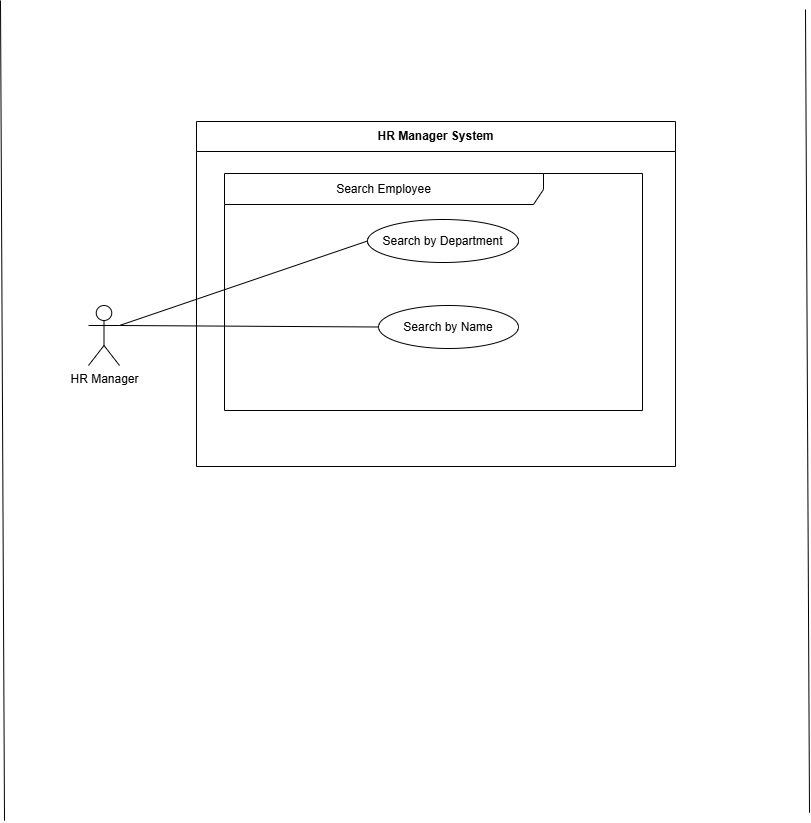


Figure 4 - Search Employee (HR Manager) (Use Case diagram)

### 1.4.5 **Create HR Assistant Account**

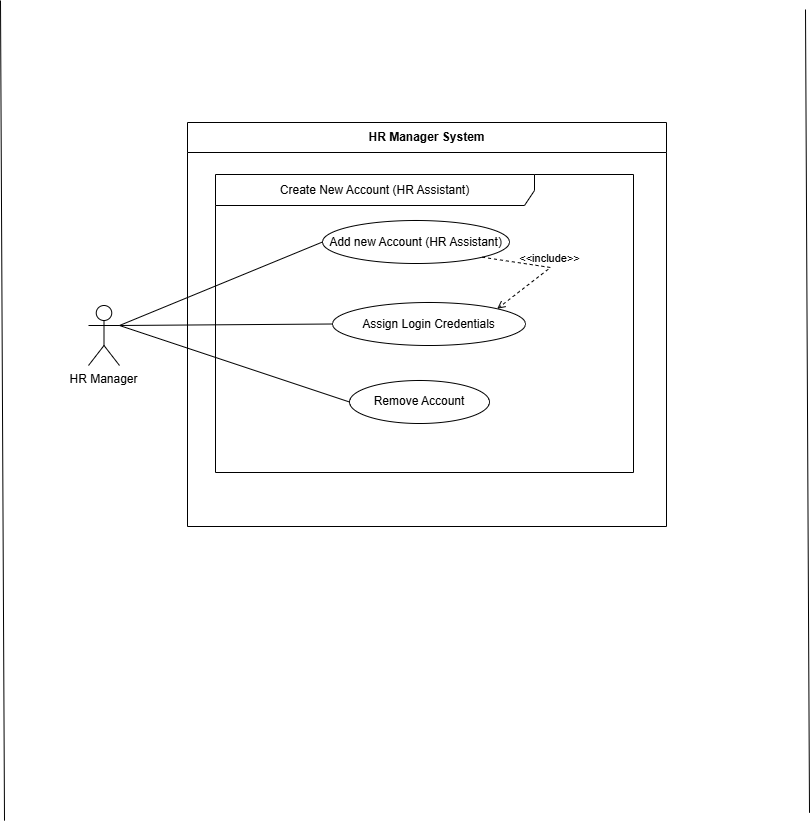


Figure 5 – Create New Account (HR Assistant) (HR Manager) (Use Case diagram)

## 1.5 HR Assistant System Use Case Diagrams

### 1.5.1 ****Login & Logout****

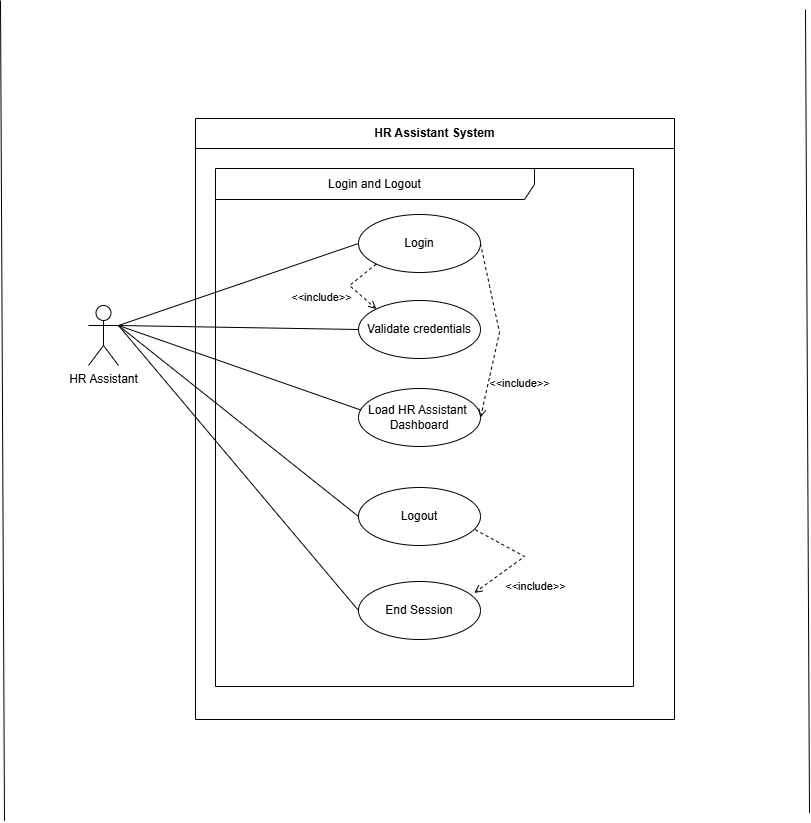


Figure 6 - Login & Logout (HR Assistant) (Use Case diagram)

### 1.5.2 ****Search Employee****

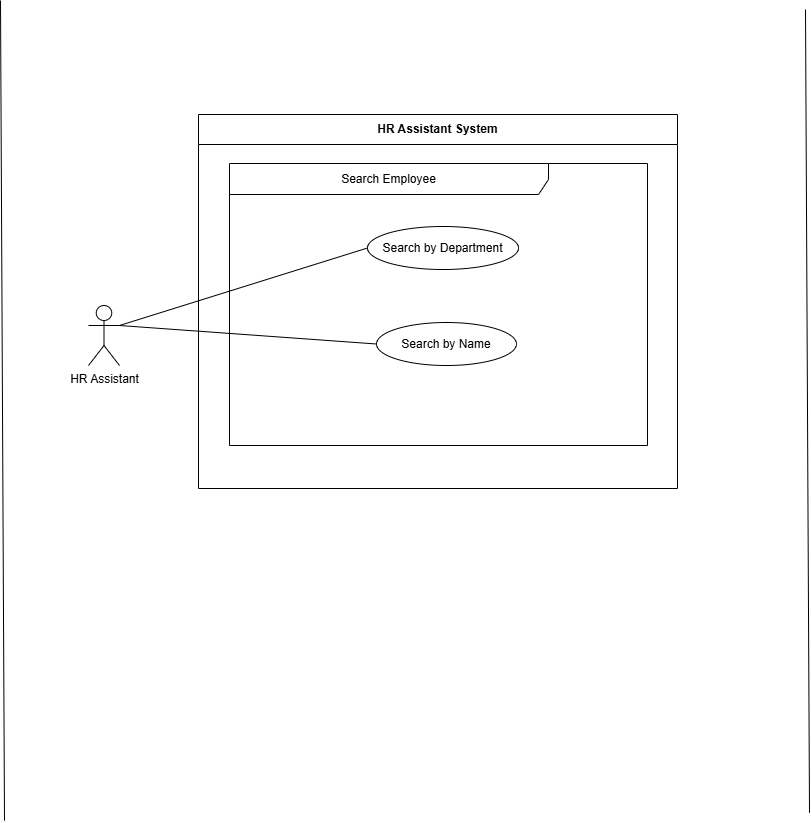


Figure 7 - Search Employee (HR Assistant) (Use Case diagram)

## 1.6 Justification of Use Case Diagrams

The Use Case Diagram for the "Connect Me" HR Management System provides a clear visual representation of system functionalities and user interactions. It defines the roles of the **HR Manager** and **HR Assistant**, outlining their specific privileges. The **HR Manager** has full control over employee, department, and designation management, as well as HR Assistant account creation, ensuring administrative authority. Meanwhile, the **HR Assistant** has restricted access, limited to searching employee details by department or name. By structuring interactions with "includes" and "extends" relationships, the diagram enhances clarity, modularity, and system functionality, making it easier to understand system scope and user permissions.

(lucidchart, 2025)

## 1.7 Class Diagram

The class diagram for the "Connect Me" HR Management System represents the system's object-oriented structure, including key classes, attributes, methods, and relationships.

This diagram defines how different components interact, focusing on:

* User Management (HR Manager & HR Assistant)
* Employee, Department, and Designation Handling
* Data Storage via File Handling

The HR Manager and HR Assistant classes inherit from the User class, ensuring proper authentication. Employees are linked to Departments and Designations, while a File Handler class manages data storage. The relationships between these components ensure a modular, scalable, and maintainable system design.

### 1.7.1 Classes & Attributes

**1. User (Superclass for HR Manager & HR Assistant)**

**Attributes:**

* -username: String
* -password: String

**Methods:**

* +login (): void
* +logout (): void

**2. HR Manager (Inherits from User)**

**Methods:**

* +add Department (name: String): void
* +add Designation (name: String): void
* +add Employee (employee: Employee): void
* +allocate Employee (employee: Employee, department: Department, designation: Designation): void
* +update Employee Details (employee: Employee): void
* +remove Employee (employee: Employee): void
* +search Employee (criteria: String): List<Employee>
* +create New Account (HR Assistant) (username: String, password: String): HR Assistant

**3. HR Assistant (Inherits from User)**

**Methods:**

* + +search Employee (criteria: String): List<Employee>

**4. Department**

**Attributes:**

* +name: String
* +employees: List<Employee>

**5. Designation**

**Attributes:**

* +title: String
* +employees: List<Employee>

**6. Employee**

**Attributes:**

* +name: String
* +id: String
* +department: Department
* +designation: Designation

**7. File Handler (Handles File Storage)**

**Methods:**

* +save Data (obj: Object): void
* +load Data (filename: String): Object

### ****1.7.2 Relationships:****

* **Inheritance:**
  + HR Manager and HR Assistant **inherit** from User.
* **Association:**
  + HR Manager **creates** HR Assistant.
* **Aggregation:**
  + Department and Designation **have multiple** Employee objects.
* **Composition:**
  + The file handler is tightly linked with the HR manager and HR assistant to save/load data.

# 1.8 Class diagram for the “Connect Me” company

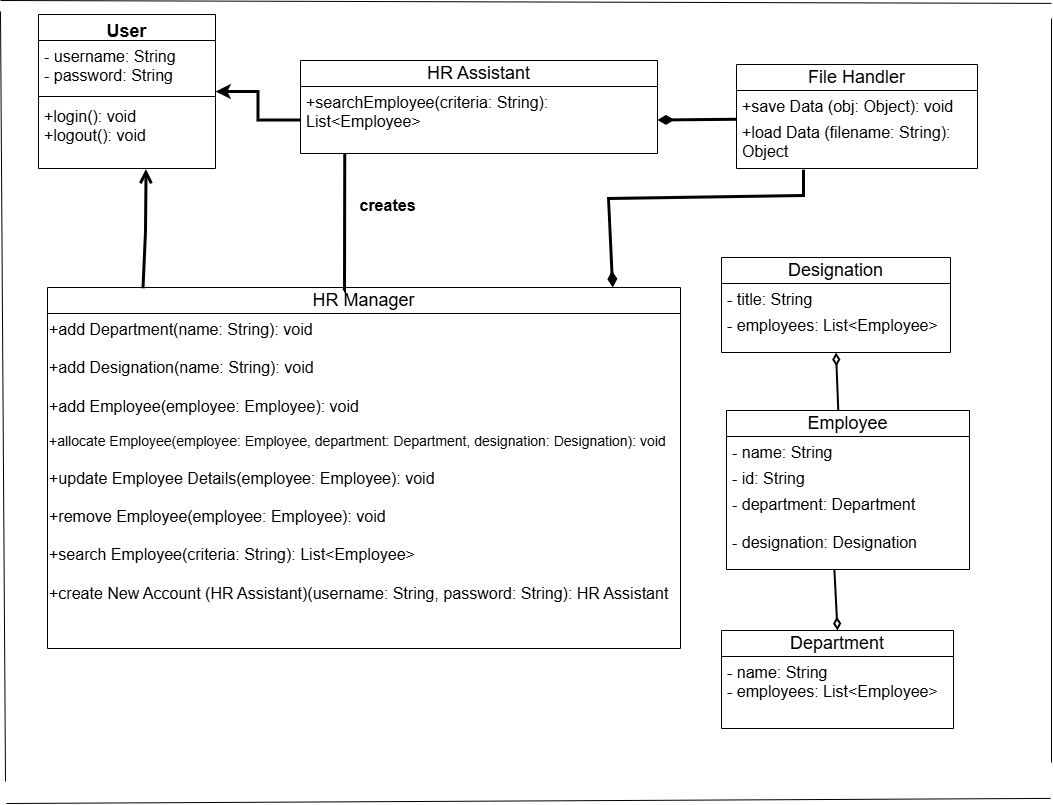


Figure 8 - Class diagram

## 1.9 ****Justification for the Class Diagram****

The class diagram for the "Connect Me" HR Management System is designed to ensure modularity, scalability, and maintainability by clearly defining object-oriented relationships. By utilizing inheritance, the system enforces authentication and role-based access, allowing HR Managers and Assistants to perform designated tasks. Aggregation between Departments, Designations, and Employees maintains flexibility, while the File Handler ensures efficient data persistence. These structured relationships enhance data organization, streamline operations, and improve overall system efficiency. (Bell, 2023)

# 1.10 Sequence Diagrams

A **Sequence Diagram** is a UML diagram that illustrates how objects interact with each other in a particular process over time. It visually represents the sequence of messages exchanged between actors (users) and system components.

In the **HR Management System for "Connect Me"**, sequence diagrams help to understand the flow of operations, such as logging in, adding employees, managing departments, searching employee details, and creating HR Assistant accounts.

Each sequence diagram contains:

* **Actors** (HR Manager, HR Assistant)
* **Objects** (System, Employee, File Handler, etc.)
* **Messages** (Synchronous and Asynchronous)
* **Control Structures** (Loops, Alternative paths like ALT, and Conditions)

### 1.10.1 Sequence Diagrams for HR Manager

**1. Login Process HR manager**

**Actors:** HR Manager.

Objects (Lifelines):

* **Login Screen**
* **HR System**

**Messages & Control Elements**

* **HR Manager → Login Screen:** login (username, password) (Synchronous message)
* **Login Screen → HR System: Send Credentials** (username, password) (Synchronous message)
* **HR System (ALT frame with Guard Conditions):**
  + If login **success:** HR System → HR Manager: Login Success (Return message)
  + If login **failure:** HR System → HR Manager: Login Failure (Return message)

**2. Managing Departments & Designations**

**Actors:** HR Manager.

Objects (Lifelines):

* HR System: Handles department and designation management.
* File Storage: Handles data persistence.

**Messages & Control Elements**

**(b) Manage Departments/Designations**

* **HR Manager → HR System:** selectManageDepartments ()
* **HR Manager → HR System:** selectManageDesignations ()

**Department Operations:**

* **HR Manager → HR System:** Add Department(name) (Synchronous message)
* **HR System → File Storage:** Save Data (department) (Asynchronous message)
* **File Storage → HR System:** Reply (Success/Failure) (Return message)
* **HR System → HR Manager:** Reply (Operation Success/Failure) (Return message)

**Designation Operations:**

* **HR Manager → HR System:** Add Designation(title) (Synchronous message)
* **HR System → File Storage:** Save Data (Designation) (Asynchronous message)
* **File Storage → HR System:** Reply (Success/Failure) (Return message)
* **HR System → HR Manager:** Reply (Operation Success/Failure) (Return message)

**3. HR Manager Managing Employees**

**Actors & Objects**

* **Actor**: HR Manager
* **Objects**: HR System

Lifelines & Messages

**3.1 HR Manager chooses to manage employees**

* HR Manager → HR System: Manage Employees (Synchronous Message)

**3.2 HR Manager performs actions (add, allocate, update, remove)**

ALT (Alternative Paths)

2.1 If the HR Manager adds an employee:

* HR Manager →HR System: Add Employee(employee) (Synchronous Message)
* HR System → Validate (Self Message)
* HR System → HR Manager: Confirmation (Return Message)

2.2 If the HR Manager allocates an employee:

* HR Manager → HR System: Allocate Employee (employee, department, designation) (Synchronous Message)
* HR System → Validate & Assign (Self Message)
* HR System → HR Manager: Confirmation (Return Message)

2.3 If the HR Manager updates an employee:

* HR Manager → HR System: Update Employee Details (employee) (Synchronous Message)
* HR System → Validate & Modify (Self Message)
* HR System → HR Manager: Confirmation (Return Message)

2.4 If the HR Manager removes an employee:

* HR Manager →HR System: Remove Employee(employee) (Synchronous Message)
* HR System → Validate & Delete (Self Message)
* HR System → HR Manager: Confirmation (Return Message)

**3.3 System saves changes**

* LOOP (Ensures that each transaction is saved):
* HR System → **File Storage:** Save Data(employee) (Asynchronous Message)
* **File Storage** → HR System: Reply (Success/Failure) (Return Message)
* HR System → HR Manager: Reply (Operation Success/Failure) (Return Message)

**4. HR Manager Searching for Employees**

**Actors & Objects**

* **Actor:** HR Manager
* **Objects:** System, File Storage

Lifelines & Messages

**4.1 HR Manager enters search criteria**

* HR Manager → HR System: Search Employee(criteria) (**Synchronous Message**)

4.2 **System retrieves employee details**

* HR System → File Storage: Load Data(filename) (**Asynchronous Message**)
* File Storage →HR System: Employee Data (Return Message)
* HR System → Process Search Request (**Self Message**)

4.3 **System processes the search request**

**LOOP (For each matching employee record):**

* HR System → Process each employee's data (**Synchronous Message**)
* HR System → Filter & Match Records (**Self Message**)
* HR System → Employee details (**Found Message**)

4.4 **Alternative Paths (ALT) – Handling Search Results**

* **If matching employee(s) found:**
  + HR System → HR Manager: Display Search Results (Return Message)
* **If no match found:**
  + HR System → HR Manager: No employees found (Return Message)

**5. Creating a New Account**

**Actors & Objects**

* **Actor:** HR Manager
* **Objects:** System, File storage

Lifelines & Messages

**5.1 HR Manager selects to create an HR Assistant account**

* **HR Manager → HR System: Create New Account (HR Assistant)** (**Synchronous Message**)

**5.2 HR Manager enters new HR Assistant details**

* **HR Manager → HR System:** Enter Assistant Details(details) **(Synchronous Message**)
* **HR System → Validate Details** (**Self Message**)

**5.3 ALT (Alternative Path - Invalid Details)**

If details are invalid, the system sends an error message:

* HR System → HR Manager: Invalid Details. Retry. (Return Message)

HR Manager re-enters details:

* HR Manager → HR System: Enter Assistant Details(details) **(Synchronous Message**)
* HR System → Validate Details Again (**Self Message**)

**5.4** System assigns login credentials

* **HR System** → Assign Login Credentials () (**Self Message**)

**5.5** System saves HR Assistant details to file

**LOOP (Ensuring Successful Data Storage) While the data is not saved, retry saving:**

* System → File Storage: Save Data (HR Assistant) (**Asynchronous Message**)
* File Storage → System: Success (Return Message)

**5.6** System confirms success

* **HR System → HR Manager: Account Created Successfully** (Return Message)

**6. HR Manager Logout (Self Message & ALT Frame for Confirmation)**

**Actors:** HR Manager.

Objects (Lifelines):

* **Log out Screen**
* **HR System**

**Messages & Control Elements**

* **HR Manager → Log out Screen:** logout () (Synchronous message)
* **Log out Screen → HR System: Logout ()** (Synchronous message)
* **HR System → HR System:** clearSessionData () (Self message)
* **HR System → Log out Screen:** Reply (Logout Successful) (Return message)
* **Log out Screen → HR Manager:** Logout Successful (Return message)

### 1.10.2 Sequence Diagrams for HR Assistant

**1. Login Process HR Assistant**

**Actors:** HR Assistant.

Objects (Lifelines):

* **Login Screen**
* **HR Assistant System**

**Messages & Control Elements**

* **HR Assistant → Login Screen:** login (username, password) (Synchronous message)
* **Login Screen → HR Assistant System: Send Credentials** (username, password) (Synchronous message)
* **HR Assistant System (ALT frame with Guard Conditions):**
  + If login **success:** HR Assistant System → HR Assistant: Login Success (Return message)
  + If login **failure:** HR Assistant System → HR Assistant: Login Failure (Return message)

**2. HR Assistant Searching for Employees**

**Actors & Objects**

* **Actor:** HR Assistant
* **Objects:** HR Assistant System, File Storage

Lifelines & Messages

**2.1 HR Assistant enters search criteria**

* HR Assistant→ HR Assistant System: Search Employee(criteria) (**Synchronous Message**)

2.2 **System retrieves employee details**

* HR Assistant System → File Storage: Load Data(filename) (**Synchronous Message**)
* File Storage →HR Assistant System: Employee Data (Return Message)
* HR Assistant System → Process Search Request (**Self Message**)

2.3 **System processes the search request**

**LOOP (For each matching employee record):**

* HR Assistant System → Process each employee's data (**Synchronous Message**)
* HR Assistant System → Filter & Match Records (**Self Message**)
* HR Assistant System → Employee details (**Found Message**)

2.4 **Alternative Paths (ALT) – Handling Search Results**

* **If matching employee(s) found:**
  + HR Assistant System → HR Assistant: Display Search Results (Return Message)
* **If no match found:**
  + HR Assistant System → HR Assistant: No employees found (Return Message)

**3. HR Assistant Logout (Self Message & ALT Frame for Confirmation)**

**Actors:** HR Assistant.

Objects (Lifelines):

* **Log out Screen**
* **HR Assistant System**

**Messages & Control Elements**

* **HR Assistant → Log out Screen:** logout () (Synchronous message)
* **Log out Screen → HR Assistant System: Logout ()** (Synchronous message)
* **HR Assistant System → HR Assistant System:** clearSessionData () (Self message)
* **HR Assistant System → Log out Screen:** Reply (Logout Successful) (Return message)
* **Log out Screen → HR Assistant:** Logout Successful (Return message)

**1. Login Process HR Manager Diagram**

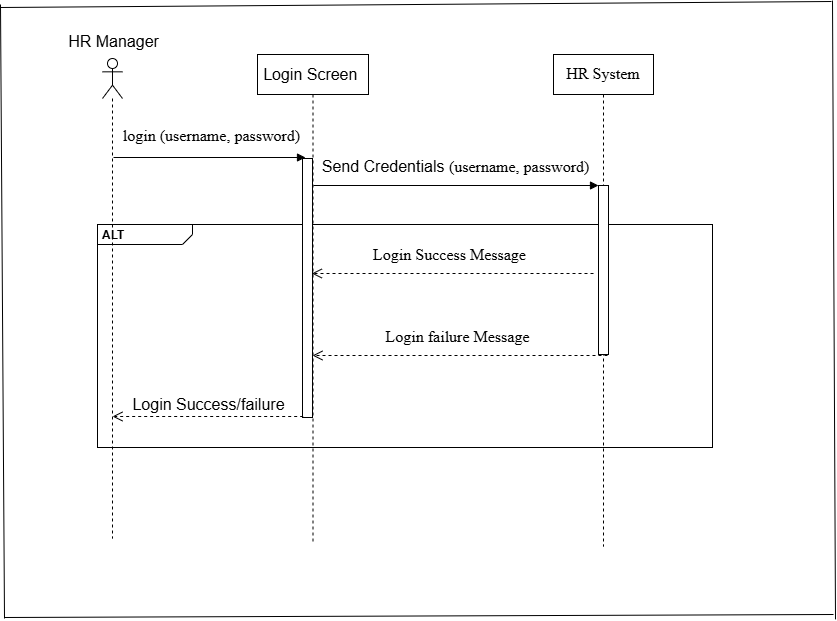
****

Figure 9 - Login Process HR Manager (sequence diagram)

**2. HR Manager Manages Departments and Designations diagram**

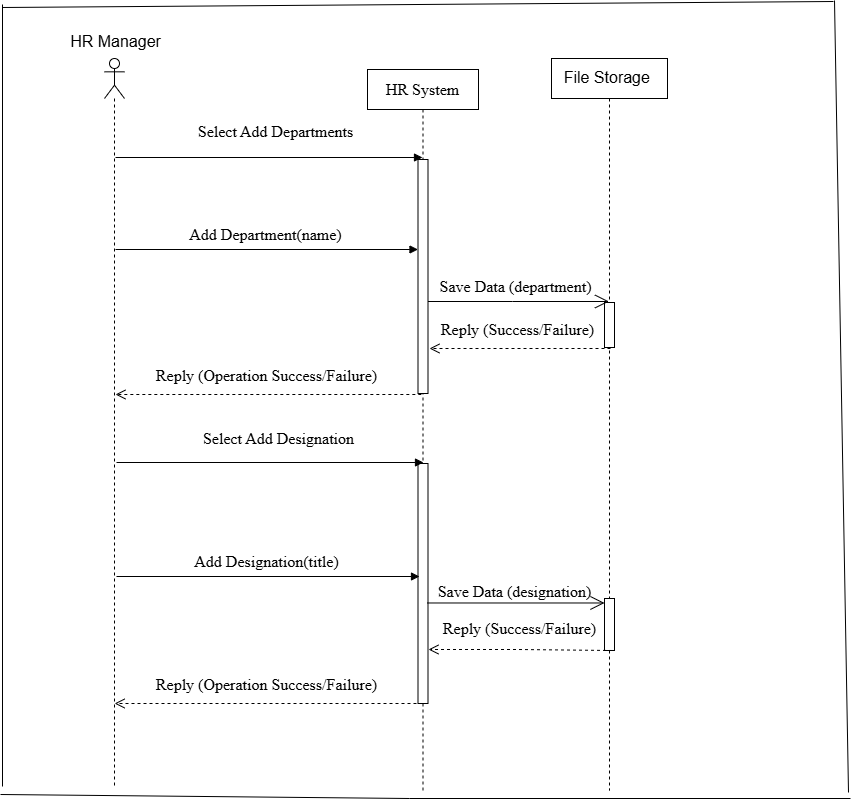


Figure 10 - HR Manager Manages Departments and Designations (sequence diagram)

**3. HR Manager Managing Employees**

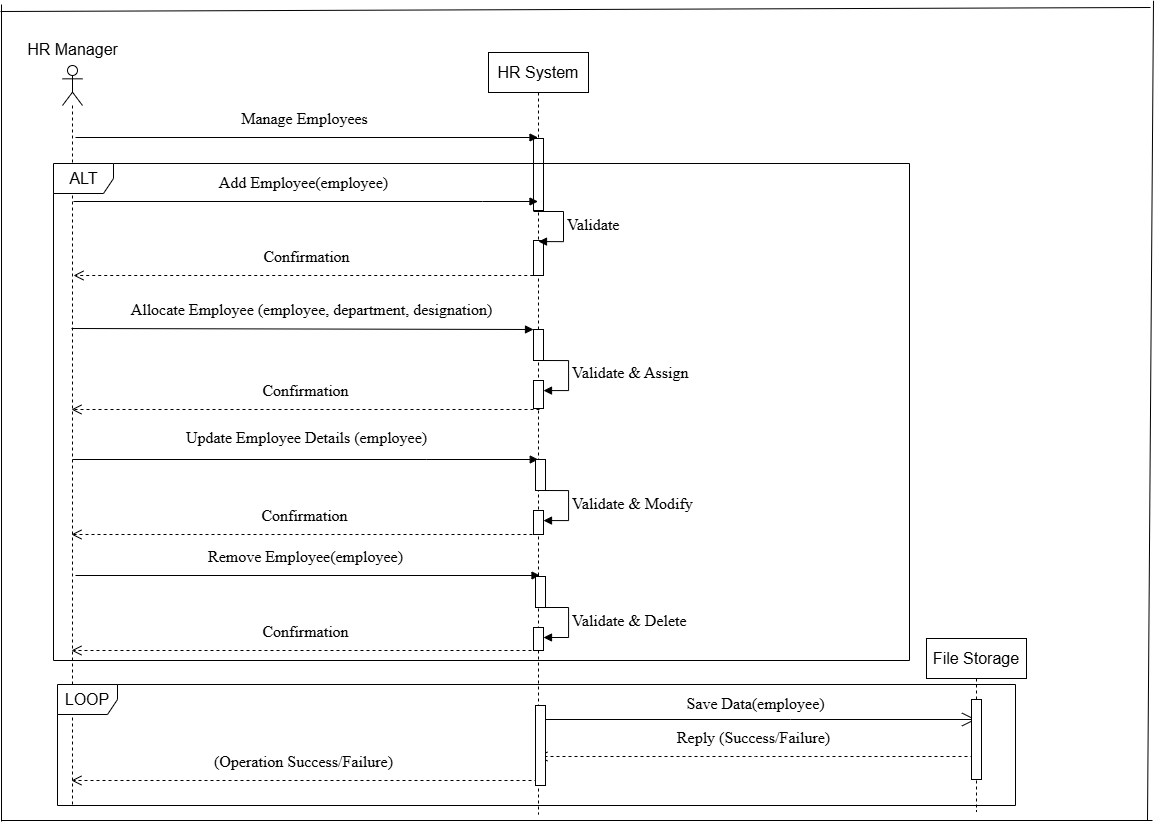


Figure 11 - HR Manager Managing Employees (sequence diagram)

4. **HR Manager Searching for Employees**

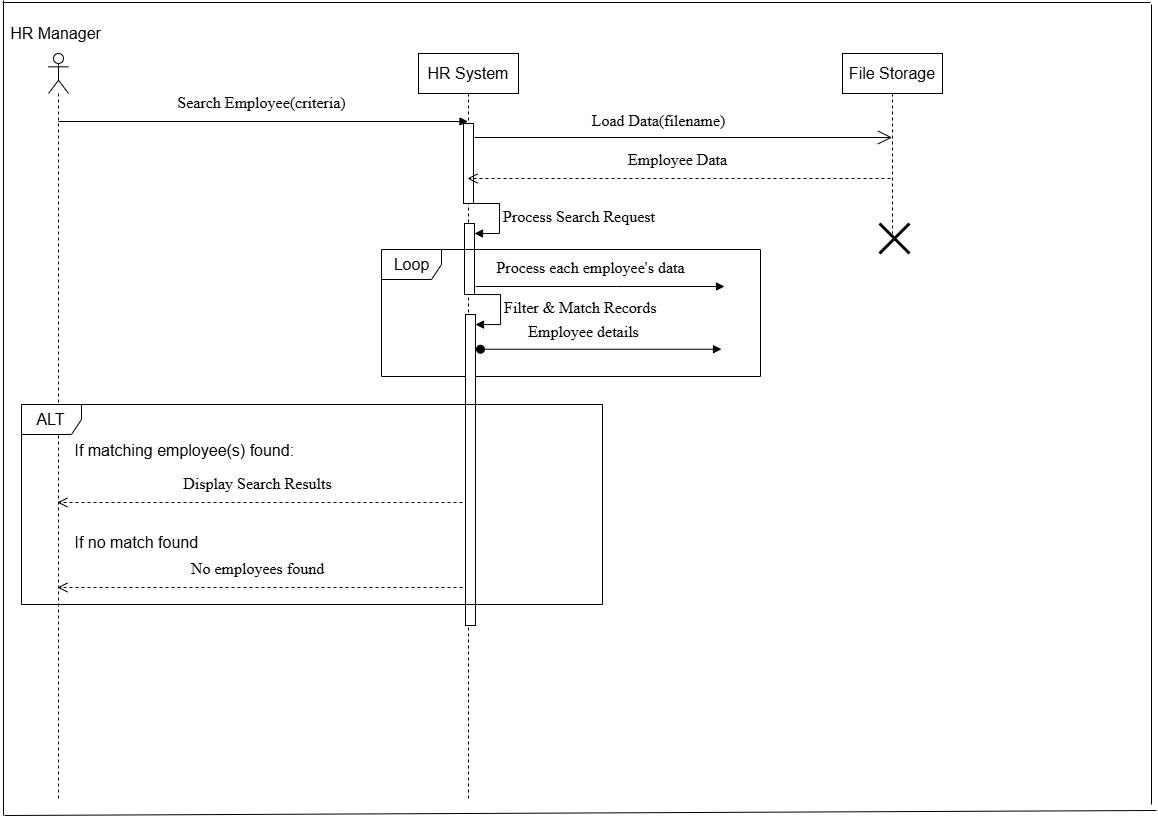


Figure 12 - HR Manager Searching for Employees (sequence diagram)

**5. Creating a New Account (HR Assistant)**

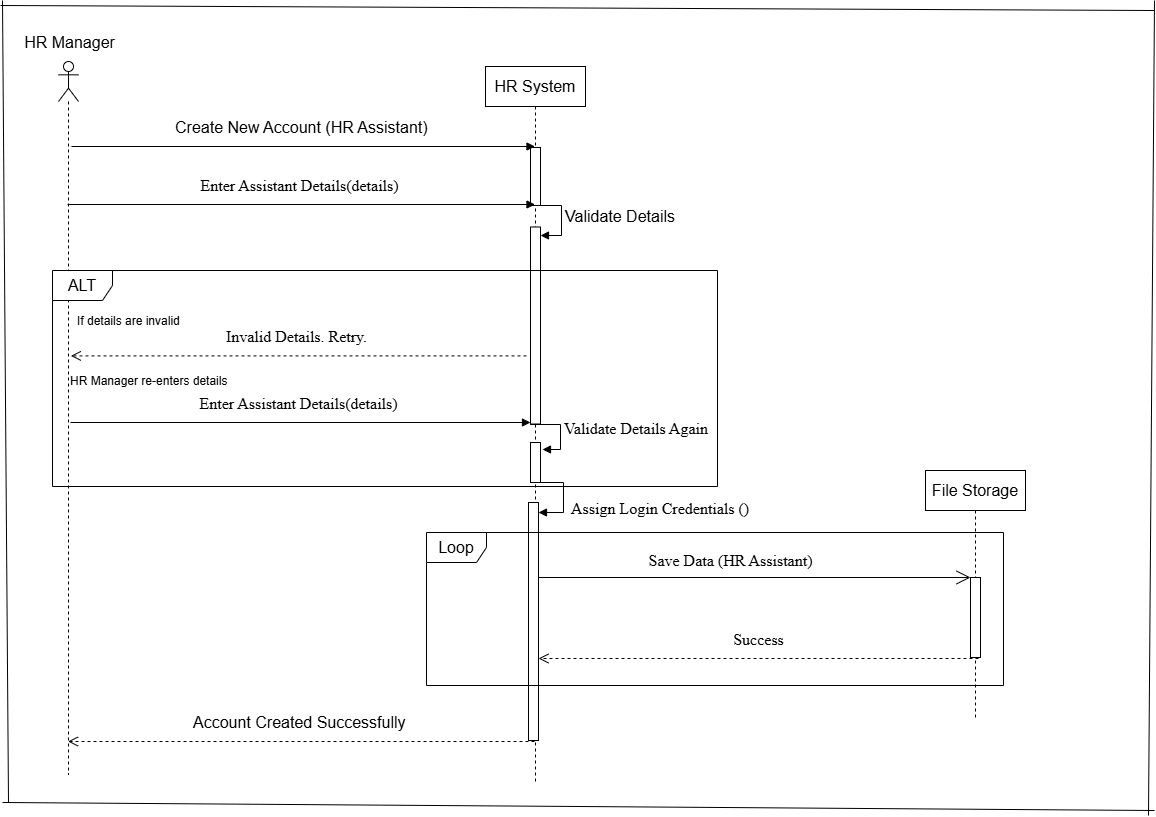


Figure 13 - HR Manager Creating a New Account (HR Assistant) (sequence diagram)

**6**. **HR Manager Logout**

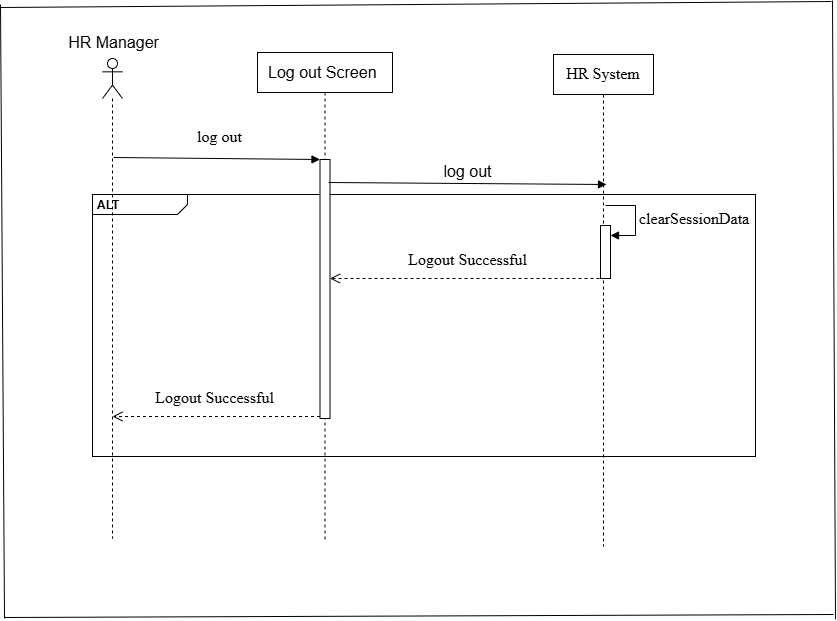


Figure 14 - HR Manager Logout (sequence diagram)

**7.** **Login Process HR Assistant**

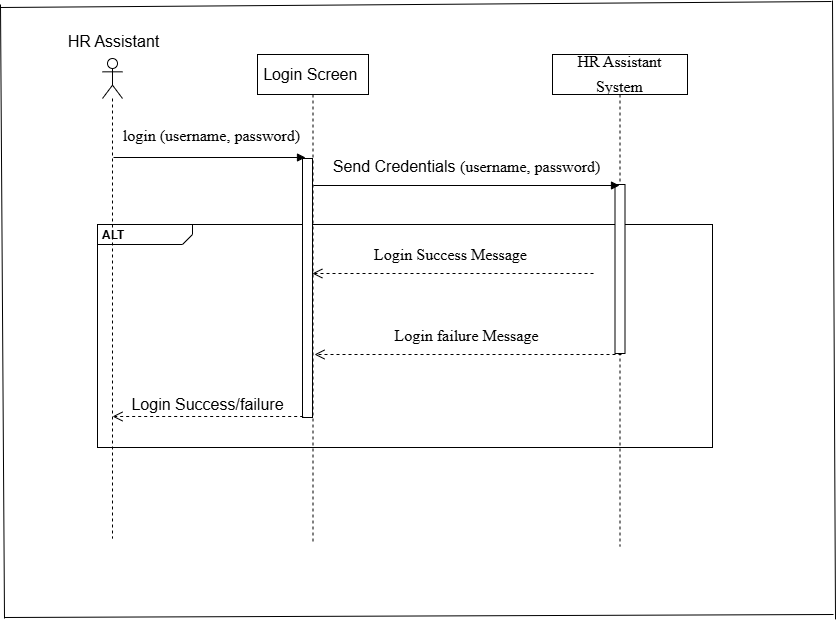


Figure 15 - Login Process HR Assistant (sequence diagram)

**8. HR Assistant Searching for Employees**

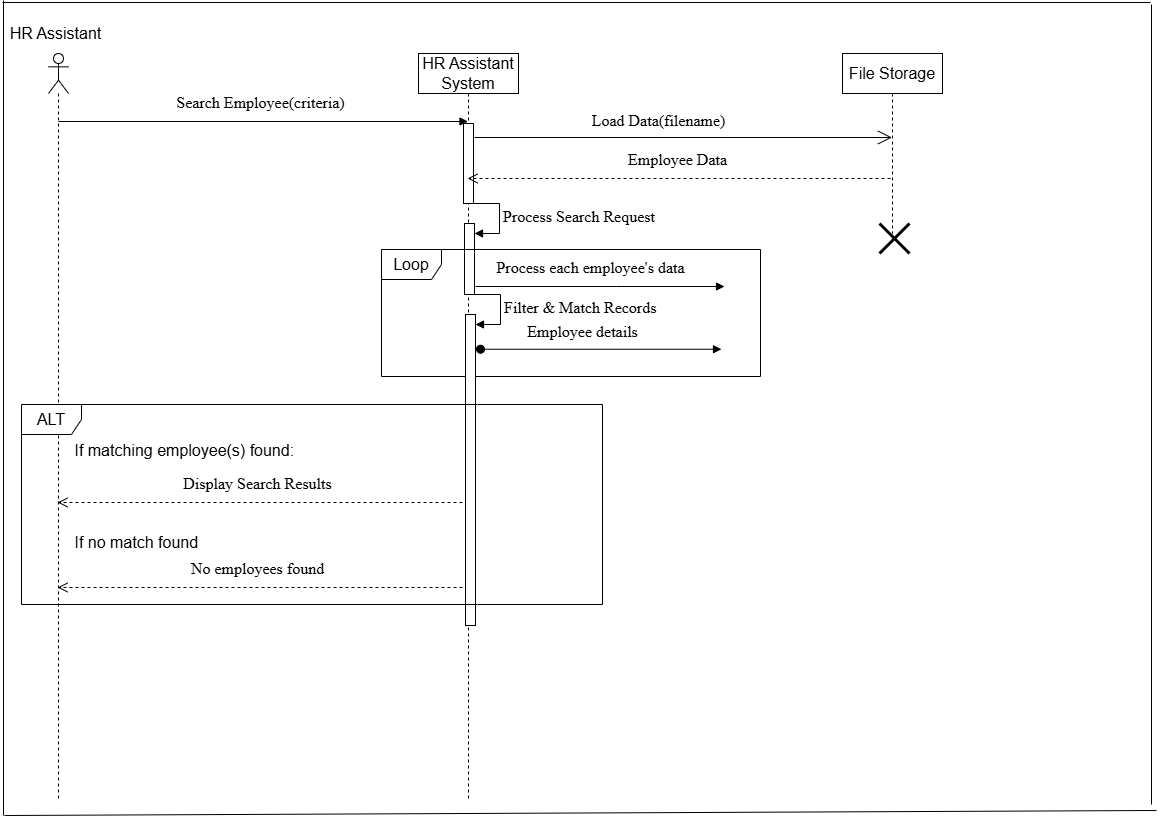


Figure 16 - HR Assistant Searching for Employees (sequence diagram)

**9.** **HR Assistant Logout**

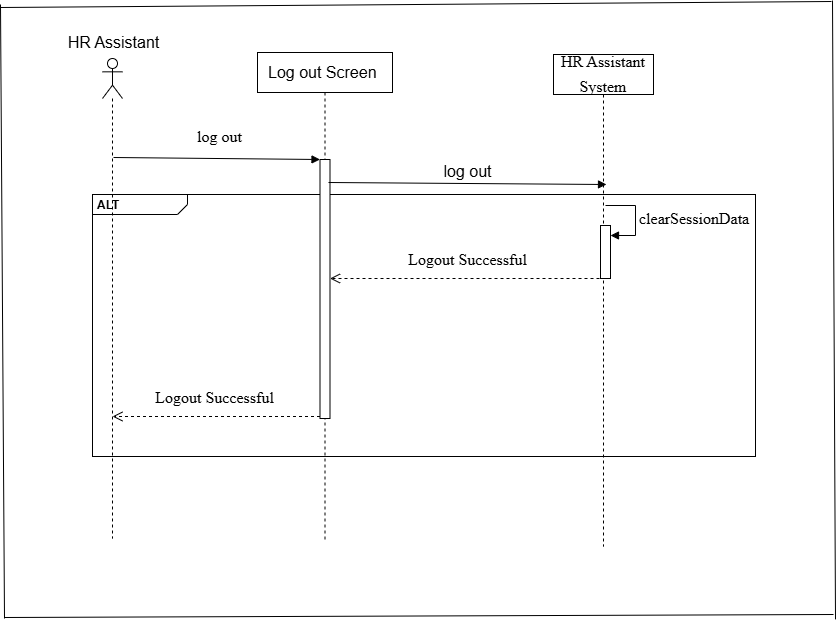


Figure 17 - HR Assistant Logout (sequence diagram)

## 1.11 ****Justification for Sequence Diagrams****

The sequence diagrams provide a clear and structured representation of interactions between actors (HR Manager and HR Assistant) and system components, ensuring a well-defined flow of actions and data exchange. By depicting synchronous, asynchronous, and self-messages, along with control elements like loops and alternative paths (ALT frames), these diagrams effectively illustrate login, employee management, account creation, and logout processes. This enhances system clarity, ensures logical process flow, and aids developers in implementing precise functionality while maintaining consistency in user interactions within the HR automation system.

(visual-paradigm, 2024)

# Task 02 - Object-Oriented Concepts in Connect-me

## ****2.1 Introduction****

The project is built using **Object-Oriented Programming (OOP)**, which makes it easier to manage and expand. Instead of writing everything in one place, OOP helps organize code into **classes and objects**, making it reusable and structured.

In this system, different parts of HR management—like handling employees, departments, and accounts—are split into separate **classes**, each with specific responsibilities. The project also follows key OOP principles:

* **Encapsulation** keeps important data private and only allows access through specific methods.
* **Abstraction** hides complex details, so users interact with simple, clear functions.
* **Inheritance** lets one class reuse the features of another, avoiding duplicate code.
* **Polymorphism** allows methods to work differently based on the situation, like handling button clicks in various forms.

Using these OOP concepts makes this system **efficient, organized, and easy to maintain**. Now, let's see how each principle is applied in your project.

## 2.2 Classes and Objects

The project is built using multiple Java classes, where each class has a specific role. These classes work together to handle different tasks, making the system organized and easy to manage. By using objects, which are created from these classes, the program can store and process data efficiently. This approach helps keep the code structured and reusable.

Examples:

* Sign in
* Login
* HR Manager Dashboard
* HR Assistant Dashboard
* Manage Employee
* Add Departments and Designation
* Search Employees
* Create New Account
* File Storage

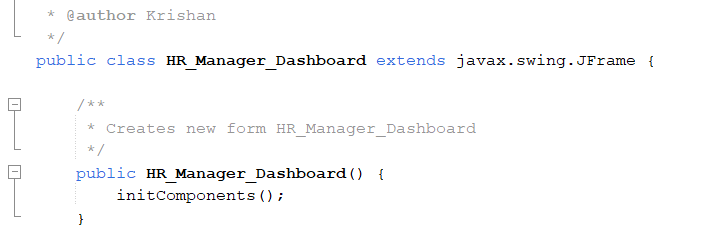


Figure 18 - Class in the program

Here, HR Manager Dashboard is a **class**, and when instantiated (new HR Manager Dashboard()), it creates an **object** of the dashboard.

Another example:

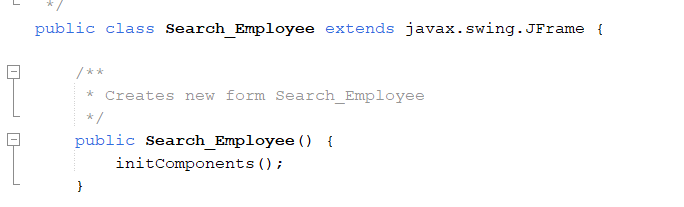


Figure 19 - Class in the program

## 2.3 Encapsulation

Encapsulation is the process of keeping variables and methods private and allowing controlled access through public methods.

The program encapsulates UI components using private attributes.

Example from File\_Storage.java:

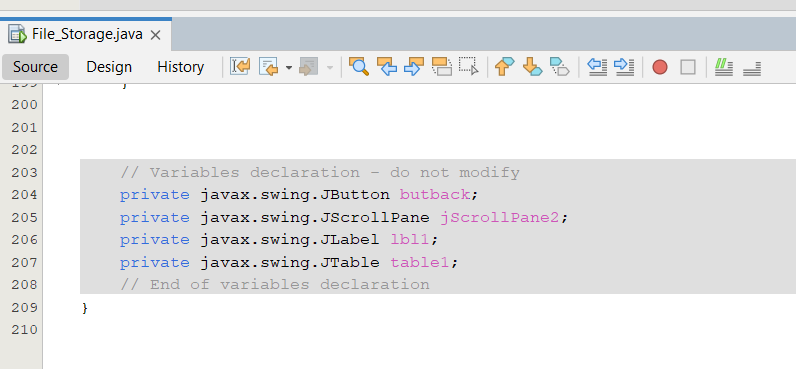


Figure 20 - Encapsulation

These are private attributes and cannot be directly accessed outside the class.

## 2.4 Abstraction

Abstraction in the project means hiding the complex details of the user interface (UI) and only showing what is necessary. Use Java Swing components like buttons, text fields, and panels to achieve this. These components handle the low-level details, such as how clicks and inputs work, and focus on designing a clear and simple interface for users.

Example: The initComponents () method initializes UI elements but hides details.

Example: Add Department and Designation

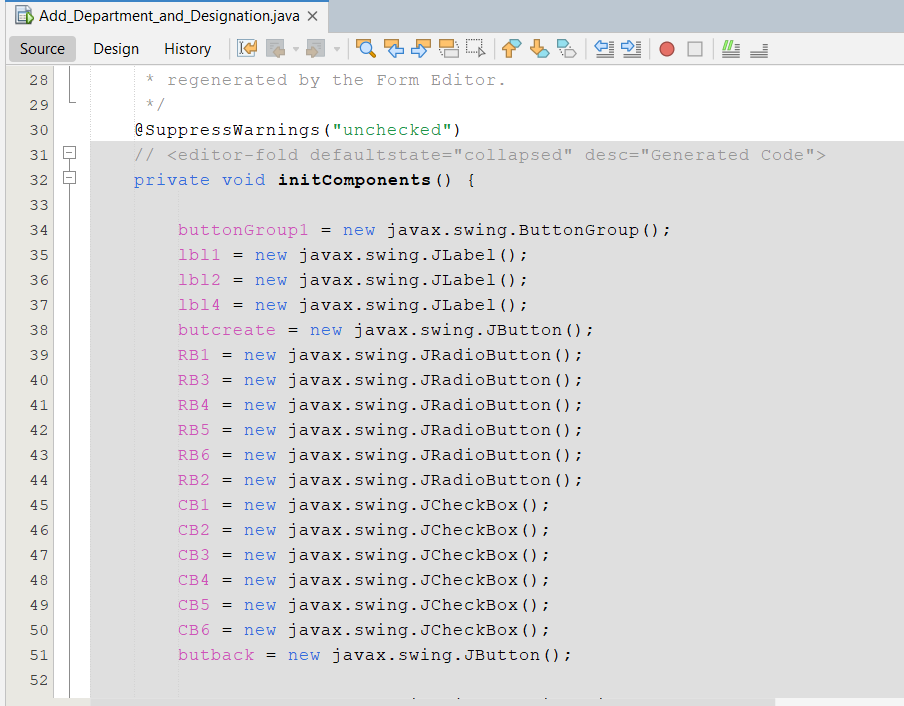


Figure 21 – Abstraction

The internal UI behavior is abstracted from the user.

## 2.5 Inheritance

Inheritance means that GUI classes get features from javax. swing. Frame. This allows them to use built-in properties and methods for displaying and managing the user interface without writing extra code. It helps in reusing existing functionality and makes the program more organized.

Example:

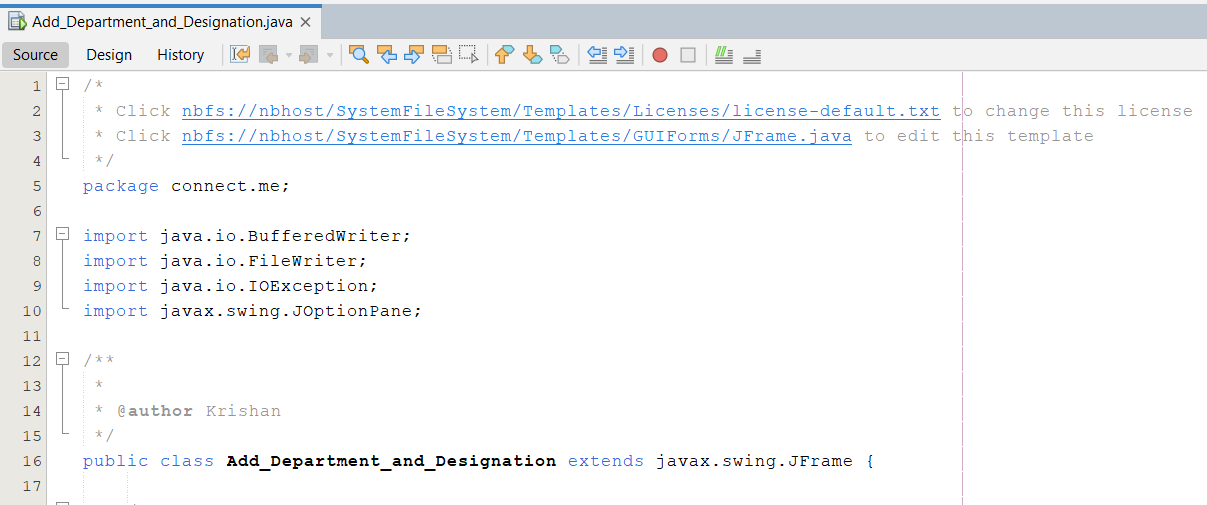


Figure 22 – Inheritance

This means Add Department and Designation inherits JFrame properties without needing to redefine them.

## 2.6 Polymorphism

Polymorphism in the HR Assistant Dashboard project allows the same method to work differently based on the situation. For example, event handling uses overridden methods, meaning the same function name is used in different ways depending on the action taken. This makes the system more flexible and easier to manage.

Example:

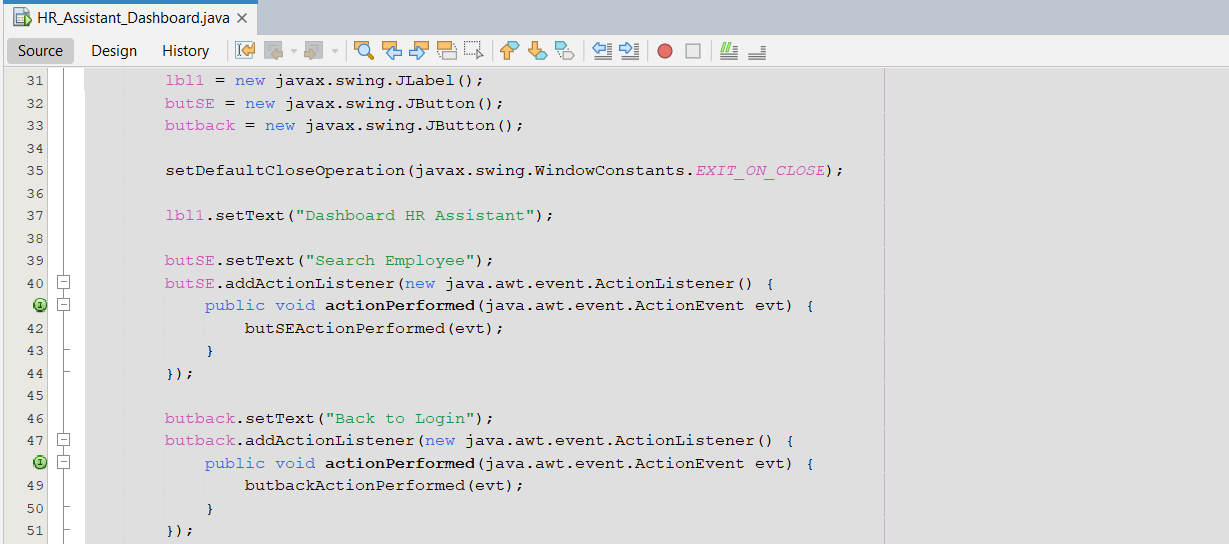


Figure 23 – Polymorphism

## 2.7 Array

Arrays are used in many parts of your project to store and manage data efficiently. They help organize related information, such as employee records, department lists, and account details. For example, an array might store all employee names or track different department IDs. This makes it easier to access, update, and process data quickly within the system.

Example:

### ****1. String Array for Department and Designation (Add\_Department\_and\_Designation.java)****



Figure 24 - String Array for Department and Designation

This defines a string array val with a fixed size of 6, which can store department or designation names.

### ****2. Splitting a String into an Array (Create\_New\_Account.java & Manage\_Employee.java)****

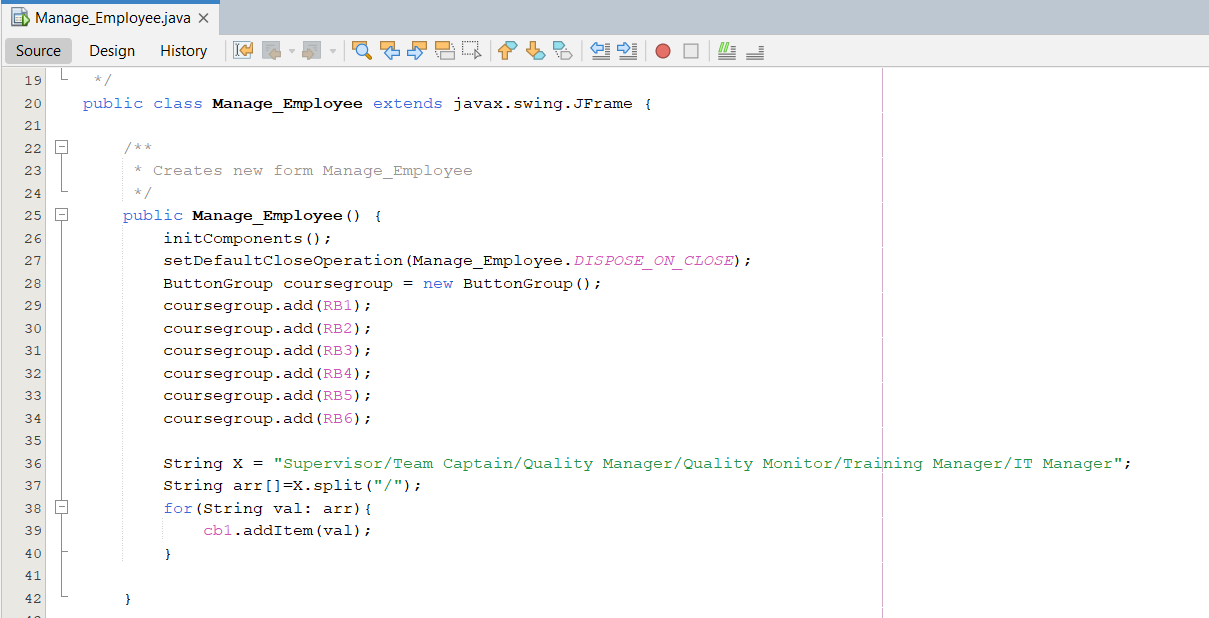


Figure 25 - Splitting a String into an Array

The split ("/") method breaks the string X into an array, where each role is stored as an element in arr.

## 2.8 File Handling in Java

In Java, file handling refers to saving and retrieving data from files on the computer. This uses classes from the java.io package. For example, File helps you work with files, while File Writer and BufferedWriter are used to write data into a file. On the other hand, FileReader and BufferedReader are used for reading data from a file. These classes allow your program to store information in files and load it whenever needed.

### 2.8.1 Writing and Reading Employee Data from a File

1. Writing Employee Data to a File (EmployeeData ())

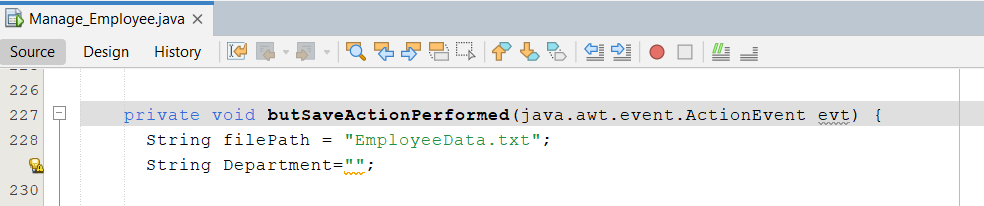


Figure 26 - File path

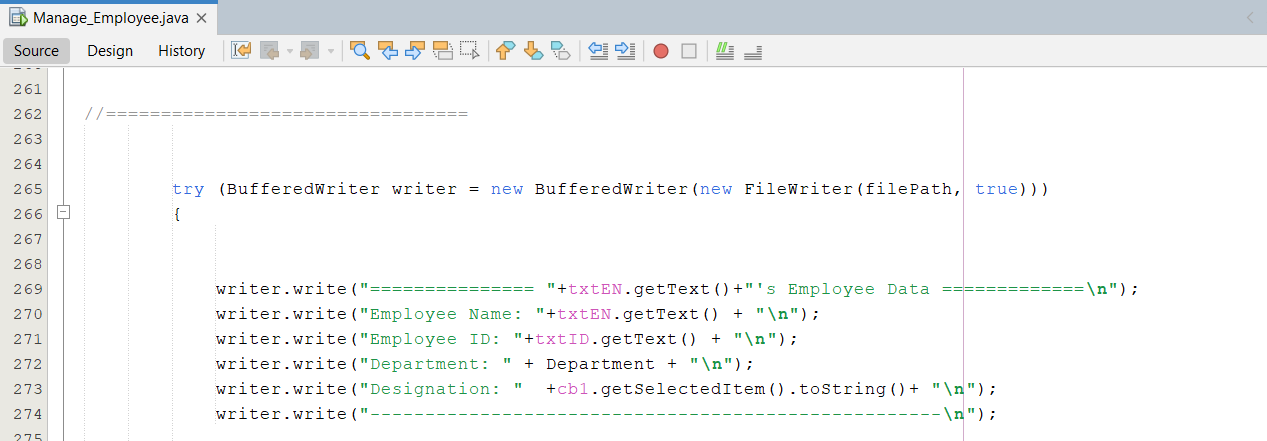


Figure 27 – BufferedWriter

2. Reading Employee Data to a File (EmployeeData ())

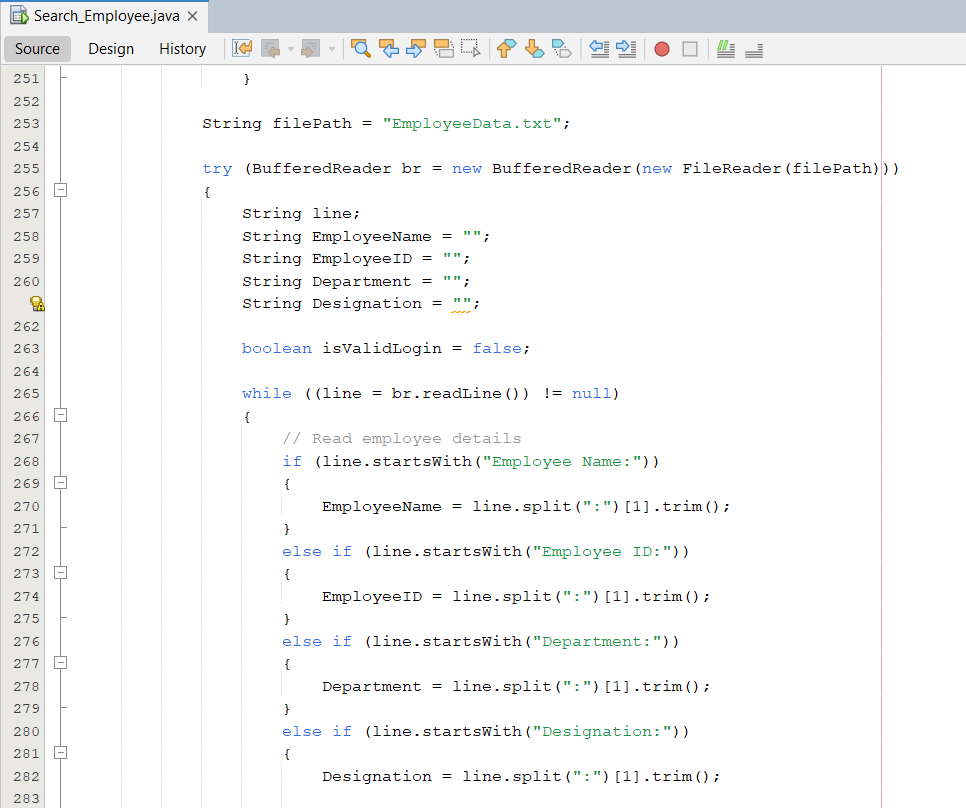


Figure 28 – BufferedReader

### 2.8.2 Example Output

Contents of EmployeeData.txt

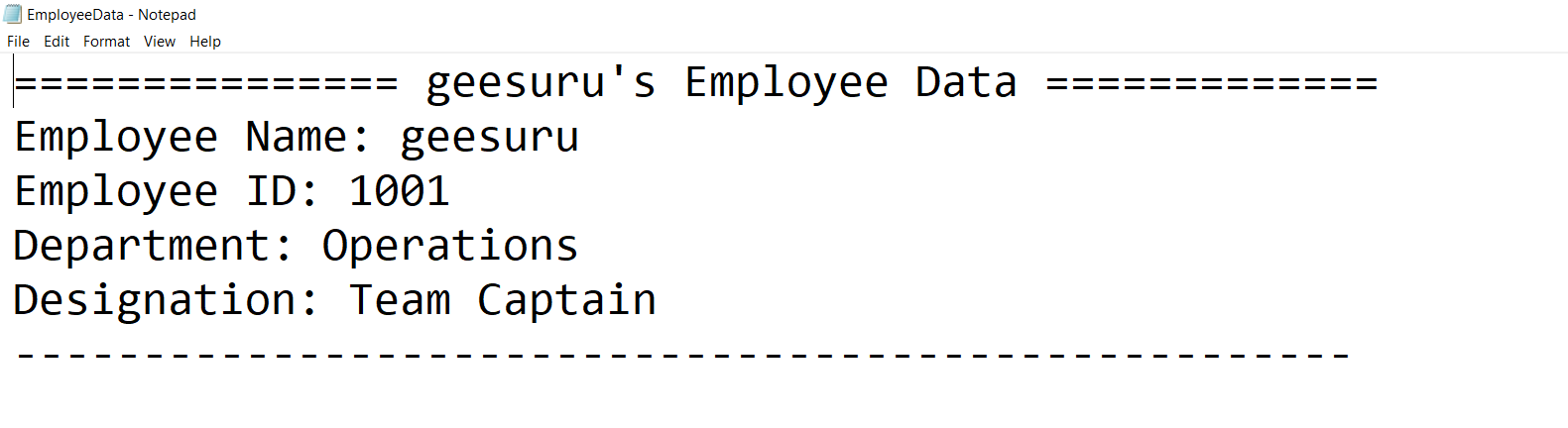


Figure 29 - Example Output

## 2.9 Error Handling in Java

In Java, error handling is managed through a system called exception handling. When a program encounters an unexpected situation, such as trying to open a file that doesn't exist, it throws an exception, which is a special object that interrupts the normal flow of the program. To handle these exceptions and prevent the program from crashing, Java uses try and catch blocks. You place the code that might cause an exception inside a try block, and then use one or more catch blocks to define how to respond to different types of exceptions. Additionally, a finally block can be used after the catch blocks to include code that will always run, regardless of whether an exception was thrown or not, often for cleanup activities like closing files or releasing resources

Example in Search Employee

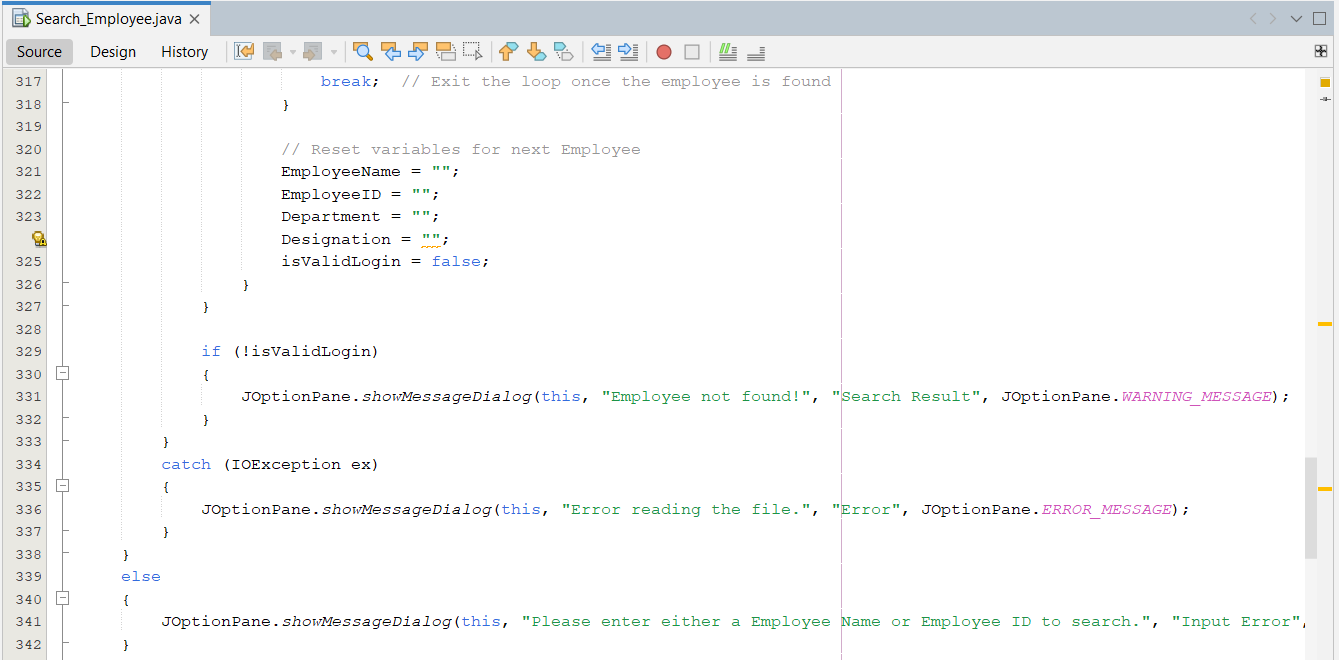


Figure 30 - Error Handling

(Herrity, 2025)

# Task 03 - ****User Manual for 'Connect Me' Project****

## 3.****1. Overview****

The 'Connect Me' HR Management System is designed for HR Managers and Assistants to manage employee details, departments, and user accounts.

## 3.****2 System Requirements****

* Java Development Kit (JDK) installed
* NetBeans IDE for execution

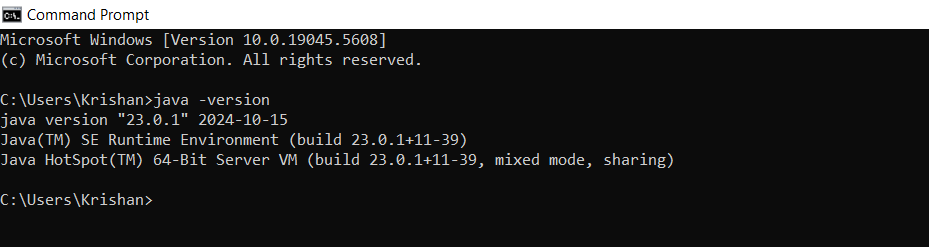


Figure 31 - Java Development Kit (JDK) installed

## 3.3 ****Installation & Setup****

* Open NetBeans and load the project.
* Build and run the project.
* Ensure required text files (EmployeeData.txt, loginInfo.txt, Department & Designation) are in place.

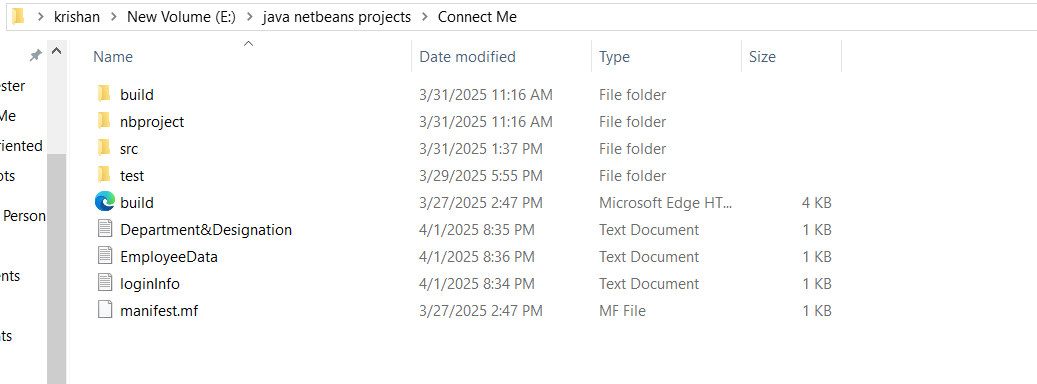


Figure 32 - required text files

## 3.4 ****Features & Usage****

### 3.4.1 ****Login Page****

* Enter username and password.
* Click **Login** to access the system.

### 

Figure 33 - Login Page

### 3.4.2 Sign in Page

* Create Admin (HR Manager)

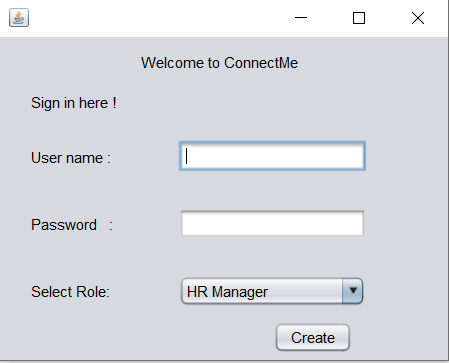


Figure 34 - Sign in Page

### 3.4.3 ****HR Manager Dashboard****

* Add departments and designations.
* Manage employee records.
* Create New Account
* Search Employees
* Logout functionality.

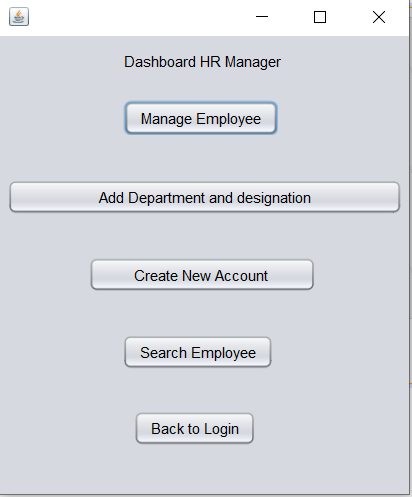


Figure 35 - HR Manager Dashboard

### 3.4.4 ****HR Assistant Dashboard****

* Search employee details.
* Logout functionality.

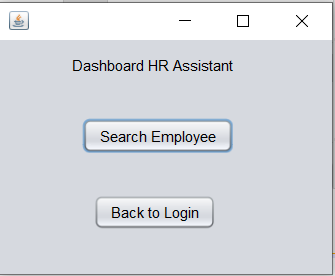


Figure 36 - HR Assistant Dashboard

### 3.4.5 Add Departments and Designations

Adding recommended Designations to each Department on this page.

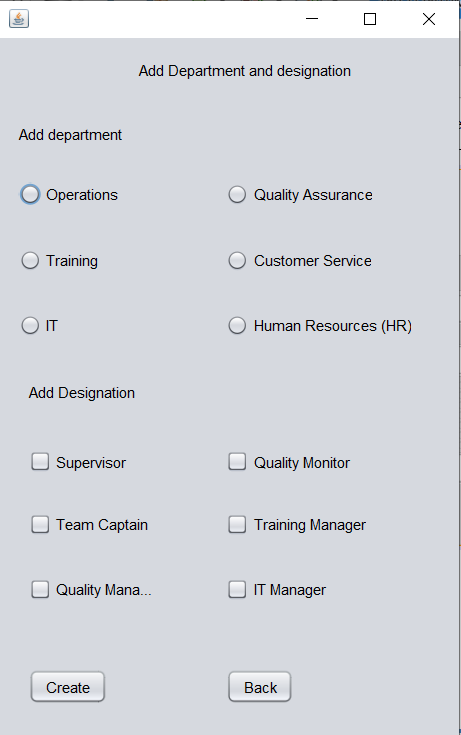


Figure 37 - Add Departments and Designation

### 3.4.6 Manage Employee

Add a new Employee with name and ID on this page and allocate the employee to the Department and assign the designation, and save it as EmployeeData.txt, and we can also go to the file storage page.

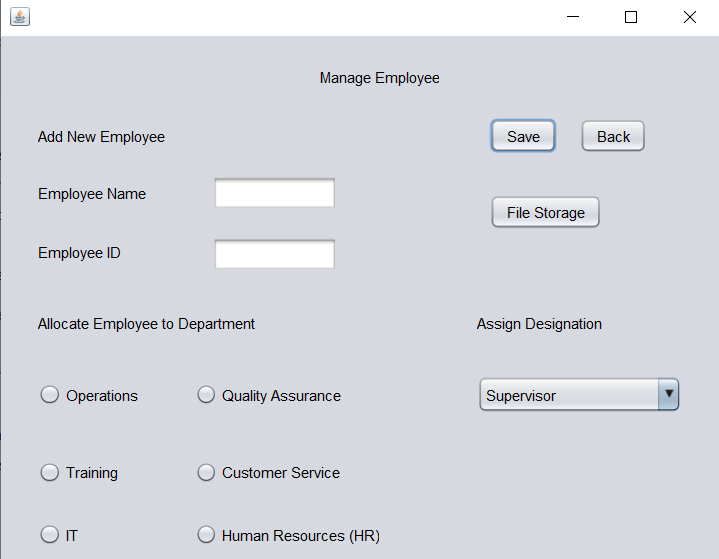


Figure 38 - Manage Employee

### 3.4.7 Create a new Account

Create a new account with the username and password on this page and save it as a LoginInfo.txt file by choosing a designation role.

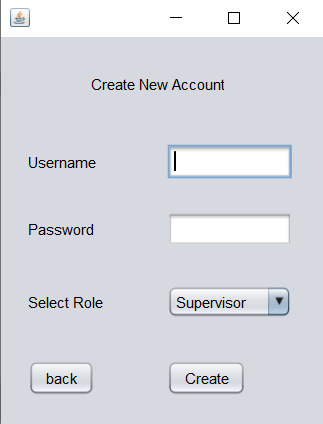


Figure 39 - Create a new Account

### 3.4.8 Search Employee

This page can search for an employee by name or by department. when you select the by name button, the department button will automatically close, and you can update and save the details when the employee details are received.

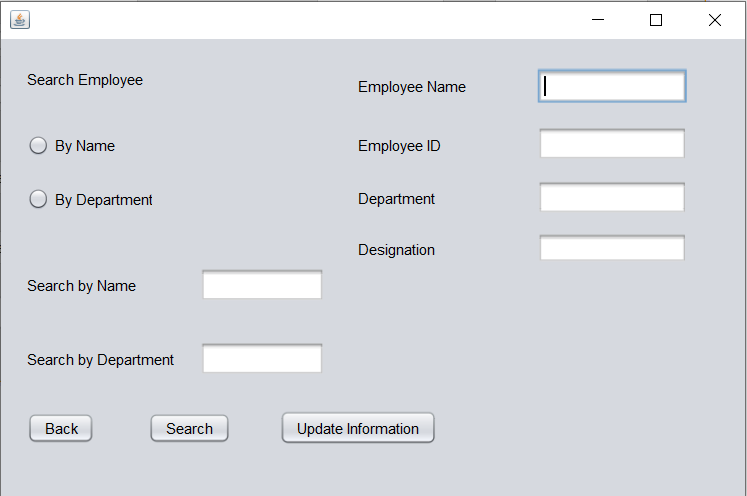


Figure 40 - Search Employee

### 3.4.9 File Storage

Load Employee details as a table on this page.

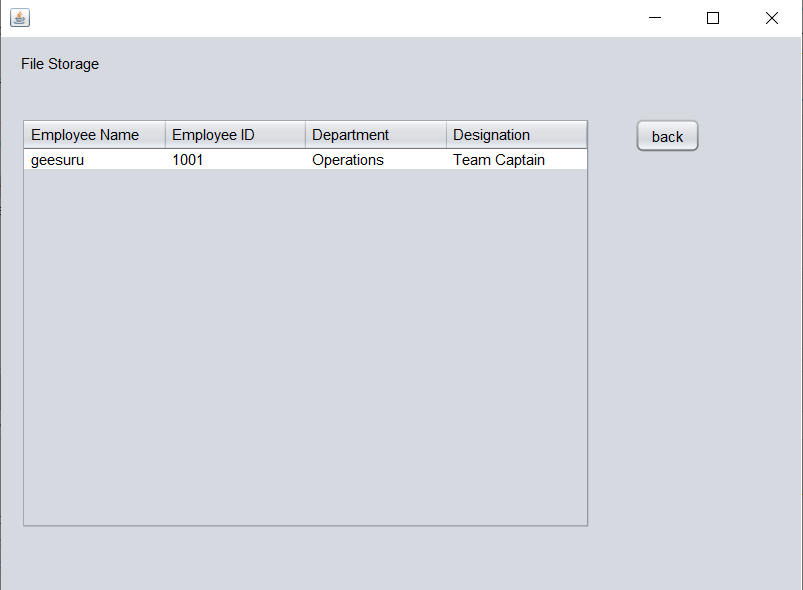


Figure 41 - File Storage

### 3.5 ****Troubleshooting****

* If the application does not run, ensure Java and NetBeans are properly installed.
* If login fails, verify that loginInfo.txt contains valid credentials.

# Conclusion

The "Connect Me" HR Management System successfully demonstrates the power of Object-Oriented Programming (OOP) by applying core concepts such as encapsulation, abstraction, inheritance, and polymorphism. The solution has been developed using Java, with a well-structured design that ensures modularity and scalability. Key tasks in the project included:

We create simple diagrams like use case, class, and sequence diagrams to show how the system works and how different parts interact. We also define clear roles for HR Managers and HR Assistants, making sure that each person only accesses the parts of the system they are allowed to.

The program uses object-oriented programming ideas to make managing employee data easier and safer. It hides sensitive data inside private attributes and only lets it be changed through specific methods, keeping the information secure. It also hides the complicated parts of the user interface behind simple and clear designs. The code reuses common parts by extending built-in components like JFrame, and it allows the same method to act differently in different situations. Additionally, it uses arrays to store lists of departments and designations, and it reads from and writes to text files to save employee details and login information.

A detailed user manual was developed that covers installation, system requirements, navigation through various dashboards (HR Manager and HR Assistant), and the specific functionalities like login, account creation, employee management, and file storage. Screenshots of key interfaces provide users with visual guidance, making it easier to understand and operate the system.

# Recommendation

* Improve Error Handling:
  + Expand exception handling across the entire application to ensure that all potential runtime errors (such as file access issues or invalid input) are gracefully managed. Consider logging errors to a file for further diagnostics.
* User Interface Enhancements:
  + Modernize the GUI with updated layouts and styles for a better user experience. Implement responsive design elements to accommodate various screen sizes.
  + Provide in-app tooltips or a help section that guides users on how to use each feature.
* Security Measures:
  + Integrate stronger authentication methods and consider encrypting sensitive data (e.g., passwords) before storing them.
  + Regularly update the system with security patches to protect against vulnerabilities.
* Code Maintenance and Documentation:
  + Maintain clear inline comments and documentation within the codebase. This is critical for future developers who might work on the system.
  + Consider adopting version control practices (if not already implemented) to track changes and facilitate collaboration.

# References

Bell, D., 2023. *The UML 2 class diagram.* [Online]   
Available at: https://developer.ibm.com/articles/the-class-diagram/  
[Accessed 27 March 2025].

Herrity, K., 2025. *What Is Object-Oriented Programming (OOP)? A Complete Guide.* [Online]   
Available at: https://www.indeed.com/career-advice/career-development/what-is-object-oriented-programming  
[Accessed 31 March 2025].

lucidchart, 2025. *UML Use Case Diagram Tutorial.* [Online]   
Available at: https://www.lucidchart.com/pages/uml-use-case-diagram  
[Accessed 26 March 2025].

visual-paradigm, 2024. *What is Sequence Diagram?.* [Online]   
Available at: https://www.visual-paradigm.com/guide/uml-unified-modeling-language/what-is-sequence-diagram  
[Accessed 28 March 2025].