BSc (Hons) in Computing

**Programming Data Structures and Algorithms**

Course Work

Group Assignment

Faculty of Engineering, Environment, and Computing

Coventry University

School of Computing

National Institute of Business Management

**School of Computing**

**National Institute of Business Management**

**Module ID**: NIB230CT

**Module Name**: Programming Data Structures and Algorithms

**Lecturer Name**: Mr. Sameera Jayasekara

**Title**: Develop Game Application

**Group Members**:

COBSCCOMP222P – 029

COBSCCOMP222P – 0

COBSCCOMP222P – 060 M.A.U Piyasena

**Table of contents**

Contents

[1. Source Code 1](#_Toc136291863)

[1.1 Source Code Path 1](#_Toc136291864)

[2. Database 2](#_Toc136291865)

[2.1 Database Details 2](#_Toc136291866)

[2.1.1 Fsfsgfwetwt 2](#_Toc136291867)

[3. Report 3](#_Toc136291868)

[3.1 Chapter 1 - Eight Queen Puzzle 3](#_Toc136291869)

[3.1.1 Program Logic used to Identify maximum number of solutions for Eight queen puzzle 3](#_Toc136291870)

[3.1.2 UI screenshot allowing game players to provide answers 3](#_Toc136291871)

[3.1.3 UI screenshots when game players to provide correct answer & incorrect answers 3](#_Toc136291872)

[3.1.4 Code Segment: When a game player correctly identifies an answer, save that person's name along with the correct response in the database 4](#_Toc136291873)

[3.1.5 Code Segment: If another game player provides the same right response, indicate that the solution has already been recognized 4](#_Toc136291874)

[3.1.6 Code Segment: When all the solutions have been identified by game players, the system should clear the flag that indicate solution has already been recognized 4](#_Toc136291875)

[3.1.7 UI screenshots when game players to provide correct answer & incorrect answers 4](#_Toc136291876)

[3.1.8 Indicate the Data Structures used with its purpose 5](#_Toc136291877)

[3.1.9 Specify using Code Segments or Screenshots the Validations and Exception Handling in this application 5](#_Toc136291878)

[3.1.10 Screenshot of the Normalized DB Table Structure used for this Game Option 5](#_Toc136291879)

[3.1.11 Code Segment: unit Testing 5](#_Toc136291880)

[3.2 Chapter 2 - Encode /Decode using Huffman 6](#_Toc136291881)

[3.2.1 Program Logic used to implement Encode /Decode using Huffman Coding Algorithm 6](#_Toc136291882)

[3.2.2 UI screenshot allowing game players to provide answers 6](#_Toc136291883)

[3.2.3 Code Segment: for Option 1 6](#_Toc136291884)

[3.2.4 Code Segment: for Option 2 6](#_Toc136291885)

[3.2.5 UI screenshots when game players to provide correct answer & incorrect answers 6](#_Toc136291886)

[3.2.6 Indicate the Data Structures used with its purpose 6](#_Toc136291887)

[3.2.7 Specify using Code Segments or Screenshots the Validations and Exception Handling in this application. 6](#_Toc136291888)

[3.2.8 Screenshot of the Normalized DB Table Structure used for this Game Option 6](#_Toc136291889)

[3.2.9 Code Segment: unit Testing 6](#_Toc136291890)

[3.3 Chapter 3 - Tic-Tac-Toe 7](#_Toc136291891)

[3.3.1 Program Logic used to implement Tic-Tac-Toe 7](#_Toc136291892)

[3.3.2 UI screenshot allowing game players to provide answers 7](#_Toc136291893)

[3.3.3 Code Segment: Determining the optimal Tic-Tac-Toe move for a computer player using the Minimax Algorithm in Game Theory 7](#_Toc136291894)

[3.3.4 Screenshots of the user interface when players win, lose, or draw a game 7](#_Toc136291895)

[3.3.5 Indicate the Data Structures used with its purpose 7](#_Toc136291896)

[3.3.6 Specify using Code Segments or Screenshots the Validations and Exception Handling in this application. 7](#_Toc136291897)

[3.3.7 Screenshot of the Normalized DB Table Structure used for this Game Option 7](#_Toc136291898)

[3.3.8 Code Segment: unit Testing 7](#_Toc136291899)

[3.4 Chapter 4 - Identify Shortest Path 8](#_Toc136291900)

[3.4.1 Program Logic used to implement Identify Shortest Path 8](#_Toc136291901)

[3.4.2 Code Segment used to set random distance 8](#_Toc136291902)

[3.4.3 UI screenshot allowing game players to provide answers 8](#_Toc136291903)

[3.4.4 Code Segment: find the shortest path and distance for other cities from the system's randomly selected city 8](#_Toc136291904)

[3.4.5 Code Segment used to save person's name along with the correct answer 8](#_Toc136291905)

[3.4.6 Code Segment used to save distance between cities when they correctly identify an answer 8](#_Toc136291906)

[3.4.7 UI screenshots when game players to provide correct answer & incorrect answers 8](#_Toc136291907)

[3.4.8 Indicate the Data Structures used with its purpose 8](#_Toc136291908)

[3.4.9 Specify using Code Segments or Screenshots the Validations and Exception Handling in this game option 8](#_Toc136291909)

[3.4.10 Screenshot of the Normalized DB Table Structure used for this Game Option 8](#_Toc136291910)

[3.4.11 Code Segment: unit Testing 8](#_Toc136291911)

[3.5 Chapter 5 - Identify minimum connecters 9](#_Toc136291912)

[3.5.1 Program Logic used to Identify minimum connecters 9](#_Toc136291913)

[3.5.2 Code Segment used to set random distance 9](#_Toc136291914)

[3.5.3 UI screenshots when game players to provide correct answer & incorrect answers 9](#_Toc136291915)

[3.5.4 Code Segment used to save person's name along with the correct answer 9](#_Toc136291916)

[3.5.5 Code Segment used to save distance between cities when they correctly identify an answer 9](#_Toc136291917)

[3.5.6 Indicate the Data Structures used with its purpose 9](#_Toc136291918)

[3.5.7 Specify using Code Segments or Screenshots the Validations and Exception Handling in this game option. 9](#_Toc136291919)

[3.5.8 Screenshot of the Normalized DB Table Structure used for this Game Option 9](#_Toc136291920)

[3.5.9 Code Segment: unit Testing 9](#_Toc136291921)

[4. Presentation 10](#_Toc136291922)

# Source Code

## Source Code Path

Path:

# Database

## Database Details

Database is designed using JSON file in java package for the game assignment.

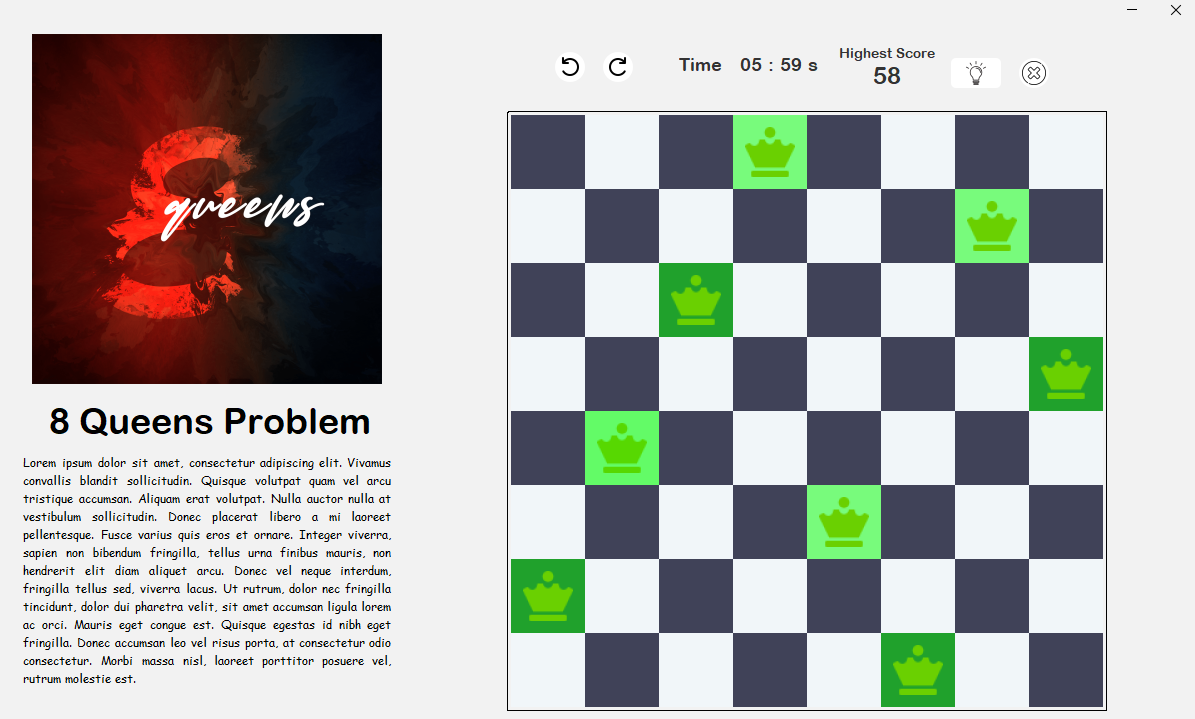
### Fsfsgfwetwt

# Report

## Chapter 1 - Eight Queen Puzzle

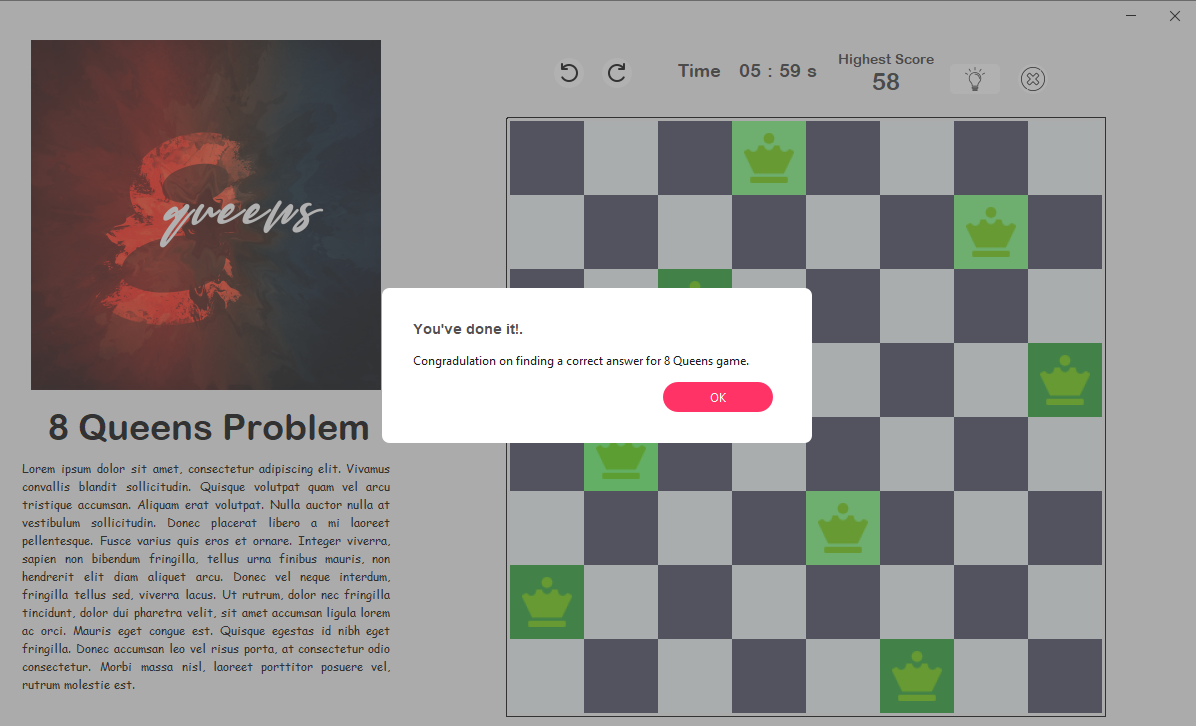
### Program Logic used to Identify maximum number of solutions for Eight queen puzzle

### UI screenshot allowing game players to provide answers



### UI screenshots when game players to provide correct answer & incorrect answers

* Correct Answer



* Incorrect Answer

### Code Segment: When a game player correctly identifies an answer, save that person's name along with the correct response in the database

### Code Segment: If another game player provides the same right response, indicate that the solution has already been recognized

### Code Segment: When all the solutions have been identified by game players, the system should clear the flag that indicate solution has already been recognized

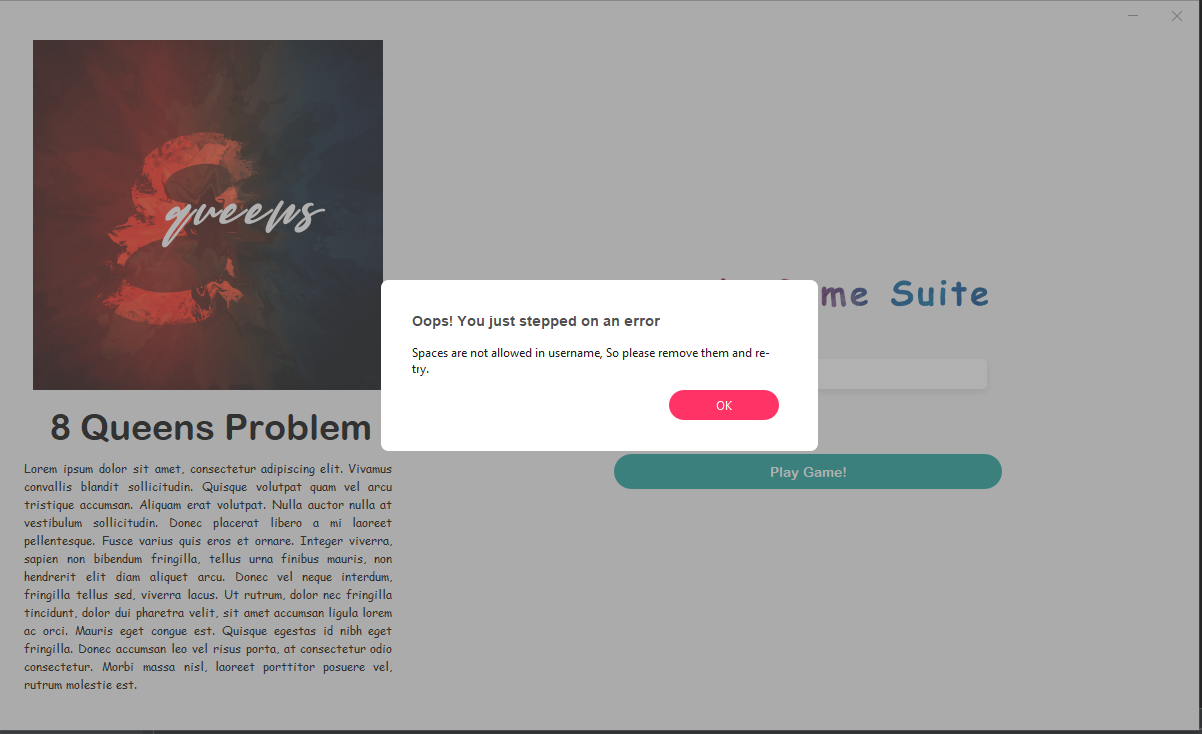
### UI screenshots when game players to provide correct answer & incorrect answers

### Indicate the Data Structures used with its purpose

* 1D Integer Array for representing the location of the Queen
* 2D Array for representing the Labels

### Specify using Code Segments or Screenshots the Validations and Exception Handling in this application

* Validations



* Exception Handling

### Screenshot of the Normalized DB Table Structure used for this Game Option

### Code Segment: unit Testing

## Chapter 2 - Encode /Decode using Huffman

1. Program Logic used to implement Encode /Decode using Huffman Coding Algorithm
2. UI screenshot allowing game players to provide answers
3. Code Segment: for Option 1
4. Code Segment: for Option 2
5. UI screenshots when game players to provide correct answer & incorrect answers
6. Indicate the Data Structures used with its purpose
7. Specify using Code Segments or Screenshots the Validations and Exception Handling in this application.
8. Screenshot of the Normalized DB Table Structure used for this Game Option
9. Code Segment: unit Testing

## Chapter 3 - Tic-Tac-Toe

1. Program Logic used to implement Tic-Tac-Toe
2. UI screenshot allowing game players to provide answers
3. Code Segment: Determining the optimal Tic-Tac-Toe move for a computer player using the Minimax Algorithm in Game Theory
4. Screenshots of the user interface when players win, lose, or draw a game
5. Indicate the Data Structures used with its purpose
6. Specify using Code Segments or Screenshots the Validations and Exception Handling in this application.
7. Screenshot of the Normalized DB Table Structure used for this Game Option
8. Code Segment: unit Testing

## Chapter 4 - Identify Shortest Path

1. Program Logic used to implement Identify Shortest Path
2. Code Segment used to set random distance
3. UI screenshot allowing game players to provide answers
4. Code Segment: find the shortest path and distance for other cities from the system's randomly selected city
5. Code Segment used to save person's name along with the correct answer
6. Code Segment used to save distance between cities when they correctly identify an answer
7. UI screenshots when game players to provide correct answer & incorrect answers
8. Indicate the Data Structures used with its purpose
9. Specify using Code Segments or Screenshots the Validations and Exception Handling in this game option
10. Screenshot of the Normalized DB Table Structure used for this Game Option
11. Code Segment: unit Testing

## Chapter 5 - Identify minimum connecters

1. Program Logic used to Identify minimum connecters
2. Code Segment used to set random distance
3. UI screenshots when game players to provide correct answer & incorrect answers
4. Code Segment used to save person's name along with the correct answer
5. Code Segment used to save distance between cities when they correctly identify an answer
6. Indicate the Data Structures used with its purpose
7. Specify using Code Segments or Screenshots the Validations and Exception Handling in this game option.
8. Screenshot of the Normalized DB Table Structure used for this Game Option
9. Code Segment: unit Testing

# Presentation