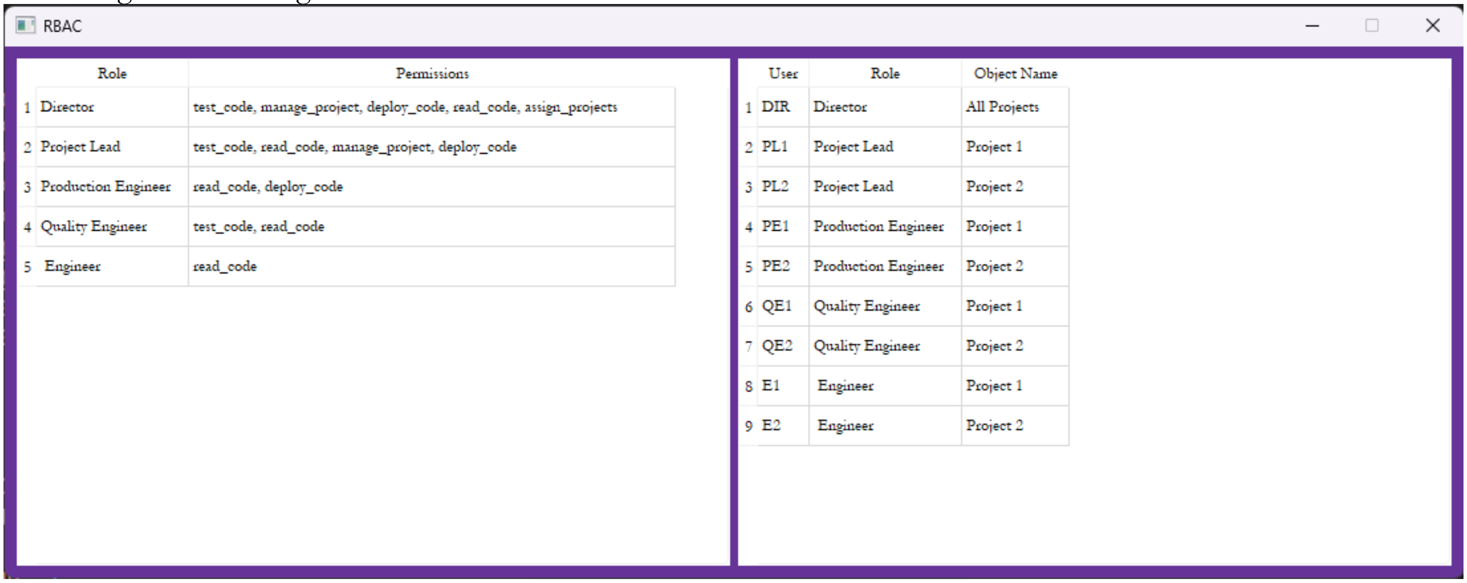
COSC 3364 – Principles of Cybersecurity

Lab 04

Role-Based Access Control

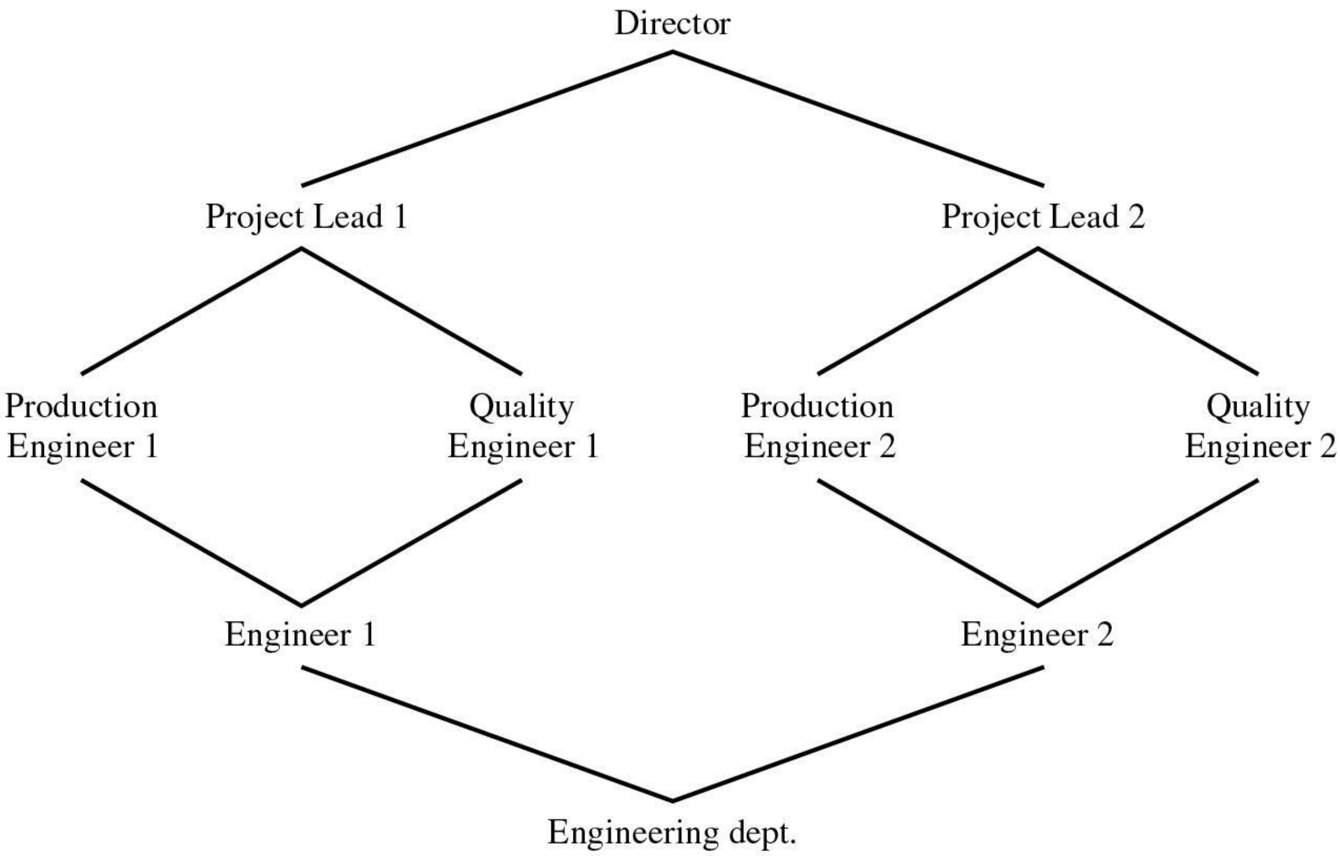
1. Develop a role-accessed control program based on the role hierarchy diagram of a software development team where each role will have permissions related to their position such as: read code, test code, deploy code, manage project, or assign projects. A line between two roles implies that the upper role includes all of the access rights of the lower role, as well as other access rights not available to the lower role. One role can inherit access rights from multiple subordinate roles. For example, the Project Lead role includes all of the access rights of the Production Engineer role and of the Quality Engineer role. More than one role can inherit from the same subordinate role. For example, both the Production Engineer role and the Quality Engineer role include all of the access rights of the Engineer Role.



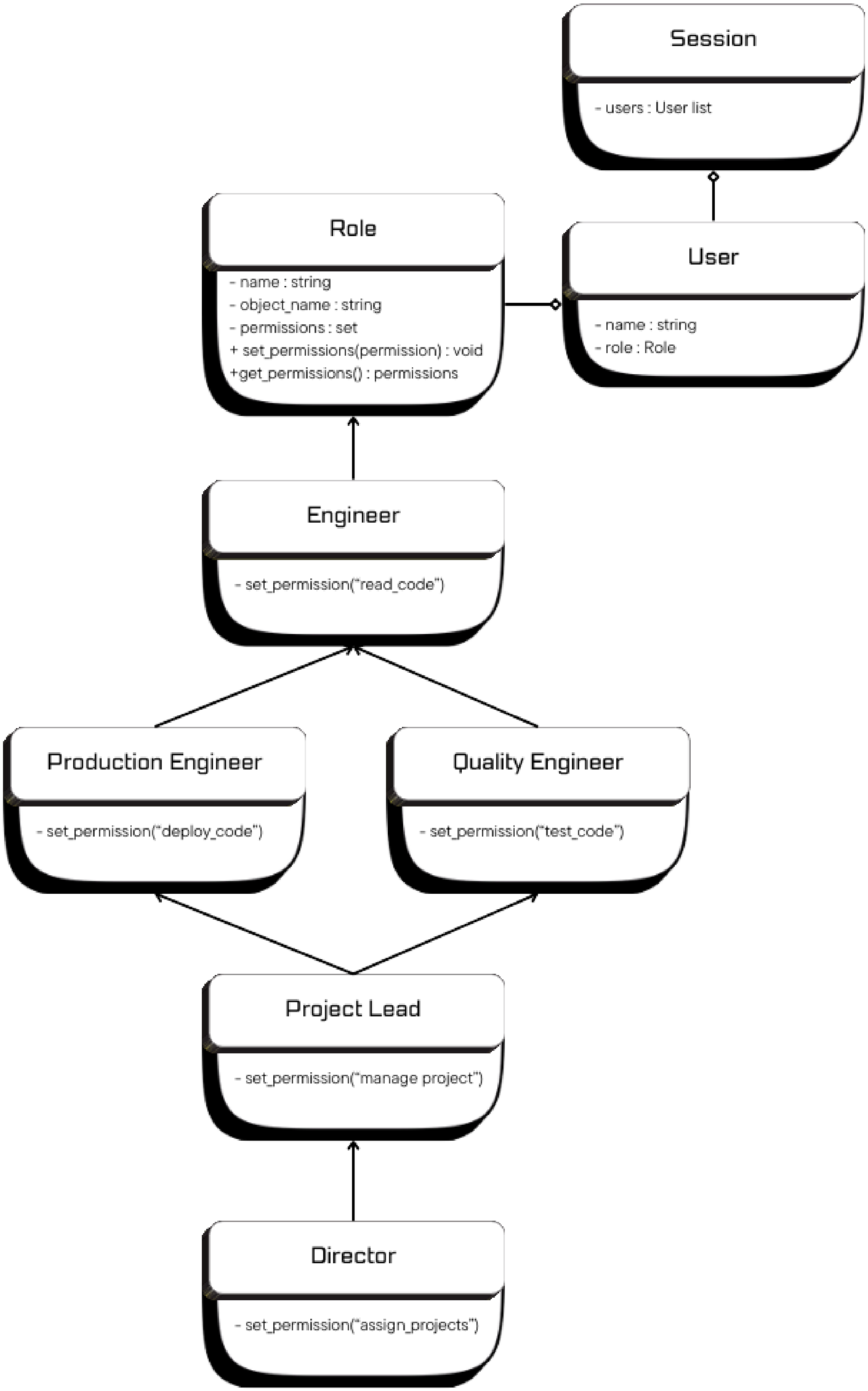
	Role	Permissions
1	Director	test_code, manage_project, deploy_code, read_code, assign_projects
2	Project Lead	test_code, read_code, manage_project, deploy_code
3	Production Engineer	read_code, deploy_code
4	Quality Engineer	test_code, read_code
5	Engineer	read_code

	User	Role	Object Name
1	DIR	Director	All Projects
2	PL1	Project Lead	Project 1
3	PL2	Project Lead	Project 2
4	PE1	Production Engineer	Project 1
5	PE2	Production Engineer	Project 2
6	QE1	Quality Engineer	Project 1
7	QE2	Quality Engineer	Project 2
8	E1	Engineer	Project 1
9	E2	Engineer	Project 2

Role Hierarchy



UML



Session
- users : User list
Role
- name : string
- object_name : string
- permissions : set
+ set_permissions(permission) : void
+get_permissions() : permissions
User
- name : string
- role : Role
Engineer
- set_permission("read_code")
Production Engineer
- set_permission("deploy_code")
Quality Engineer
- set_permission("test_code")
Project Lead
- set_permission("manage project")
Director
- set_permission("assign_projects")