Mokasin

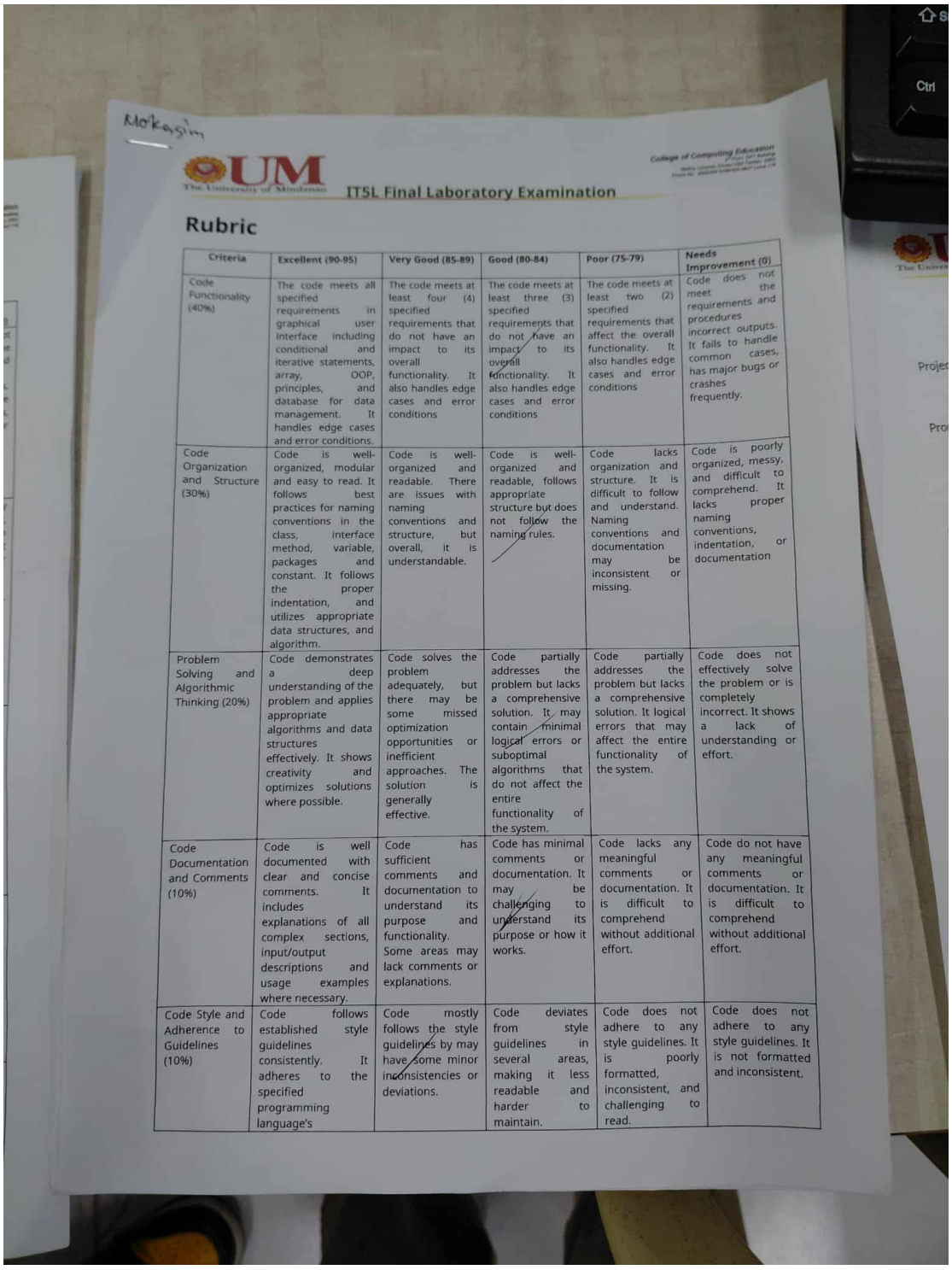
UM The University of Mindanao

College of Computing Education

IT5L Final Laboratory Examination

Rubric

Criteria	Excellent (90-95)	Very Good (85-89)	Good (80-84)	Poor (75-79)	Needs Improvement (0)
Code Functionality (40%)	The code meets all specified requirements in graphical user interface including conditional and iterative statements, array, OOP, principles, and database for data management. It handles edge cases and error conditions.	The code meets at least four (4) specified requirements that do not have an impact to its overall functionality. It also handles edge cases and error conditions	The code meets at least three (3) specified requirements that do not have an impact to its overall functionality. It also handles edge cases and error conditions	The code meets at least two (2) specified requirements that affect the overall functionality. It also handles edge cases and error conditions	Code does not meet the requirements and procedures incorrect outputs. It fails to handle common cases, has major bugs or crashes frequently.
Code Organization and Structure (30%)	Code is well-organized, modular and easy to read. It follows best practices for naming conventions in the class, interface method, variable, packages and constant. It follows the proper indentation, and utilizes appropriate data structures, and algorithm.	Code is well-organized and readable. There are issues with naming conventions and structure, but overall, it is understandable.	Code is well-organized and readable, follows appropriate structure but does not follow the naming rules.	Code lacks organization and structure. It is difficult to follow and understand. Naming conventions and documentation may be inconsistent or missing.	Code is poorly organized, messy, and difficult to comprehend. It lacks proper naming conventions, indentation, or documentation
Problem Solving and Algorithmic Thinking (20%)	Code demonstrates a deep understanding of the problem and applies appropriate algorithms and data structures effectively. It shows creativity and optimizes solutions where possible.	Code solves the problem adequately, but there may be some missed optimization opportunities or inefficient approaches. The solution is generally effective.	Code partially addresses the problem but lacks a comprehensive solution. It may contain minimal logical errors or suboptimal algorithms that do not affect the entire functionality of the system.	Code partially addresses the problem but lacks a comprehensive solution. It logical errors that may affect the entire functionality of the system.	Code does not effectively solve the problem or is completely incorrect. It shows a lack of understanding or effort.
Code Documentation and Comments (10%)	Code is well documented with clear and concise comments. It includes explanations of all complex sections, input/output descriptions and usage examples where necessary.	Code has sufficient comments and documentation to understand its purpose and functionality. Some areas may lack comments or explanations.	Code has minimal comments or documentation. It may be challenging to understand its purpose or how it works.	Code lacks any meaningful comments or documentation. It is difficult to comprehend without additional effort.	Code do not have any meaningful comments or documentation. It is difficult to comprehend without additional effort.
Code Style and Adherence to Guidelines (10%)	Code follows established style guidelines consistently. It adheres to the specified programming language's	Code mostly follows the style guidelines by may have some minor inconsistencies or deviations.	Code deviates from style guidelines in several areas, making it less readable and harder to maintain.	Code does not adhere to any style guidelines. It is poorly formatted, inconsistent, and challenging to read.	Code does not adhere to any style guidelines. It is not formatted and inconsistent.



Mokeyim

UM
The University of Mindanao

College of Computing Education

IT5L Final Laboratory Examination

Rubric

Criteria	Excellent (90-95)	Very Good (85-89)	Good (80-84)	Poor (75-79)	Needs Improvement (0)
Code Functionality (40%)	The code meets all specified requirements in graphical user interface including conditional and iterative statements, array, OOP, principles, and database for data management. It handles edge cases and error conditions.	The code meets at least four (4) specified requirements that do not have an impact to its overall functionality. It also handles edge cases and error conditions	The code meets at least three (3) specified requirements that do not have an impact to its overall functionality. It also handles edge cases and error conditions	The code meets at least two (2) specified requirements that affect the overall functionality. It also handles edge cases and error conditions	Code does not meet the requirements and procedures incorrect outputs. It fails to handle common cases, has major bugs or crashes frequently.
Code Organization and Structure (30%)	Code is well-organized, modular and easy to read. It follows best practices for naming conventions in the class, interface method, variable, packages and constant. It follows the proper indentation, and utilizes appropriate data structures, and algorithm.	Code is well-organized and readable. There are issues with naming conventions and structure, but overall, it is understandable.	Code is well-organized and readable, follows appropriate structure but does not follow the naming rules.	Code lacks organization and structure. It is difficult to follow and understand. Naming conventions and documentation may be inconsistent or missing.	Code is poorly organized, messy, and difficult to comprehend. It lacks proper naming conventions, indentation, or documentation
Problem Solving and Algorithmic Thinking (20%)	Code demonstrates a deep understanding of the problem and applies appropriate algorithms and data structures effectively. It shows creativity and optimizes solutions where possible.	Code solves the problem adequately, but there may be some missed optimization opportunities or inefficient approaches. The solution is generally effective.	Code partially addresses the problem but lacks a comprehensive solution. It may contain minimal logical errors or suboptimal algorithms that do not affect the entire functionality of the system.	Code partially addresses the problem but lacks a comprehensive solution. It logical errors that may affect the entire functionality of the system.	Code does not effectively solve the problem or is completely incorrect. It shows a lack of understanding or effort.
Code Documentation and Comments (10%)	Code is well documented with clear and concise comments. It includes explanations of all complex sections, input/output descriptions and usage examples where necessary.	Code has sufficient comments and documentation to understand its purpose and functionality. Some areas may lack comments or explanations.	Code has minimal comments or documentation. It may be challenging to understand its purpose or how it works.	Code lacks any meaningful comments or documentation. It is difficult to comprehend without additional effort.	Code do not have any meaningful comments or documentation. It is difficult to comprehend without additional effort.
Code Style and Adherence to Guidelines (10%)	Code follows established style guidelines consistently. It adheres to the specified programming language's	Code mostly follows the style guidelines by may have some minor inconsistencies or deviations.	Code deviates from style guidelines in several areas, making it less readable and harder to maintain.	Code does not adhere to any style guidelines. It is poorly formatted, inconsistent, and challenging to read.	Code does not adhere to any style guidelines. It is not formatted and inconsistent.

UM
The University of Mindanao

College of Computing Education

IT5L Final Laboratory ENRED

Enhancement Recommendation Discussion

Project: Car Rentals and Services

Proponents: Motasim, John Saher

Date: 10-09-2025

Document: TABLE OF CONTENTS MISALIGNMENT, USE CASES DIAGRAM CROPPED, PARAGRAPH IN-DENTATION LACKING

* SIMPLIFY USE CASES
* DATA DICTIONARY SHOULD BE SCREENSHOT from PHPMY-ADMIN
* FIX ERD

System:

GOOD SYSTEM UI AESTHETIC LOOKING,
dasboard add search input and filters
customer dashboard proper label spacing
remove log in notife above
admin dashbord bottom make it a sidebar or navigation
admin dashboard need to add Graphs

☐Approve Minor Revision Defense ☐Re-Checking Major Revision ☐Re-

NEIL [illegible] GALAS

ARTHUR JAMES DARUNDAY

10-09-2025
ARCO EXECUTIVE J.

7.0 TOOLS

The Car Rental and Services System can be developed using Python with frameworks like Flask to handle core features such as booking and billing. MySQL, XAMMP

Use Cases

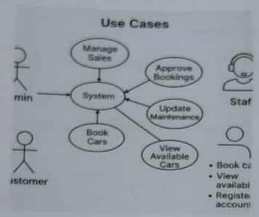


Figure 5: Use Cases

Entity Relationship Diagram

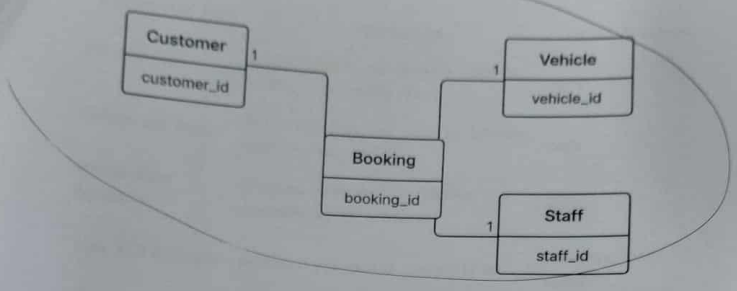


Figure 6:

Data Dictionary

Table 1: admin

Field Name	Data Type	Description
admin _id	INT (PK)	Unique ID for the admin
full_name	VARCHAR(100)	Admin' s full name
email	VARCHAR(100)	Admin' s email address
password	VARCHAR(100)	Admin' s login password
role	ENUM('admin')	User role type
phone	VARCHAR(20)	Admin contact number
address	VARCHAR(200)	Admin' s address