**Text

Description automatically generated with medium confidenceCollege Of Computer and Cyber SciencesSE324 Project *Spring 2021 - 2022***

**Software Construction**

**Bank Account Transactions**

**Project Report**

**Students**

**Zahraa Badawi | 3820003**

**Donna Al-Amri | 3910233**

**Fatma Alrehaily | 3910090**

**Supervised by**

**Dr. Khaled Khankan**

**Table of Contents**

Contents

[I. Introduction 3](#_Toc102554443)

[II. Project description 3](#_Toc102554444)

[III. Requirements specifications 4](#_Toc102554445)

[ Functional requirements 4](#_Toc102554446)

[ Non-functional requirements 4](#_Toc102554447)

[IV. Models 5](#_Toc102554448)

[ Use-case diagram 5](#_Toc102554449)

[ Overall architecture 5](#_Toc102554450)

[V. Implementation 6](#_Toc102554451)

[VI. Conclusion 6](#_Toc102554452)

[VII. References 7](#_Toc102554453)

# Introduction

Multi-threading systems is the ability of a program or an operating system process to handle numerous requests from the same user at the same time without the need for several copies of the programming to be executing in the computer. “Concurrency is the simultaneous execution of several instruction sequences”1. When numerous process threads are executing in parallel in the operating system, this occurs. The threads of a running process constantly communicate with one another via shared memory or message forwarding. Concurrency causes resource sharing, which leads to issues like as deadlocks and resource scarcity. In this report, we will show multi-threading programs in bank transactions by displaying the requirements and design besides implementation with source code. This report requires for the project of (software construction course).

# Project description

The project describes concurrency systems in bank transactions. It is a system for a user bank account who has different deposits/withdraws in same time with different ways. For each process, the balance will be increased/decreased by 25 SAR. For depositing, we create 5 threads to depositing money 1 by 1, while in withdrawing they are 2.

The idea of this project can be imagined by banks ATM, in depositing the machine checks all papers added before actual adding. Also, in withdrawing, money is checked one by one before actual withdrew. But we imagine it like adding 25 SAR and check it 1 by 1.

# Requirements specifications

## Functional requirements

* The system most create thread for each user, who will deposit amount into bank account (multiple people depositing into same account via bank transfer / ATM).
* The system most create thread for each user, who will withdraw amount into bank account (multiple people withdrawing into same account via bank transfer / ATM).
* The system most create number of users depositing/withdrawing in the account simultaneously.
* The system most add amount in bank account with each transaction of specific SAR until user completes the total amount deposit using a loop.
* The system most delete amount in bank account with each transaction of specific SAR until user completes the total amount withdraw using a loop.
* The system will keep each transaction related to money in a text file.

## Non-functional requirements

* if account has zero balance no withdraw is allowed.
* The system will not allow other thread to go beyond critical area until lock is released.
* The history file created will be shown only by account owner.
* The data in the file should be up to date.
* The multithreading system should be working with a good performance.
* Threads most be running with same priority using sleep time.

# Models

## Use-case diagram

Diagram

Description automatically generated

## Overall architecture

Diagram

Description automatically generated

# Implementation

The system is implemented using python language programming which has threading library help in our system.

# Conclusion

In conclusion, this project provides a real-life scenario of bank account transactions where concurrency and multithreading programming should be used to make the solution viable and responsive. First, the report discusses the software requirements specifications, including the functional and non-functional requirements. Secondly, Essential analysis and design models using UML notation. Finally, we implement the code by using Python language and importing the threading library.

# References

*Concurrency in operating system*. GeeksforGeeks. (2021, November 26). Retrieved May 1, 2022, from https://www.geeksforgeeks.org/concurrency-in-operating-system/

Contributor, T. T. (2005, September 21). *What is multithreading? - definition from whatis.com*. WhatIs.com. Retrieved May 1, 2022, from https://www.techtarget.com/whatis/definition/multithreading

*Diagrams.net - free flowchart maker and diagrams online*. Flowchart Maker & Online Diagram Software. (n.d.). Retrieved April 30, 2022, from https://app.diagrams.net/