



# INDUSTRIAL ACCOUNT MANAGEMENT SYSTEM

Course Title: Software Engineering Lab  
Course Code: (ICT 4354)

## SUBMITTED TO:

Farzana Akter  
Lecturer,  
Department of ICT, BDU

Md. Habibur Rahman  
Lecturer,  
Department of ICT, BDU.

## Submitted BY:

Sharika Khan  
ID: 1901013  
Tasnima Hamid  
ID: 1901043  
Session: 2019-20  
Third Year First Semester  
Internet of Things(IoT) Program  
Department of ICT

## Abstract

The industrial account management system is designed to streamline financial operations within an organization. The system will allow the Managing Director to oversee the overall financial performance of the organization, while Income Manager and Cash Manager will have specific access and control over the income and cash sections of the system respectively.

## Acknowledgements

I would like to express my sincere gratitude to our teacher Farzana Akter ma'am and Md. Habibur Rahman sir for their guidance, comments and advice throughout the research. Their wisdom and insight have considerably improved the quality of this dissertation. They have a great personality, being very considerate and had never thinks twice for helping people. We feel very lucky and honored to be their student and being having them as our course teacher. Their experience and suggestion made the whole process more very constructive and manageable.

## Table of Contents

Abstract .....	1
Acknowledgements.....	1
<b>Introduction .....</b>	<b>5</b>
<i>Introduction.....</i>	<i>5</i>
<i>Problem Statement .....</i>	<i>5</i>
<i>Objective .....</i>	<i>5</i>
<i>Motivation.....</i>	<i>6</i>
<b>Project Management Plan.....</b>	<b>6</b>
<i>Project Organization .....</i>	<i>6</i>
<i>Lifecycle Model Used.....</i>	<i>6</i>
<i>Risk Analysis .....</i>	<i>7</i>
<i>Hardware and Software Resource Requirements .....</i>	<i>7</i>
<i>Deliverables and Schedule .....</i>	<i>7</i>
<i>Monitoring, Reporting, and Controlling Mechanisms.....</i>	<i>8</i>
<i>Impact of the project on individuals and organizations.....</i>	<i>8</i>
<b>Requirement Specification .....</b>	<b>9</b>
<i>Stakeholders for the system .....</i>	<i>9</i>
<i>Use case model .....</i>	<i>10</i>
Graphic Use Case Model .....	10
Textual Use Case Model .....	10
<i>Rationale for your use case model .....</i>	<i>13</i>
<i>Non-functional requirements .....</i>	<i>13</i>
<b>Architecture .....</b>	<b>14</b>
<i>Architectural style(s) used .....</i>	<i>14</i>
<i>Architectural model.....</i>	<i>14</i>
<i>Technology, software and hardware used .....</i>	<i>14</i>
<i>Rationale for your architectural style and model.....</i>	<i>15</i>
<b>Design .....</b>	<b>15</b>
<i>GUI(Graphical User Interface) Design .....</i>	<i>15</i>
<i>Static model - class diagrams.....</i>	<i>21</i>
<i>Dynamic model - sequence diagrams .....</i>	<i>22</i>
<i>Rationale for your detailed design model .....</i>	<i>23</i>

<i>Traceability from requirements to detailed design .....</i>	<i>23</i>
<b>Test Plan .....</b>	<b>24</b>
<i>    Requirements/specifications-based system level test cases .....</i>	<i>24</i>
<b>Cost .....</b>	<b>32</b>
<i>    Cost estimation .....</i>	<i>32</i>
<b>Result .....</b>	<b>33</b>
<i>    Result.....</i>	<i>33</i>
<b>Conclusion and Discussion .....</b>	<b>33</b>
<i>    References .....</i>	<i>33</i>

Figure 1 Login Page .....	15
Figure 2 Income Section landing page .....	16
Figure 3 View Income transactions .....	16
Figure 4 Cash Section landing page.....	17
Figure 5 View Accepted bills .....	17
Figure 6 View Declined bills .....	18
Figure 7 Managing Section landing page .....	19
Figure 8 Add employee page .....	20
Figure 9 View employee info, cash transactions and income transactions .....	21
Figure 10 Successful login .....	25
Figure 11 Entry added successfully .....	26
Figure 12 Entry added successfully .....	27
Figure 13 Data added successfully .....	29
Figure 14 Data deleted successfully .....	30
Figure 15 Employee successfully added.....	32

Table 1 Test case for login .....	24
Table 2 Test case for Income section page.....	26
Table 3 Test case for Cash section page .....	27
Table 4 Test case for managing section page .....	28
Table 5 Test case for add employee page.....	31

# Introduction

## *Introduction*

Keeping track of all revenues, expenses, and other financial transactions is made easier for organizations by using an accounting management system or accounting management software. It also enables business owners to obtain the information they require for conducting business. An internal management accounting system's principal goal is to give managers information they can use to make informed decisions. The objective is to gather useful information that may be used to better manage and oversee a business, not to comply with external demands like those of bankers. Industrial managers make plans for how to use a company's resources, such as personnel, materials, machinery, time, capital, energy, and information, effectively and inexpensively.

## *Problem Statement*

Industries or organizations deal with a huge amount of money daily. Thus, they must manage multiple accounts with multiple transactions and keep records of them, which can be error prone. As these huge amounts of transactions are managed by humans now, there can be miscalculations or corruption in the system and the industry/organization can suffer from this. To avoid these, we intend to invent a software which will keep clean records and safely store all the transaction records of all the accounts, daily earnings and expenditures of the industry/organization for future reference.

## *Objective*

This project will be admin monitored, who will store daily earning and expenditures records. Thus, the project will hold daily records from which a monthly insight can be achieved. This software will keep all the financial records. This project will be easy to use, low-cost software. The project can easily be used in industries, which will pave the way for digitalization.

- Keep clean and digital records of transactions

- User friendly interface
- More accurate calculations
- Enriched database
- Easy lookup for data
- Easy and effective fund management
- Automation
- Real-time insight into expenditures
- Effective budgeting

### *Motivation*

The working through registers system is not only very laborious and time consuming, but there are number of errors found. In comparison to old system the proposed system of computer processes many advantages. As a student of ICT department, we want to improve our surroundings. This is a little step to fulfill our dream or aim.

## Project Management Plan

### *Project Organization*

The project team will consist of a project manager, software developers and testers. The Managing Director, Income Manager, and Cash Manager will also provide guidance and feedback during the development process.

### *Lifecycle Model Used*

The project team will follow the Agile methodology for software development to ensure the project is completed in an iterative and incremental manner, with constant feedback and collaboration among team members and stakeholders.

## *Risk Analysis*

Potential risks associated with the project include delays in development, integration issues with existing systems, and security breaches. To mitigate these risks, the project team will implement a risk management plan that includes regular risk assessments and contingency plans.

## *Hardware and Software Resource Requirements*

The industrial account management system will require a server for hosting the software, as well as desktop and mobile devices for accessing the system. The system will be developed using the following software tools and technologies:

Programming Languages: Java, SQL

Database Management System: MySQL Workbench

Project Management: JIRA

We also used software like draw.io, visual paradigm, NetBeans IDE to implement the project.

## *Deliverables and Schedule*

The deliverables and schedule for this 6-month project to develop an industrial account management system:

- Requirements Specification (1 month)
  - Stakeholder Analysis Report
  - Use Case Model
  - Non-Functional Requirements
  - Traceability Matrix
- System Design (1 month)
  - Architectural Design
  - GUI Design
  - Class Diagram
  - Sequence Diagram



- Implementation (2 months)
  - User Interface Development
  - Back-end Development
  - Database Integration
  - Testing
- Testing and Quality Assurance (1 month)
  - Unit Testing
  - Integration Testing
  - System Testing
  - User Acceptance Testing
  - Bug Fixing
- Deployment (1 month)
  - System Deployment
  - Data Migration
  - User Training
- Maintenance and Support (ongoing)
  - Technical Support
  - System Maintenance and Upgrades

### *Monitoring, Reporting, and Controlling Mechanisms*

The industrial account management system will be regularly monitored and updated by the project team to ensure it remains functional and secure. Regular reports will be provided to the Managing Director, Income Manager, and Cash Manager to keep them informed of the system's performance.

### *Impact of the project on individuals and organizations*

The industrial account management system will have a positive impact on individuals and organizations by improving industrial account management efficiency and reducing errors. The Managing Director will have better visibility into the organization's financial performance, while Income Manager and Cash Manager will have specific access and control over their respective

sections of the system. This will lead to more accurate financial reporting and better decision-making. The industrial account management system will also free up time for staff to focus on other critical tasks, leading to increased productivity and ultimately, better financial outcomes for the organization.

## Requirement Specification

### *Stakeholders for the system*

Industry, Income Manager, Cash Manager, Managing Director are the stakeholders of the system. Industry or Industrial Management deals with industrial design, construction, management, and application of science and engineering principles to improve the entire industrial infrastructure and industrial processes. Industrial Management focuses on the management of industrial processes.

Income Manager The income management team of a company is supported by the income manager, who distributes its income resources, oversees budget planning, and who provides income guidance so that they can make the best business decisions possible for the organization.

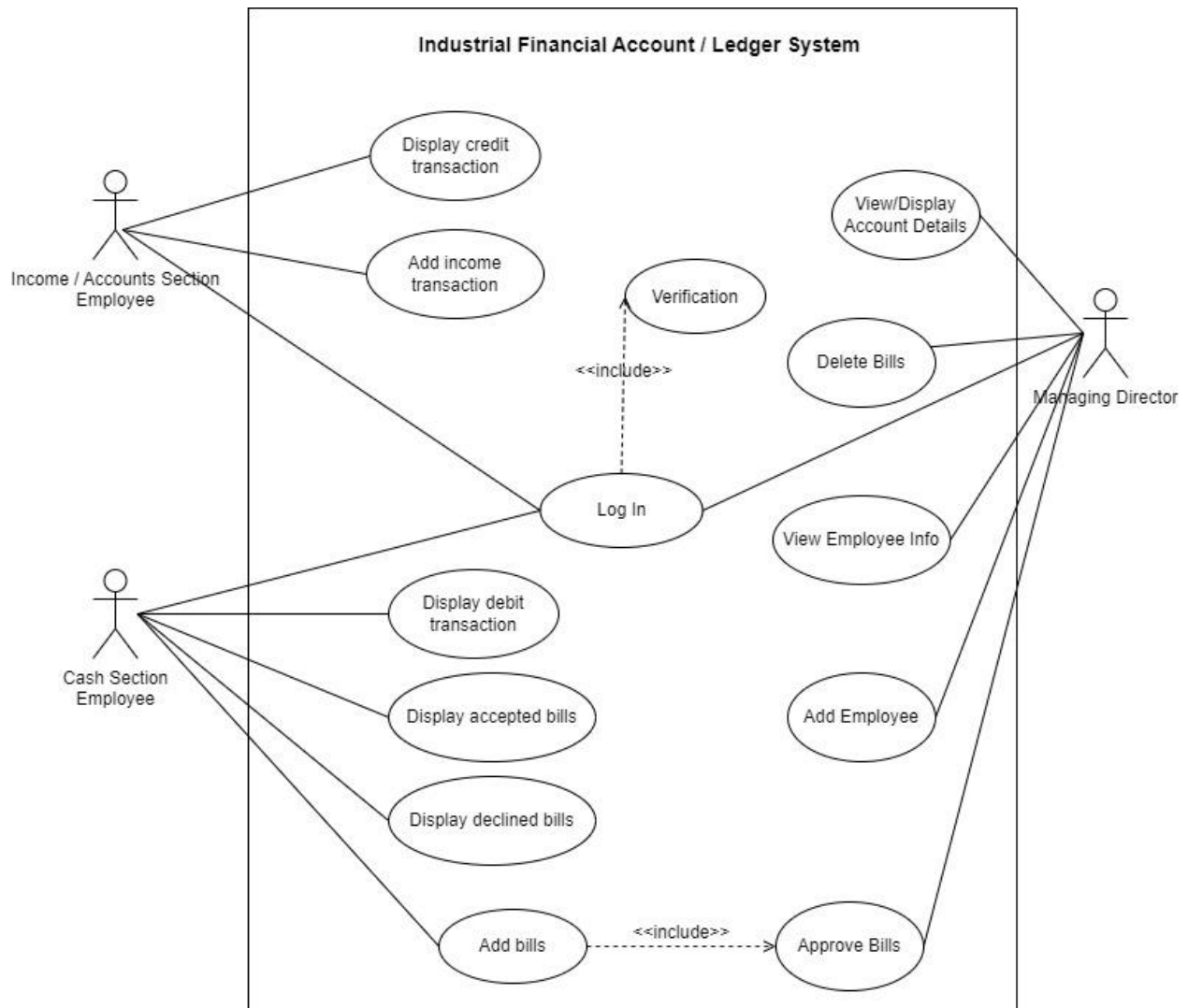
Cash Manager Monitoring cash flow, examining financial transactions, and assigning sufficient funds and resources for each department's operations are all tasks that fall under the purview of a cash manager. Cash managers analyze data and statistics to identify costs and potential financial losses for the organization and plan ways to reduce those risks. Additionally, they assist senior management in locating new prospects that could boost the company's revenue and profitability. A cash manager deals with billing issues, ties up loose ends in accounts, and prepares precise financial reports.

Managing Director Ensure that the appropriate policies are produced to meet the company's mission and objectives and to comply with all applicable statutory and other restrictions. Give strategic counsel and direction to the chair and members of the board.

Developers will also be a stakeholder of the system as they will develop the system.

## Use case model

### Graphic Use Case Model



### Textual Use Case Model

The use case model for the industrial account management system will include the following use cases:

**Actors:** Income Manager, Cash Manager and Managing Director

Income Manager can perform the following functions-

- View/Display income transactions
- Add/Create Income transactions

Cash manager can perform the following functionalities-

- View/Display credit transactions
- Add/Create cash bill records
- View accepted bills
- View declined bills

Managing Director can perform the following functions-

- Approve bills
- Delete bills
- Add Employee
- View Employee Info
- View Income transactions
- View cash bills

### **Functions in details:**

Login:

The user enters their login credentials to access the industrial account management system. The system verifies the credentials and grants access if they are correct.

View income/cash transactions:

The user clicks the button to view the transaction details. The system takes the requests, connects the database and displays in the table.

Create income records:

The Income Manager selects the option to create a new income record. The system prompts the Income Manager to enter the details of the income record, such as the source of income, amount, and date. The Income Manager enters the details and submits the record. The system validates the input and stores the income record in the database.

Create cash records:

The Cash Manager selects the option to create a new cash record. The system prompts the Cash Manager to enter the details of the cash record, such as the source of cash, amount, and date. The Cash Manager enters the details and submits the record. The system validates the input and stores the cash record in the database.

Approve bills:

The Managing Director selects the option to update an existing cash record. The system displays a list of available cash records. Managing Director makes the necessary changes to the record and submits the changes. The system validates the input and updates the cash record in the database.

Delete income records:

The system displays a list of available cash records. The Managing Director selects the record they want to delete. The system deletes the income record from the database after the managing director clicks delete button.

Add Employee:

The Managing Director clicks add employee, then a form opens. Managing Director fills up the form and submits the info in database.

View Employee Info:

The Managing Director clicks and can view the employee info.

### *Rationale for your use case model*

The use case model was designed to reflect the key functionalities of the industrial account management system and the roles of the stakeholders. The model is intuitive and easy to use, allowing stakeholders to quickly and efficiently complete their tasks within the system.

### *Non-functional requirements*

**Performance:** The system should be able to handle many concurrent users and transactions without experiencing any performance issues.

**Usability:** The system should have a user-friendly interface and be easy to navigate for all stakeholders.

**Scalability:** The system should be designed to handle future growth and changes in the organization's financial needs.

**Accessibility:** The system should be accessible to all stakeholders, including those with disabilities, and should comply with accessibility guidelines.

**Compatibility:** The system should be compatible with a range of devices, browsers, and operating systems commonly used by stakeholders.

**Maintainability:** The system should be easy to maintain and update, with clear documentation and processes in place for making changes and resolving issues.

**Data Backup and Recovery:** The system should have regular backups and a disaster recovery plan in place to minimize the risk of data loss and ensure continuity of service in case of any unforeseen events.

**Compliance:** The system should comply with all relevant laws, regulations, and industry standards governing industrial account management systems.

Performance Metrics: The system should have metrics in place to monitor system performance and identify any bottlenecks, with a plan to optimize system performance where necessary.

## Architecture

### *Architectural style(s) used*

The proposed industrial account management system will use a client-server architecture. This architecture style will allow for separation of concerns, scalability, and ease of maintenance.

### *Architectural model*

The proposed system will be designed using the Model-View-Controller (MVC) architectural pattern. This pattern will separate the concerns of data storage and management, user interface, and application logic. The model will be responsible for managing the data and enforcing the business rules. The view will handle the user interface and presentation of data, while the controller will act as an intermediary between the model and view.

### *Technology, software and hardware used*

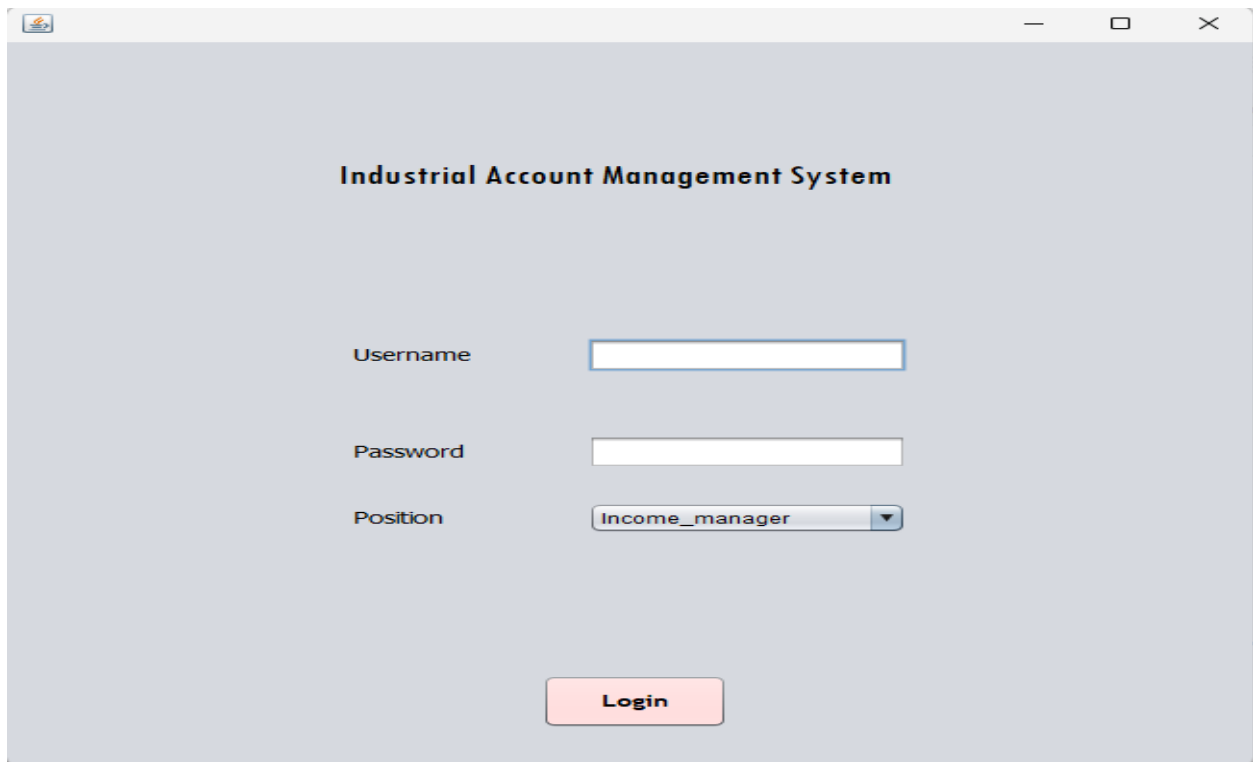
The proposed system will be developed using Java programming language, MySQL database management system, and a localhost server. The system will be developed using NetBeans and will run on a Windows operating system.

## *Rationale for your architectural style and model*

The client-server architecture was chosen because it separates the concerns of data storage and management, business logic, and user interface. This separation will allow for scalability, ease of maintenance, and the ability to easily modify each tier without impacting the others. The MVC pattern was chosen because it promotes separation of concerns and allows for flexibility in development. The use of Java and MySQL was chosen because they are well-established and widely used technologies that offer reliability, security, and scalability. The use of a localhost server was chosen for ease of development and testing.

## Design

### *GUI(Graphical User Interface) Design*



The image shows a web browser window displaying the login page of the 'Industrial Account Management System'. The page has a light gray background. At the top center, the title 'Industrial Account Management System' is displayed in a bold, black font. Below the title, there are three input fields arranged vertically. The first field is labeled 'Username' and is a simple text input. The second field is labeled 'Password' and is also a text input. The third field is labeled 'Position' and is a dropdown menu with 'Income\_manager' selected. Below these fields, there is a red 'Login' button with white text. The browser window has a standard title bar with a small icon on the left and minimize, maximize, and close buttons on the right.

*Figure 1 Login Page*



# Income Section

View Transactions

Cheque No

Amount

Date Issued

Source

Add Records

ID	Cheque_No	Amount	Date Issued	Income_source
----	-----------	--------	-------------	---------------

Sign Out

Figure 2 Income Section landing page

# Income Section

View Transactions

Cheque No

Amount

Date Issued

Source

Add Records

ID	Cheque_No	Amount	Date Issued	Income_source
1	0104-258963	5000.0	2023-03-01	Selling goods
2	0103-2587410	7000.0	2022-02-28	Profit
3	0107-259763	9510.0	2023-03-18	Selling goods

Sign Out

Figure 3 View Income transactions

Cash Section

Cheque No

Amount

Date Added

Add Bills

Sign Out

View All Records

View Accepted Bills

View Declined Bills

ID	Cheque no	Amount	Date Issued	Status
1	0104-25312A	5000.0	2023-03-01	Pending
5	0103-45325A	7000.0	2023-03-03	Declined
6	0102-34217	7000.0	2023-03-07	Accepted
7	0108-2587413	7530.0	2023-03-17	Pending
8	0108-2587753	4590.0	2023-03-17	Pending
9	0107-78541B	1560.0	2023-03-18	Pending

Figure 4 Cash Section landing page

Cash Section

Cheque No

Amount

Date Added

Add Bills

Sign Out

View All Records

View Accepted Bills

View Declined Bills

ID	Cheque no	Amount	Date Issued	Status
1	0102-34217	7000.0	2023-03-07	Accepted

Figure 5 View Accepted bills

Cash Section

Cheque No

0107-78541B

Amount

1560.0

Date Added

2023/03/18

Add Bills

View All Records

View Accepted Bills

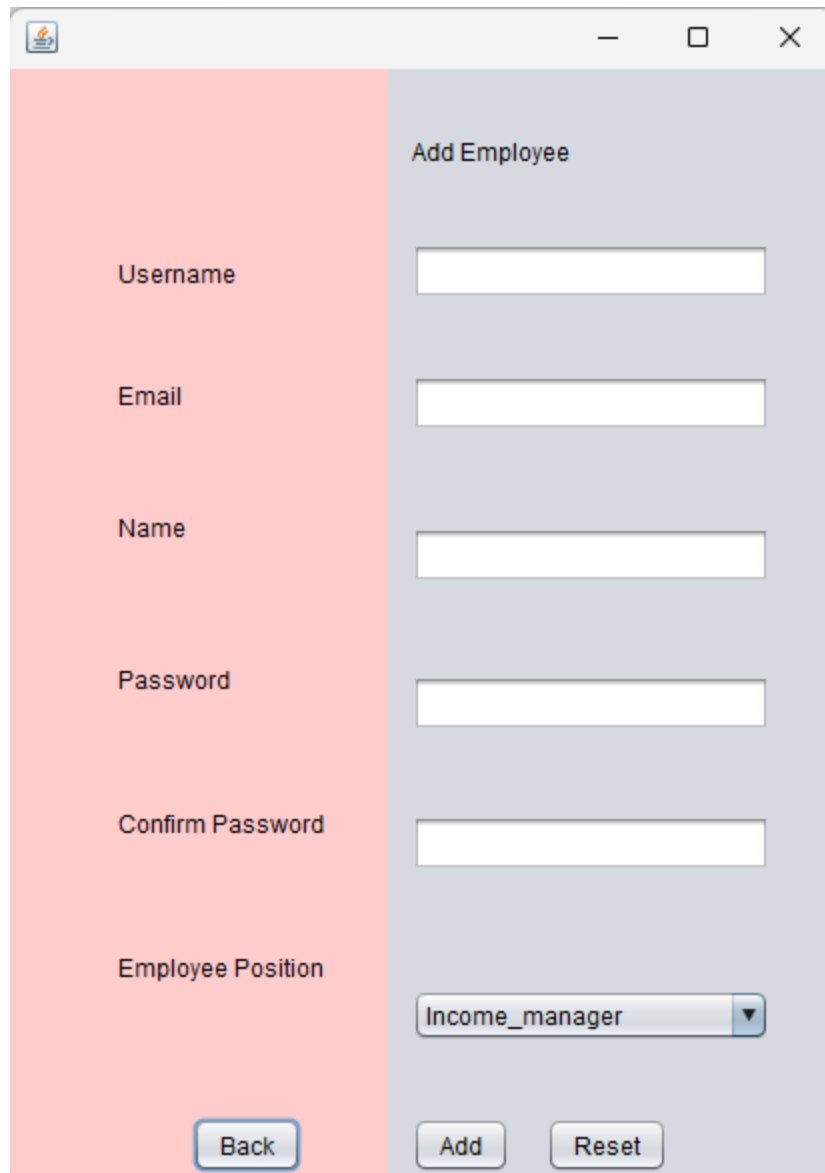
View Declined Bills

ID	Cheque no	Amount	Date Issued	Status
1	0103-45325A	7000.0	2023-03-03	Declined

Sign Out

Figure 6 View Declined bills





A screenshot of a web application window titled "Add Employee". The window has a light gray header bar with a small icon on the left and standard window controls (minimize, maximize, close) on the right. The main content area is split into two vertical sections. The left section has a light red background and contains labels for the form fields: "Username", "Email", "Name", "Password", "Confirm Password", and "Employee Position". The right section has a light gray background and contains the corresponding input fields: five text boxes and one dropdown menu. The dropdown menu for "Employee Position" currently shows "Income\_manager" with a downward arrow. At the bottom of the form, there are three buttons: "Back" (on the red section), "Add" (on the gray section), and "Reset" (on the gray section).

Add Employee	
Username	<input type="text"/>
Email	<input type="text"/>
Name	<input type="text"/>
Password	<input type="password"/>
Confirm Password	<input type="password"/>
Employee Position	<input type="text" value="Income_manager"/>
<input type="button" value="Back"/>	<input type="button" value="Add"/> <input type="button" value="Reset"/>

Figure 8 Add employee page

View Employee Info

Add Employee

View Cash Transaction

View Accepted Bills

View Declined Bills

View Income Transaction

Sign Out

ID

7

Account

0108-2587413

Status

Accepted

UPDATE

DELETE

ID	Username	Name	Email	Position
1	inmanager	Tasnima	1901043@iot.bd...	Income_manager
2	cmanager1	Sharika	1901013@iot.bd...	Cash_manager
3	ceo	admin	ceo@gmail.com	MD
4	wer	asd	wer@gmail.com	MD
5	ceo2	Rity	md1@yahoo.com	MD

ID	Account_no	Amount	Date	Status
1	0104-25312A	5000.0	2023-03-01	Pending
5	0103-45325A	7000.0	2023-03-03	Declined
6	0102-34217	7000.0	2023-03-07	Accepted
7	0108-2587413	7530.0	2023-03-17	Accepted
8	0108-2587753	4590.0	2023-03-17	Pending
9	0107-78541B	1560.0	2023-03-18	Pending

ID	Cheque No	Amount	Date Issued	Income Source
1	0104-258963	5000.0	2023-03-01	Selling goods
2	0103-2587410	7000.0	2022-02-28	Profit
3	0107-259763	9510.0	2023-03-18	Selling goods

Figure 9 View employee info, cash transactions and income transactions

## Static model - class diagrams

income_ledger	cash_ledger	accepted_bills	rejected_bills
ID int	ID int	ID int	ID int
Cheq_no varchar(255)	account_no varchar(45)	account_no varchar(45)	account_no varchar(45)
Amount float	amount varchar(45)	amount float	amount float
Date_added date	date_added date	date_added date	date_added date
income_source varchar(255)	cheq_status enum	cheq_status set(Accepted)	cheq_status set

registered_users
id int
username varchar(255)
mail varchar(255)
name varchar(255)
pass varchar(255)
emp_position enum

Dynamic model - sequence diagrams

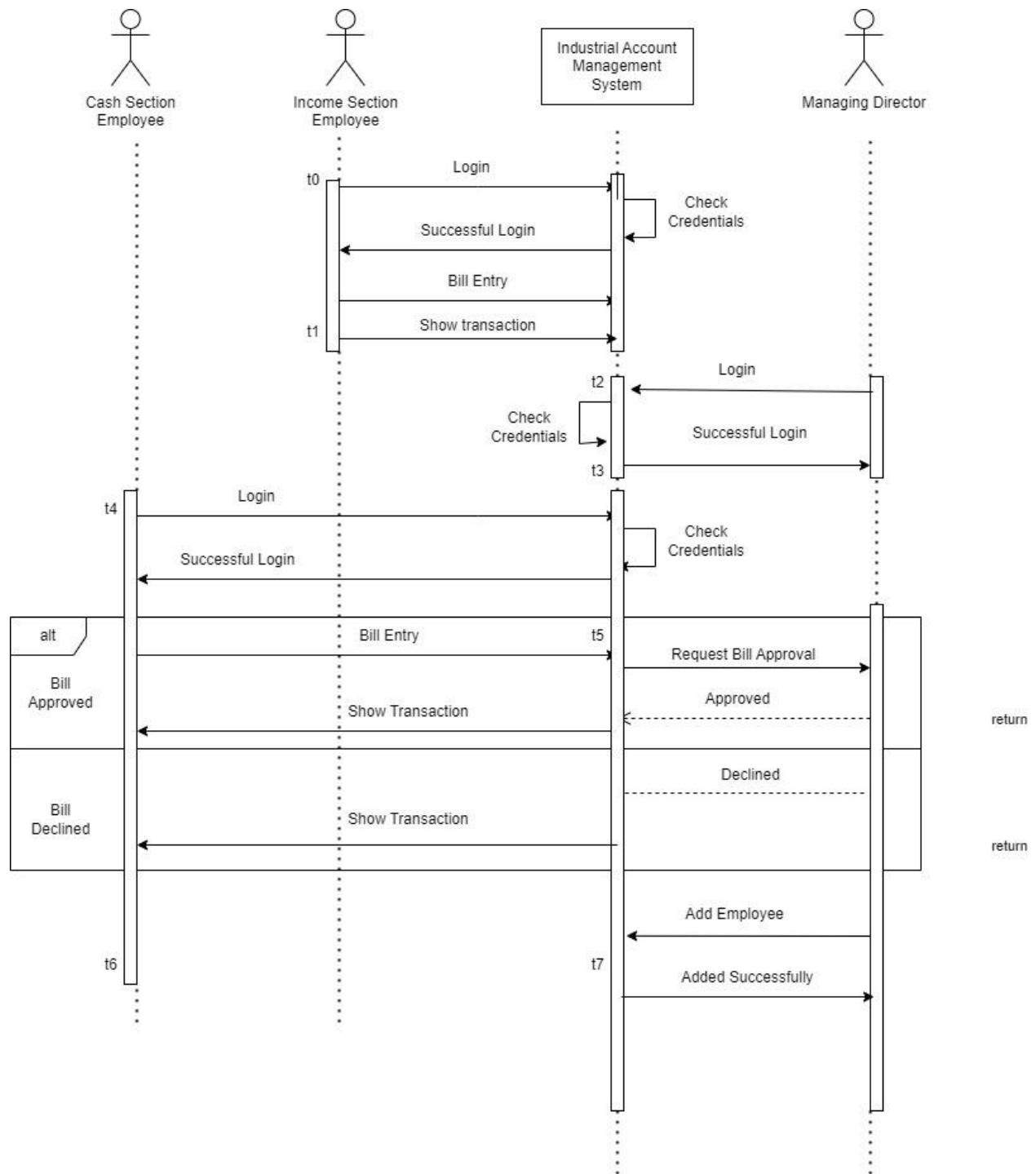


Figure: Sequence Diagram for Industrial Ledger

### *Rationale for your detailed design model*

The GUI design was chosen to ensure that the system is user-friendly and easy to navigate for all stakeholders. The use of UML class diagrams and sequence diagrams will help to ensure that the system is well-designed and that all stakeholders understand how the system works. The use of a consistent color scheme and layout will help to ensure that the system is visually appealing and easy to use.

### *Traceability from requirements to detailed design*

The detailed design model will be traceable to the requirements by ensuring that each requirement is met by the corresponding design element. For example, the GUI design will be traceable to the requirements by ensuring that all required functionality is included in the GUI, and that the GUI is designed in a user-friendly way. The class diagrams and sequence diagrams will be traceable to the requirements by ensuring that they accurately reflect the required functionality and flow of events within the system.



	A	B	C	D	E	F	G	H	I	J	K
1	Test Case ID		TC_01	Test Case Description	Test the login						
2	Created By		Group_10	Reviewed By	Farzana Akter, Habibur Rahman			Version	1.1		
3											
4	QA Tester's Log	Review comments from Farzana Akter and Habibur Rahman in version 1									
5											
6	Tester's Name		Taslima	Date Tested	March 11, 2023			Test Case (Pass/Fail/Not	Pass		
7											
8	S #	Prerequisites:				S #	Test Data				
9	1	The user is already registered.				1	Username = Inmanager, Password = 12345a, Position= Income_manager				
10	2	The user has unique username and				2	Username = cmanager1, Password = 54321b Position= Cash_manager				
11						3	Username = ceo, Password = 12345c Position= MD				
12						4	Username = cmanager, Password = 54321b Position= Cash_manager				
13						5	Username = cmanager1, Password = 654321c Position= Cash_manager				
14	Test Scenario	Verify the login field functionalisys									
15											
16	Step #	Step Details		Expected Results		Actual Results			Pass / Fail / Not executed / Suspended		
17											
18	1	log in with correct credentials		Show successful message and redirects to correct page		Show successful message and redirects to correct page			Pass		
19	2	login attempt with incorrect credentials		Shows error message		Shows error message and rejects login			Pass		

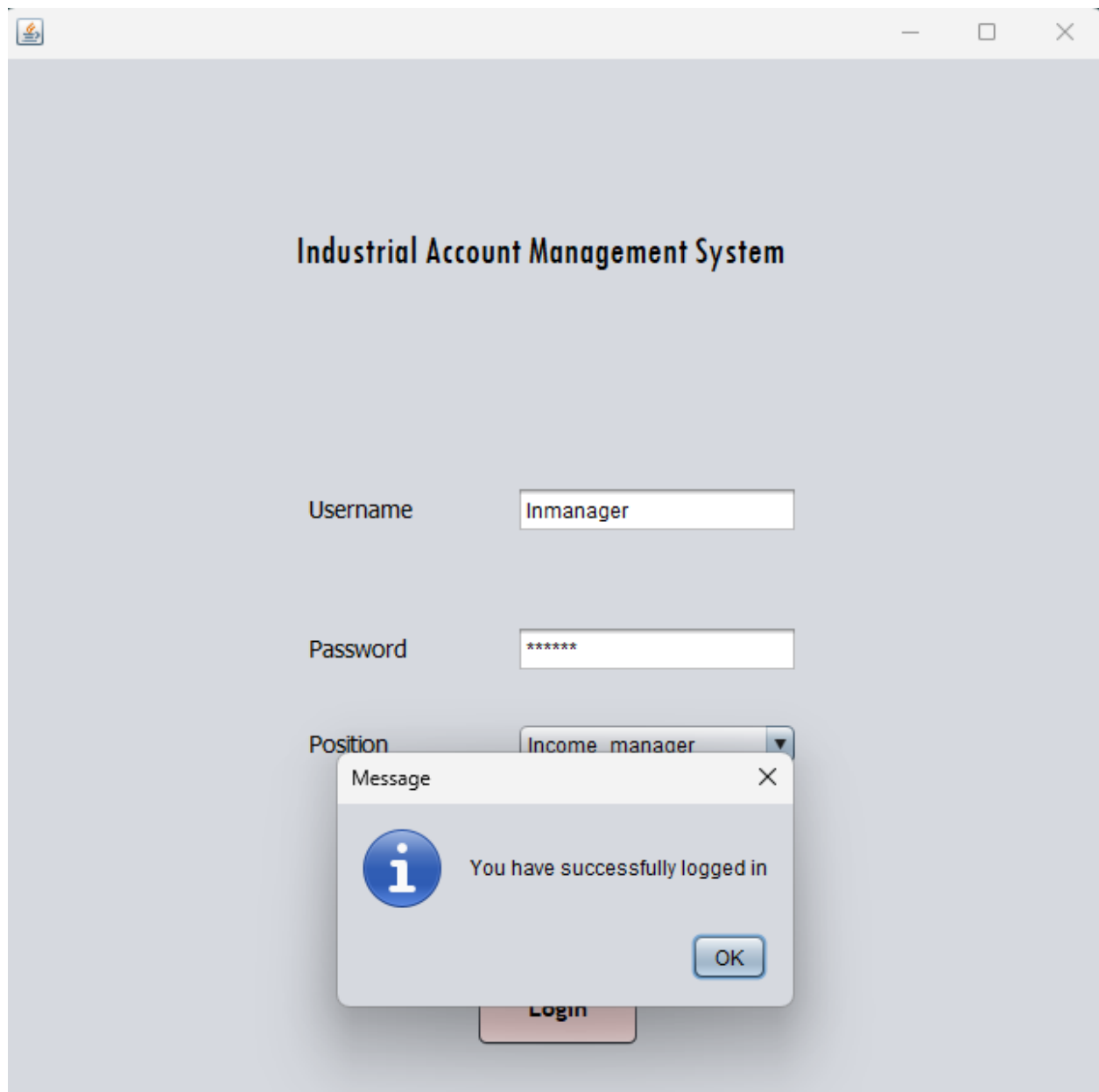


Figure 10 Successful login

Table 2 Test case for Income section page

	A	B	C	D	E	F	G	H	I	J	K
1	Test Case ID		TC_02	Test Case Description		Test the income section functionality					
2	Created By		Group_10	Reviewed By		Farzana Akter, Habibur Rahman		Version		1.1	
3											
4	QA Tester's Log		Review comments from Farzana Akter and Habibur Rahman in version 1								
5											
6	Tester's Name		Sharika	Date Tested		March 11, 2023		Test Case (Pass/Fail/Not Executed)		Pass	
7											
8	S #	Prerequisites:				S #	Test Data				
9	1	Income manager must be registered				1	Cheque No= 0107-25963, Amount= 9510.0 Date_Issued= 2023/03/18, Source= Selling				
10	2					2	Cheque No= 0107-25963, Amount= , Date_Issued= 2023/03/18, Source= Selling goods				
11	3					3	Cheque No= 0107-25963, Amount= 9510.0, Date_Issued= 2023/03/18				
12	4					4					
13						5					
14	Test Scenario		Verify the cash section functionalitys								
15											
16	Step #	Step Details		Expected Results		Actual Results			Pass / Fail / Not executed / Suspended		
17											
18	1	Entering new income record with all the required fields		Show successful message and redirects to correct page		Show successful message and adds details to database			Pass		
19	2	Adding incomplete or invalid data		Shows error message		Shows error message and rejects data			Pass		
20	3	View income details		Displays data		Displays data			Pass		

The screenshot displays the 'Income Section' of a web application. On the left, there is a pink sidebar. The main content area has a light blue background. At the top, the title 'Income Section' is centered. Below it, there is a 'View Transactions' button. A modal message box is open in the center, displaying an information icon and the text 'Entry successfully added' with an 'OK' button. In the background, a table shows transaction details:

ID	Cheque No	Amount	Date Issued	Income_source
0107-259763		00.0	2023-03-01	Selling goods
		00.0	2022-02-28	Profit

Below the table, there is an 'Add Records' button. At the bottom of the page, there is a 'Sign Out' button.

Figure 11 Income Entry added successfully

Table 3 Test case for Cash section page

	A	B	C	D	E	F	G	H	I	J	K
1	Test Case ID		TC_03	Test Case Description		Test the cash section functionality					
2	Created By		Group_10	Reviewed By		Farzana Akter, Habibur Rahman		Version		1.1	
3											
4	QA Tester's Log		Review comments from Farzana Akter and Habibur Rahman in version 1								
5											
6	Tester's Name		Taslima	Date Tested		March 11, 2023		Test Case (Pass/Fail/Not		Pass	
7											
8	S #	Prerequisites:				S #	Test Data				
9	1	Cash manager must be registered				1	Cheque No= 0107-78541B Amount= 1560.0, Date_Issued= 2023/03/18				
10	2					2	Cheque No= 0107-25963, Amount= , Date_Issued= 2023/03/18				
11	3					3	Cheque No= 0107-78541B Amount= 1560.0, Date_Issued= 2023-03-18				
12	4					4					
13						5					
14	Test Scenario	Verify the cash section functionalitys									
15											
16	Step #	Step Details		Expected Results		Actual Results			Pass / Fail / Not executed / Suspended		
17											
18	1	Entering new cash record with all the required fields		Show successful message and redirects to correct page		Show successful message and adds details to database			Pass		
19	2	Adding incomplete or invalid data		Shows error message		Shows error message and rejects data			Pass		
20	3	View bill details		Displays data		Displays data			Pass		
21	4	View accepted bill details		Only displays data which are accepted		Only displays data which are accepted			Pass		
22	5	View rejected bill details		Only displays data which are declined		Only displays data which are declined			Pass		

The screenshot displays the 'Cash Section' web application. On the left, there is a pink sidebar with labels: 'Cheque No', 'Amount', and 'Date Added'. The main area has a light blue background with the title 'Cash Section'. Below the title, there are three buttons: 'View All Records', 'View Accepted Bills', and 'View Declined Bills'. A form for adding a new cash entry is visible, with fields for 'Cheque No' (0107-78541B), 'Amount' (1560.0), and 'Date Added' (2023/03/18). An 'Add Bills' button is at the bottom left. A 'Sign Out' button is at the bottom center. A modal message box is open in the center, displaying an information icon and the text 'Entry successfully added' with an 'OK' button. In the background, a table shows a list of cash entries with columns: 'Cheque No', 'Amount', 'Date Issued', and 'Status'. The table contains five rows of data.

Cheque No	Amount	Date Issued	Status
312A	5000.0	2023-03-01	Pending
325A	7000.0	2023-03-03	Declined
217	7000.0	2023-03-07	Accepted
87413	7530.0	2023-03-17	Pending
87753	4590.0	2023-03-17	Pending

Figure 12 Cash Entry added successfully

Table 4 Test case for managing section page

	A	B	C	D	E	F	G	H	I	J	K
1	Test Case ID		TC_04	Test Case Description		Test the managing director section functionality					
2	Created By		Group_10	Reviewed By		Farzana Akter, Habibur Rahman			Version		1.1
3											
4	QA Tester's Log		Review comments from Farzana Akter and Habibur Rahman in version 1								
5											
6	Tester's Name		Tasnima	Date Tested		March 11, 2023			Test Case (Pass/Fail/Not		Pass
7											
8	S #	Prerequisites:				S #	Test Data				
9	1	Managing director must be registered				1	Status = Accpeted				
10	2	Data must be entered in Cash ledger				2	Status = Declined				
11	3					3					
12	4					4					
13						5					
14	Test Scenario Verify the managing section functionalits										
15											
16	Step #	Step Details		Expected Results		Actual Results			Pass / Fail / Not executed / Suspended		
17											
18	1	View employee info		Displays registered employee's data		Displays registered employee's data			Pass		
19	2	Adding incomplete or invalid data		Shows error message		Shows error message and rejects data			Pass		
20	3	View bill details		Displays data		Displays data			Pass		
21	4	View accepted bill details		Only displays data which are accepted		Only displays data which are accepted			Pass		
22	5	View rejected bill details		Only displays data which are declined		Only displays data which are declined			Pass		
23	6	Click on cash records table		Data are selected and displayed in update fields		Data are selected and displayed in update fields			Pass		
24	7	Click Update data(Accept/Decline)		Updates data successdully and shows message		Updates data successdully and shows message			Pass		
25	8	Click Delete button		Deletes data successdully and shows message		Deletes data successdully and shows message			Pass		
26	9	Accept bills		Add data in accepted bills table		Add data in accepted bills table			Pass		
27	10	Decline bills		Add data in rejected bills table		Add data in rejected bills table			Pass		

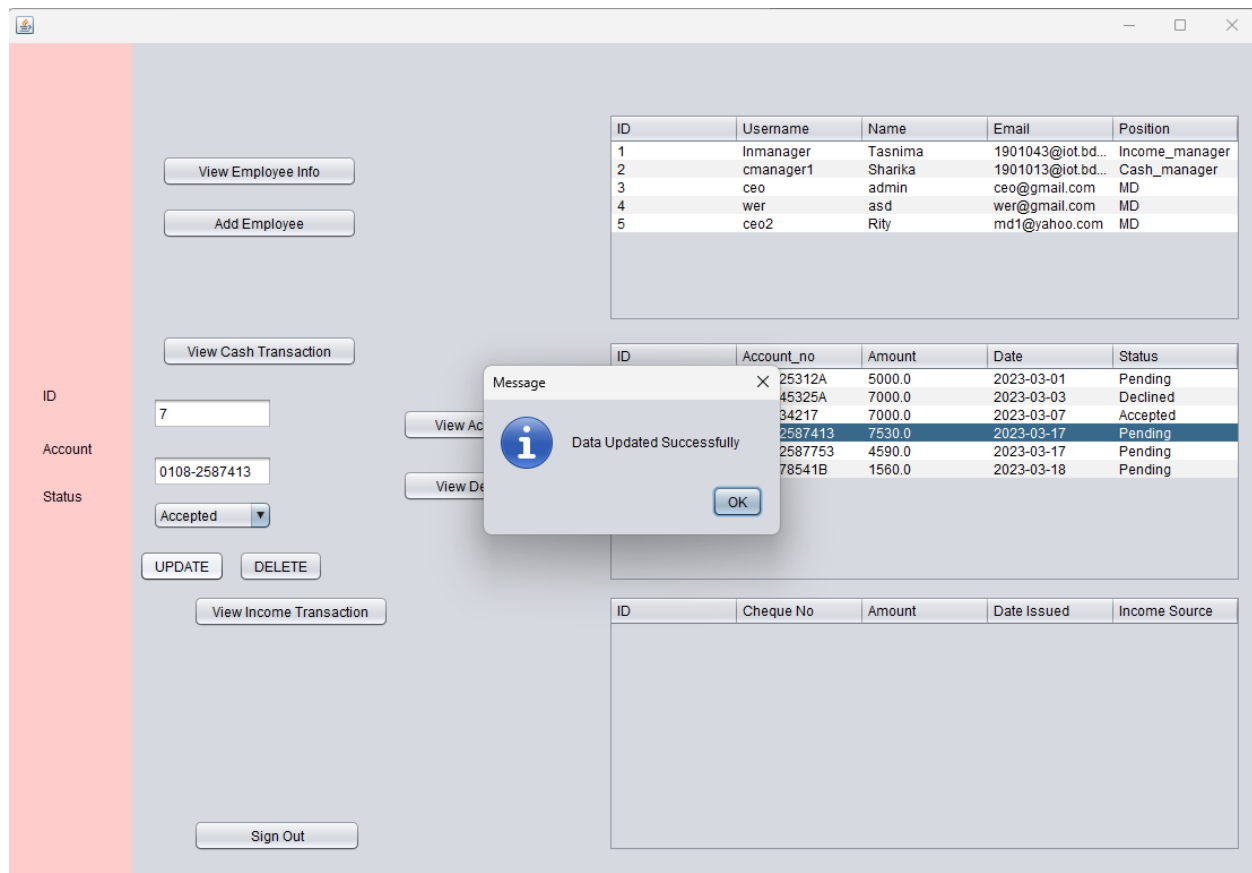


Figure 13 Data added successfully

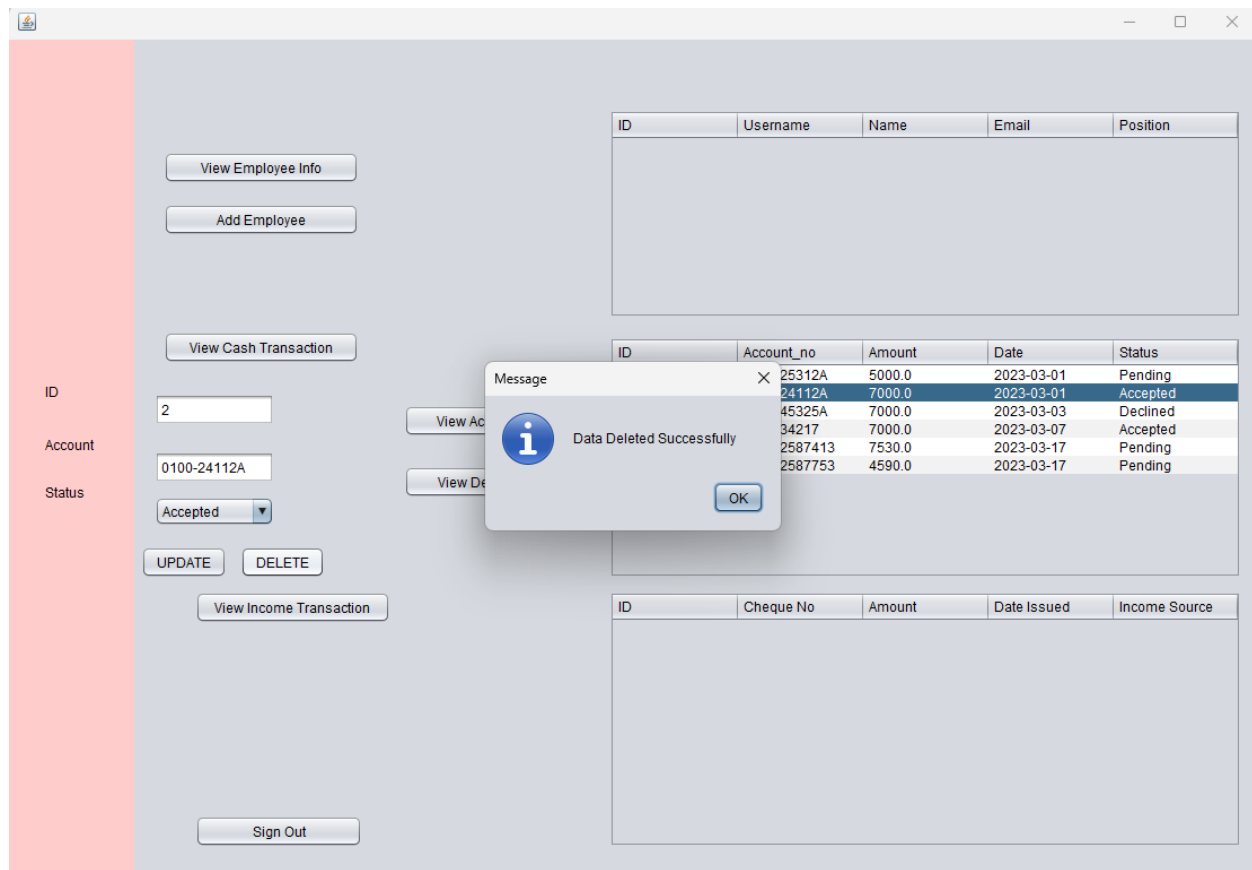


Figure 14 Data deleted successfully

Table 5 Test case for add employee page

	A	B	C	D	E	F	G	H	I	J	K
1	Test Case ID		TC_05	Test Case Description		Test the add employee section					
2	Created By		Group_10	Reviewed By		Farzana Akter, Habibur Rahman		Version		1.1	
3											
4	QA Tester's Log		Review comments from Farzana Akter and Habibur Rahman in version 1								
5											
6	Tester's Name		Sharika	Date Tested		March 11, 2023		Test Case (Pass/Fail/Not		Pass	
7											
8	S #	Prerequisites:				S #	Test Data				
9	1	Managing director is adding employee				1	Username=ceo2, Email=md1@yahoo.com, Name=Rity, Password=654321b, Confirm Password=654321b, Employee_position=MD				
10	2	Managing director must be registered				2	Username=ceo2, Email=md1@yaho, Name=Rity, Password=654321b, Confirm Password=654321b, Employee_position=MD				
11						3	Username=ceo2, Email=md1@yahoo.com, Name=Rity, Password=654321b, Confirm Password=6543, Employee_position=MD				
12						4	Username=ceo2, Email=md1@yahoo.com, Name= , Password=654321b, Confirm Password=654321b, Employee_position=MD				
13											
14	Test Scenario		Verify the add employee functionalitys								
15											
16	Step #	Step Details	Expected Results		Actual Results			Pass / Fail / Not executed / Suspended			
17											
18	1	Entry correct credentials and data	Show successful message and add data		Show successful message and add data			Pass			
19	2	Entry data with blank fields	Shows error message and rejects data		Shows error message and rejects data			Pass			
20	3	Add duplicate entry	Shows error message and rejects data		Shows error message and rejects data			Pass			
21	4	Click Refresh button	Clear all field data		Clear all field data			Pass			
22	5	Entry Invalid email	Shows error message		Shows error message			Pass			
23	6	Enter different password	Shows error message		Shows error message			Pass			



The screenshot shows a web application window titled 'Add Employee'. The form is divided into two main sections: a red-shaded left section and a grey-shaded right section. The left section contains labels for 'Username', 'Email', 'Name', 'Password', 'Confirm Password', and 'Employee Position'. The right section contains corresponding input fields. The 'Username' field contains 'ceo2', the 'Email' field contains 'md1@yahoo.com', the 'Confirm Password' field contains '\*\*\*\*\*', and the 'Employee Position' dropdown menu is set to 'MD'. A 'Message' dialog box is overlaid on the form, displaying an information icon and the text 'Entry successfully added' with an 'OK' button. At the bottom of the form are three buttons: 'Back', 'Add', and 'Reset'.

Figure 15 Employee successfully added

## Cost

### Cost estimation

As our system has less than 50 KLOC, it will be considered as an organic model. So, values of a, b, c and d will be as follows-

Project Level	a	b	c	d
Organic	2.4	1.05	2.5	0.38

Effort Applied =  $a \cdot (\text{KLOC})^b = (2.4) \cdot (2)^{1.05} = 4.96927$  person-months [Our project contains 2024 lines of code]

Development Time =  $c \cdot (\text{Effort Applied})^d = 2.5 \cdot (4.96927)^{0.38} = 4.5976$  months

Person Required = Effort Applied / Development Time =  $4.96927 / 4.5976 = 1.08$  or 1 person

Total Product Development Cost = Development Time / Salary of the staffs =  $4.5976 / 8000$

## Result

### *Result*

Thus our system software successfully completes its requirements, which are, adding records to database, updating them, deleting them and viewing them. Thus, it saves time and brings automation to this sector.

## Conclusion and Discussion

### *References*

- 1 Hsu, Donald. (2007). Case Studies in Financial Accounting Software.
- 2 Kowalewski, M., & Niedzielska, J. (2015). A Study of the Financial Accounting Software Used by Small Enterprises in Olsztyn. Olsztyn Economic Journal, 10, 233-245.
- 3 Mohd Nasir, Mohd Herry & Abdul Talib, Yurita Yakimin. (2018). MASMe: Developing Mobile Accounting Software for Micro Entrepreneurs. Journal of Information Communication and Ethics in Society. 16. 21-41.
- 4 Sharma, Amita & Sarangdevot, S.. (2010). Investigating the application of AOP methodology in development of Financial Accounting Software using Eclipse-AJDT Environment. AIP Conference Proceedings. 1324. 224-228. 10.1063/1.3526200.
- 5 Karma, I Gede Made. (2015). THE ACCOUNTING INFORMATION SYSTEMS OF WATER SPORTS COMPANY.