

# CPSC 304 Project Cover Page

Milestone #: 2

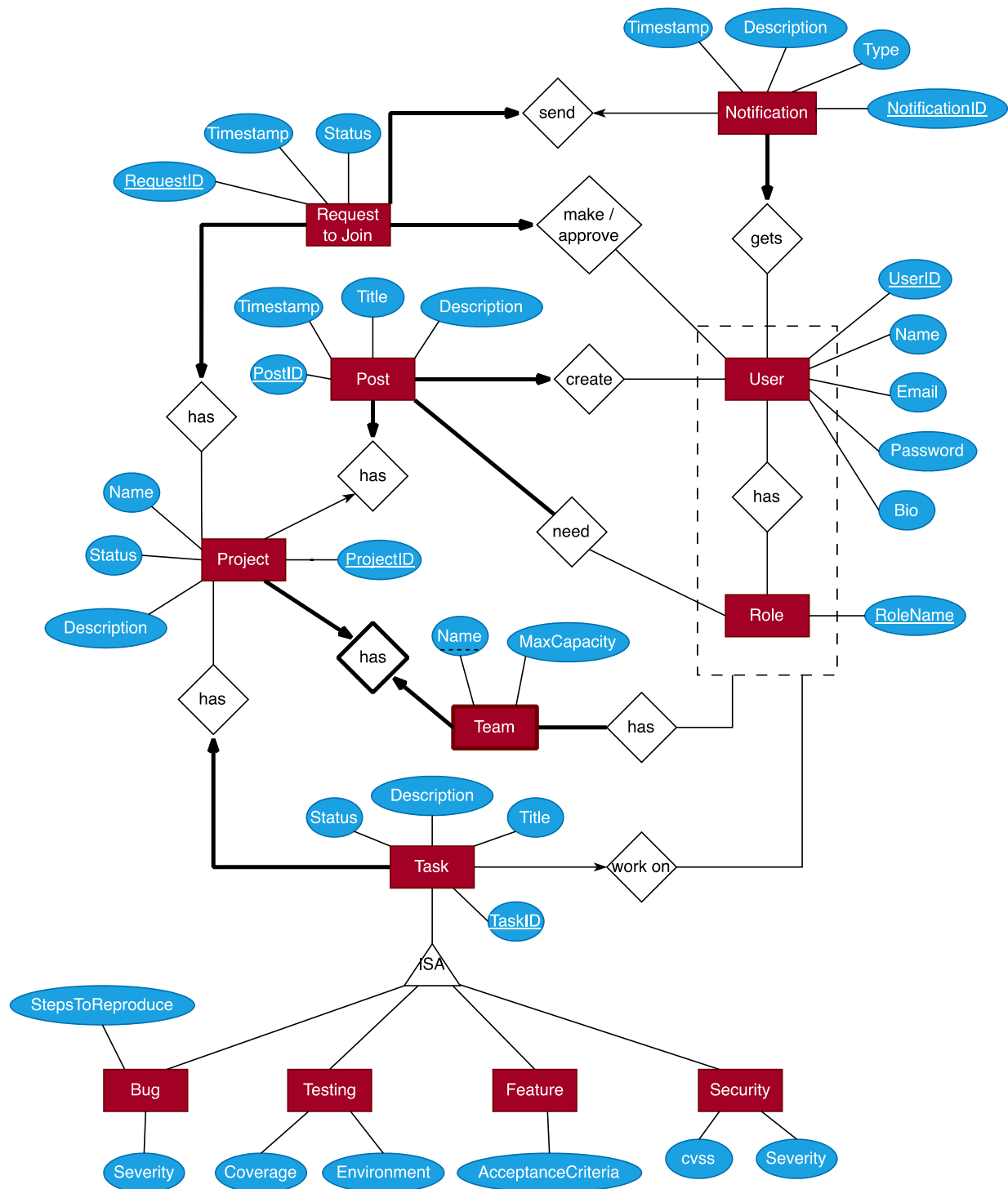
Date: 07/27/2023

Group Number: 22

Name	Student Number	CS Alias (Userid)	Preferred E-mail Address
Selin Uz	55505440	z6s4s	eceselinuz2@gmail.com
Guzide Irmak Bayir	11179330	w1q0i	irmakbayir@gmail.com
Huseyin Kutluay Cakadur	47261367	p6n2b	hk.cakadur@hotmail.com

By typing our names and student numbers in the above table, we certify that the work in the attached assignment was performed solely by those whose names and student IDs are included above. (In the case of Project Milestone 0, the main purpose of this page is for you to let us know your e-mail address, and then let us assign you to a TA for your project supervisor.)

In addition, we indicate that we are fully aware of the rules and consequences of plagiarism, as set forth by the Department of Computer Science and the University of British Columbia.



## Changes made to the ER:

1. The relationship between User and Request to Join was one-to-one. We realized that it should be many to one since a user can have a number of requests for different projects.
2. Notification must have a UserID (a notification must be sent to exactly one user, which adds a participation constraint from Notification to "gets" relationship).
3. Renamed all IDs to more meaningful names like ProjectID, PostID etc
4. Revised the attributes, added new ones and made all of them visible on the ER diagram.

## Pre-Decomposed Schemas:

The below schemas demonstrate the primary keys as underlined, and foreign keys as bold. The candidate key however is stated as a sub-bullet point as well as the not null comments.

- **Post**(PostID: char, Title: char, Description: char, Time: date, **UserID**: char, **ProjectID**: char)
  - ProjectID is a candidate key.
  - **UserID** and **ProjectID** can not be null
- **Project**(ProjectID: char, Name: char, Description: char, Status: char)
- **Team**(Name: char, MaxCapacity: int, **ProjectID**: char)
- **User**(UserID: char, Name: char, Email: char, Password: char, Bio: char)
  - Email is a candidate key.
- **Role**(RoleName: char)
- **Bug**(BugID: char, Title: char, Status: char, Description: char, Severity: char, StepsToReproduce: char, **UserID**: char, **RoleName**: char, **ProjectID**: char)
  - **ProjectID** can not be null
- **Testing**(TestingID: char, Title: char, Status: char, Description: char, Environment: char, Coverage: int, **UserID**: char, **RoleName**: char, **ProjectID**: char)
  - **ProjectID** can not be null
- **Feature**(FeatureID: char, Title: char, Status: char, Description: char, AcceptanceCriteria: char, **UserID**: char, **RoleName**: char, **ProjectID**: char)
  - **ProjectID** can not be null
- **Security**(SecurityID: char, Title: char, Status: char, Description: char, CVSS: decimal, Severity: char, **UserID**: char, **RoleName**: char, **ProjectID**: char)
  - **ProjectID** can not be null
- **RequestToJoin**(RequestID: char, Status: char, Time: date, **ProjectID**: char, **NotificationID**: char, **RequestingUserID**: char)

- NotificationID is a candidate key
  - **RequestingUserID**, **NotificationID** and **ProjectID** can not be null
- **Notification**(NotificationID: char, Type: char, Time: date, Description: char, ReviewingUserID: char)
  - **ReviewingUserID** can not be null
- **UserHasRole**(UserID: char, RoleName: char)
- **PostNeedsRole**(PostID: char, RoleName: char)
- **TeamHasPeople**(ProjectID: char, TeamName: char, RoleName: char, UserID: char)

## FDs for All Schemas:

Post:

- (PK) PostID → PostID, title, description, time, UserID, ProjectID
- (CK) ProjectID → PostID, title, description, time, UserID, ProjectID

Project:

- (PK) ProjectID → ProjectID, name, description, status

Team:

- (PK) name, ProjectID → name, max\_capacity, ProjectID

User:

- (PK) UserID → UserID, name, email, password, bio
- (CK) email → UserID, name, email, password, bio

Role:

- (PK) RoleName → RoleName

Bug:

- (PK) BugID → BugID, title, status, description, severity, steps\_to\_reproduce, UserID, RoleName, ProjectID

Testing:

- (PK) TestingID → TestingID, title, status, description, environment, type, coverage, UserID, RoleName, ProjectID
- **coverage → status**    **Violates 3NF**

Feature:

- (PK) FeatureID → FeatureID, title, status, description, acceptance\_criteria, UserID, RoleName, ProjectID

Security:

- (PK) SecurityID → SecurityID, title, status, description, cvss, severity, UserID, RoleName, ProjectID
- cvss → severity      Violates 3NF

RequestToJoin:

- (PK) RequestID → RequestID, status, time, ProjectID, NotificationID, RequestingUserID
- (CK) NotificationID → RequestID, status, time, ProjectID, NotificationID, RequestingUserID

Notification:

- (PK) NotificationID → NotificationID, type, time, description, ReviewingUserID
- type → description      Violates 3NF

UserHasRole:

- (PK) UserID, RoleName → UserID, RoleName

PostNeedsRole:

- (PK) PostID, RoleName → PostID, RoleName

TeamHasPeople:

- (PK) ProjectID, TeamName, RoleName, UserID → ProjectID, TeamName, RoleName, UserID

## Normalization to BCNF:

- **Notification**(NotificationID: char, Type: char, Time: date, Description: char, ReviewingUserID: char)
- Decomposing on type → description

Decomposed Relationships	Primary Key	Candidate Key	Foreign Key
<b>Notification</b> (NotificationID, Type, Time, ReviewingUserID)	<u>NotificationID</u>	-	<b>ReviewingUserID</b>
<b>NotificationTypeDescMap</b> (Type, Description)	<u>Type</u>	-	-

- **Testing**(TestingID: char, Title: char, Status: char, Description: char, Environment: char, Coverage: int, **UserID**: char, **RoleName**: char, **ProjectID**: char)
- Decomposing on **coverage** → **status**

Decomposed Relationships	Primary Key	Candidate Key	Foreign Key
<b>Testing</b> (TestingID, Title, Description, Environment, Type, Coverage, UserID, RoleName, ProjectID)	<u>TestingID</u>	-	<b>UserID, RoleName, ProjectID</b>
<b>TestingCoverageStatusMap</b> (Coverage, Status)	<u>Coverage</u>	-	-

- **Security**(SecurityID: char, Title: char, Status: char, Description: char, CVSS: decimal, Severity: char, **UserID**: char, **RoleName**: char, **ProjectID**: char)
- Decomposing on **cvss** → **severity**

Decomposed Relationships	Primary Key	Candidate Key	Foreign Key
<b>Security</b> (SecurityID, Title, Status, Description, CVSS, UserID, RoleName, ProjectID)	<u>SecurityID</u>	-	<b>UserID, RoleName, ProjectID</b>
<b>SecurityCvssSeverityMap</b> (CVSS, Severity)	<u>CVSS</u>	-	-

## Post-Normalization Schemas:

- **Post**(PostID: char, Title: char, Description: char, Time: date, **UserID**: char, **ProjectID**: char)
  - ProjectID is a candidate key.
  - **UserID** and **ProjectID** can not be null
- **Project**(ProjectID: char, Name: char, Description: char, Status: char)
- **Team**(Name: char, MaxCapacity: int, **ProjectID**: char)
- **User**(UserID: char, Name: char, Email: char, Password: char, Bio: char)
  - Email is a candidate key.

- **Role**(RoleName: char)
- **Bug**(BugID: char, Title: char, Status: char, Description: char, Severity: char, StepsToReproduce: char, **UserID**: char, **RoleName**: char, **ProjectID**: char)
  - **ProjectID** can not be null
- **Testing**(TestingID: char, Title: char, Description: char, Environment: char, Coverage: int, **UserID**: char, **RoleName**: char, **ProjectID**: char)
  - **ProjectID** can not be null
- **TestingCoverageStatusMap**(Coverage: int, Status: char)
- **Feature**(FeatureID: char, Title: char, Status: char, Description: char, AcceptanceCriteria: char, **UserID**: char, **RoleName**: char, **ProjectID**: char)
  - **ProjectID** can not be null
- **Security**(SecurityID: char, Title: char, Status: char, Description: char, CVSS: decimal, **UserID**: char, **RoleName**: char, **ProjectID**: char)
  - **ProjectID** can not be null
- **SecurityCvssSeverityMap**(CVSS: decimal, Severity: char)
- **RequestToJoin**(RequestID: char, Status: char, Time: date, **ProjectID**: char, **NotificationID**: char, **RequestingUserID**: char)
  - NotificationID is a candidate key
  - **RequestingUserID**, **NotificationID** and **ProjectID** can not be null
- **Notification**(NotificationID: char, Type: char, Time: date, **ReviewingUserID**: char)
  - **ReviewingUserID** can not be null
- **NotificationTypeDescMap**(Type: char, Description: char)
- **UserHasRole**(UserID: char, RoleName: char)
- **PostNeedsRole**(PostID: char, RoleName: char)
- **TeamHasPeople**(ProjectID: char, TeamName: char, RoleName: char, UserID: char)

## SQL DDLs

```
CREATE TABLE Post(
  PostID: char(11),
  Title: char(50),
  Description: char(400),
  Time: date,
  UserID: char(11) NOT NULL,
  ProjectID: char(11) NOT NULL UNIQUE,
  PRIMARY KEY (PostID),
  FOREIGN KEY (UserID) REFERENCES User,
  FOREIGN KEY (ProjectID) REFERENCES Project,
```

)

CREATE TABLE Project(

ProjectID: char(11),

Name: char(50),

Description: char(400),

Status: char(11),

PRIMARY KEY (ProjectID)

)

CREATE TABLE Team(

Name: char(50),

MaxCapacity: int,

ProjectID: char(11),

PRIMARY KEY (Name, ProjectID),

FOREIGN KEY (ProjectID) REFERENCES Project

ON DELETE CASCADE

)

CREATE TABLE User(

UserID: char(11),

Name: char(50),

Email: char(50) UNIQUE,

Password: char(50),

Bio: char(400),

PRIMARY KEY (UserID)

)

CREATE TABLE Role(

RoleName: char(25),

PRIMARY KEY (RoleName)

)

CREATE TABLE Bug(

BugID: char(11),

Title: char(50),

Status: char(11),



```
Description: char(400),
Severity: char(11),
StepsToReproduce: char(400),
UserID: char(11),
RoleName: char(25),
ProjectID: char(11) NOT NULL,
PRIMARY KEY (BugID),
FOREIGN KEY (UserID) REFERENCES User,
FOREIGN KEY (RoleName) REFERENCES Role,
FOREIGN KEY (ProjectID) REFERENCES Project,
)
```

```
CREATE TABLE Testing(
  TestingID: char(11),
  Title: char(11),
  Description: char(400),
  Environment: char(11),
  Coverage: int,
  UserID: char(11),
  RoleName: char(25),
  ProjectID: char(11) NOT NULL,
  PRIMARY KEY (TestingID),
  FOREIGN KEY (UserID) REFERENCES User,
  FOREIGN KEY (RoleName) REFERENCES Role,
  FOREIGN KEY (ProjectID) REFERENCES Project,
)
```

```
CREATE TABLE TestingCoverageStatusMap(
  Coverage: int,
  Status: char(11),
  PRIMARY KEY (Coverage)
)
```

```
CREATE TABLE Feature(
  FeatureID: char(11),
  Title: char(50),
  Status: char(11),
```

```
Description: char(400),
AcceptanceCriteria: char(400),
UserID: char(11),
RoleName: char(25),
ProjectID: char(11) NOT NULL,
PRIMARY KEY (FeatureID),
FOREIGN KEY (UserID) REFERENCES User,
FOREIGN KEY (RoleName) REFERENCES Role,
FOREIGN KEY (ProjectID) REFERENCES Project,
)
```

```
CREATE TABLE Security(
    SecurityID: char(11),
    Title: char(50),
    Status: char(11),
    Description: char(400),
    CVSS: decimal,
    UserID: char(11),
    RoleName: char(25),
    ProjectID: char(11) NOT NULL,
    PRIMARY KEY (SecurityID),
    FOREIGN KEY (UserID) REFERENCES User,
    FOREIGN KEY (RoleName) REFERENCES Role,
    FOREIGN KEY (ProjectID) REFERENCES Project,
)
```

```
CREATE TABLE SecurityCVSSSeverityMap(
    CVSS: decimal,
    Severity: char(11),
    PRIMARY KEY (CVSS)
)
```

```
CREATE TABLE RequestToJoin(
    RequestID: char(11),
    Status: char(11),
    Time: date,
    ProjectID: char(11) NOT NULL,
```

```
NotificationID: char(11) NOT NULL UNIQUE,  
RequestingUserID: char(11) NOT NULL,  
PRIMARY KEY (RequestID),  
FOREIGN KEY (ProjectID) REFERENCES Project,  
FOREIGN KEY (NotificationID) REFERENCES Notification,  
FOREIGN KEY (RequestingUserID) REFERENCES User,  
)
```

```
CREATE TABLE Notification(  
    NotificationID: char(11),  
    Type: char(11),  
    Time: date,  
    ReviewingUserID: char(11) NOT NULL,  
    PRIMARY KEY (NotificationID),  
    FOREIGN KEY (ReviewingUserID) REFERENCES User  
)
```

```
CREATE TABLE NotificationTypeDescMap(  
    Type: char(11),  
    Description: char(400),  
    PRIMARY KEY (Type)  
)
```

```
CREATE TABLE UserHasRole(  
    UserID: char(11),  
    RoleName: char(25),  
    PRIMARY KEY (UserID, RoleName),  
    FOREIGN KEY (UserID) REFERENCES User,  
    FOREIGN KEY (RoleName) REFERENCES Role  
)
```

```
CREATE TABLE PostNeedsRole(  
    PostID: char(11),  
    RoleName: char(25),  
    PRIMARY KEY (PostID, RoleName),  
    FOREIGN KEY (PostID) REFERENCES Post,  
    FOREIGN KEY (RoleName) REFERENCES Role
```

```

)

CREATE TABLE TeamHasPeople(
  ProjectID: char(11),
  TeamName: char(50),
  RoleName: char(25),
  UserID: char(11),
  PRIMARY KEY (ProjectID, TeamName, RoleName, UserID),
  FOREIGN KEY (ProjectID) REFERENCES Project,
  FOREIGN KEY (TeamName) REFERENCES Team,
  FOREIGN KEY (RoleName) REFERENCES Role,
  FOREIGN KEY (UserID) REFERENCES User,
)

```

## 5 Tuples Per Table

Post					
<u>PostID</u>	Title	Description	Time	<u>UserID</u>	<u>ProjectID</u>
"000-000-001"	"Looking for partne	"Hi all, I want to find partn	2023-01-01	"000-000-001"	"000-000-001"
"000-000-002"	"In need of develop	"Hello everyone, is there a	2023-01-01	"000-000-001"	"000-000-002"
"000-000-003"	"Help"	"URGENT! I NEED A UI D	2023-01-01	"000-000-002"	"000-000-005"
"000-000-004"	"Developers for Ins	"If you have worked with J	2023-01-01	"000-000-001"	"000-000-003"
"000-000-005"	"UI Designer neede	"Want to work on SpaceX	2023-01-01	"000-000-003"	"000-000-004"

Project			
<u>ProjectID</u>	Name	Description	Status
"000-000-001"	"This is a project"	"Random project"	"in-progress"
"000-000-002"	"Twitter "	"This is project for creating Twitter"	"completed"
"000-000-003"	"Instagram"	"Internal server building"	"in-progress"
"000-000-004"	"SpaceX"	"Rocket design and coding"	"in-progress"
"000-000-005"	"BC Hydro"	"Infrastructure of BC hydro"	"in-progress"

Role
<u>RoleName</u>
"frontend-developer"
"backend-developer"
"designer"
"project-manager"
"qa"

Team		
<u>Name</u>	<u>MaxCapacity</u>	<u>ProjectID</u>
"eagles"	3	"000-000-003"
"lions"	4	"000-000-001"
"raptors"	5	"000-000-002"
"alpha"	6	"000-000-004"
"bravo"	7	"000-000-005"

User				
UserID	Name	Email	Password	Bio
"000-000-001"	"Kutluay Cakadur"	"kutluaycakadur@gmail.com"	"1234d567890"	"I code babygirl"
"000-000-002"	"Selin Uz"	"selinuz@gmail.com"	"S34D765VHAJ"	"Hey, I focus on backend and cook!"
"000-000-003"	"Irmak Bayir"	"anan@gmail.com"	"mlnkbhj4365"	"Wazzup"
"000-000-004"	"Gittu George"	"gittugeorge@gmail.com"	"ladfn92o2bmfa"	"Professor at UBC"
"000-000-005"	"Jason Hall"	"jasonhall@gmail.com"	"325ljbkw425k"	"Hi, I'm a cool TA."

Bug								
BugID	Title	Status	Severity	Description	StepsToReproduce	UserID	RoleName	ProjectID
"000-000-001"	"unending promise"	"in progress"	"high"	"a"	"a"	"000-000-001"	"backend-develc"	"000-000-001"
"000-000-002"	"cannot see button"	"open"	"medium"	"b"	"b"	"000-000-003"	"frontend-develo"	"000-000-001"
"000-000-003"	"filter not working"	"closed"	"low"	"c"	"c"	"000-000-001"	"backend-develc"	"000-000-001"
"000-000-004"	"bug in main branch"	"in progress"	"high"	"d"	"d"	"000-000-001"	"backend-develc"	"000-000-001"
"000-000-005"	"for loop doesn't index through last i"	"in progress"	"high"	"e"	"e"	"000-000-001"	"backend-develc"	"000-000-001"

Testing								
TestingID	Title	Description	Environment	Type	Coverage	UserID	RoleName	ProjectID
"000-000-006"	"Testing"	"Testing"	"staging"	"unit"	50	"000-000-002"	"qa"	"000-000-005"
"000-000-007"	"Testing"	"Testing"	"prod"	"eco-system"	25	"000-000-002"	"qa"	"000-000-005"
"000-000-008"	"Testing"	"Testing"	"dev"	"unit"	75	"000-000-002"	"qa"	"000-000-005"
"000-000-009"	"Testing"	"Testing"	"prod"	"regression"	100	"000-000-002"	"qa"	"000-000-005"
"000-000-010"	"Testing"	"Testing"	"dev"	"unit"	25	"000-000-002"	"qa"	"000-000-005"

Security								
SecurityID	Title	Status	Description	CVSS	UserID	RoleName	ProjectID	
"000-000-016"	"security vulnerability with depender"	"in progress"	"Security"	1	"000-000-005"	"backend-develc"	"000-000-005"	
"000-000-017"	"security scan warning"	"open"	"Security"	2	"000-000-005"	"backend-develc"	"000-000-005"	
"000-000-018"	"security vulnerability"	"closed"	"Security"	3	"000-000-005"	"backend-develc"	"000-000-005"	
"000-000-019"	"dependency has vulnerability"	"in progress"	"Security"	4	"000-000-005"	"backend-develc"	"000-000-005"	
"000-000-020"	"need to update dependency"	"in progress"	"Security"	5	"000-000-005"	"backend-develc"	"000-000-005"	

Feature								
FeatureID	Title	Status	Description	AcceptanceCriteria	UserID	RoleName	ProjectID	
"000-000-011"	"feature"	"open"	"feature"	"some criteria"	"000-000-004"	"backend-developer"	"000-000-004"	
"000-000-012"	"feature"	"open"	"feature"	"some criteria"	"000-000-004"	"backend-developer"	"000-000-004"	
"000-000-013"	"feature"	"in-progress"	"feature"	"some criteria"	"000-000-004"	"backend-developer"	"000-000-004"	
"000-000-014"	"feature"	"in-progress"	"feature"	"some criteria"	"000-000-004"	"backend-developer"	"000-000-004"	
"000-000-015"	"feature"	"closed"	"feature"	"some criteria"	"000-000-004"	"backend-developer"	"000-000-004"	

TestingCoverageStatusMap	
Coverage	Status
100	"closed"
75	"in progress"
50	"in progress"
25	"in progress"
0	"open"

SecurityCvssSeverityMap	
CVSS	Severity
0.1	"Low"
3.8	"Low"
4.5	"Medium"
7.8	"High"
9.5	"Critical"

NotificationTypeDescMap	
Type	Description
"request"	"A user has requested to join your team!"
"deletion"	"A user in your team has deleted their profile"
"addition"	"A new user joined your team"
"approval"	"Your request to join a team has been approved"
"denial"	"Your request to join a team has been denied"

Notification			
<u>NotificationID</u>	Type	Time	ReviewingUserID
"000-000-001"	"request"	2023-01-01	"000-000-001"
"000-000-002"	"deletion"	2023-01-01	"000-000-001"
"000-000-003"	"addition"	2023-01-01	"000-000-001"
"000-000-004"	"approval"	2023-01-01	"000-000-001"
"000-000-005"	"denial"	2023-01-01	"000-000-003"

RequestToJoin					
<u>RequestID</u>	Status	Time	<u>ProjectID</u>	<u>NotificationID</u>	<u>RequestingUserID</u>
"000-000-001"	"approved"	2023-01-01	"000-000-005"	"000-000-006"	"000-000-005"
"000-000-002"	"approved"	2023-01-01	"000-000-004"	"000-000-007"	"000-000-004"
"000-000-003"	"approved"	2023-01-01	"000-000-001"	"000-000-008"	"000-000-003"
"000-000-004"	"approved"	2023-01-01	"000-000-005"	"000-000-009"	"000-000-002"
"000-000-005"	"denied"	2023-01-01	"000-000-001"	"000-000-010"	"000-000-005"
"000-000-006"	"pending"	2023-01-01	"000-000-003"	"000-000-011"	"000-000-001"

UserHasRole	
<u>UserID</u>	<u>RoleName</u>
"000-000-001"	"backend-developer"
"000-000-002"	"qa"
"000-000-003"	"frontend-developer"
"000-000-004"	"backend-developer"
"000-000-005"	"backend-developer"

PostNeedsRole	
<u>PostID</u>	<u>RoleName</u>
"000-000-001"	"qa"
"000-000-002"	"backend-developer"
"000-000-003"	"backend-developer"
"000-000-004"	"project-manager"
"000-000-005"	"backend-developer"
"000-000-005"	"frontend-developer"

TeamHasPeople			
<u>ProjectID</u>	<u>TeamName</u>	<u>RoleName</u>	<u>UserID</u>
"000-000-003"	"eagle"	"frontend-developer"	"000-000-001"
"000-000-003"	"eagle"	"backend-developer"	"000-000-001"
"000-000-003"	"eagle"	"designer"	"000-000-001"
"000-000-003"	"eagle"	"project-manager"	"000-000-001"
"000-000-003"	"eagle"	"qa"	"000-000-001"