Experiment No.:	3
Aim:	To draw the Class diagram of an Employee Onboarding software.
Name:	Deepanshu Aggarwal, Kunal Bhatia
UID No.:	2021300002, 2021300010
Class & Division:	TE-CE [COMPS A(A)]

Problem Statement:
A platform for individuals to report and access data on criminal activities in their community is what the Crime Reporting Website wants to offer. It makes it easier to report crimes, see crime stats, and have conversations about incidences. A smooth exchange of information and communication is made possible by the website's service to both administrators and registered users. While administrators have control over user accounts and can handle reported occurrences, users can report crimes, examine in-depth crime reports, sign up for notifications, and analyze crime data.

Noun Phrases:
User, Administrator, Crime Report, Crime Incident, Comment, Notification

Classes:
User, Administrator, CrimeReport, Crime Incident, Comment, Notification

Verb Phrases:
1. Users can register an account.
2. Users can log in to their account.
3. Users can report a crime incident.
4. Users can view a crime report.
5. Users can post comments on a crime report.
6. Users can subscribe to crime notifications.
7. Users can generate crime analysis reports.
8. Administrators can manage user accounts.
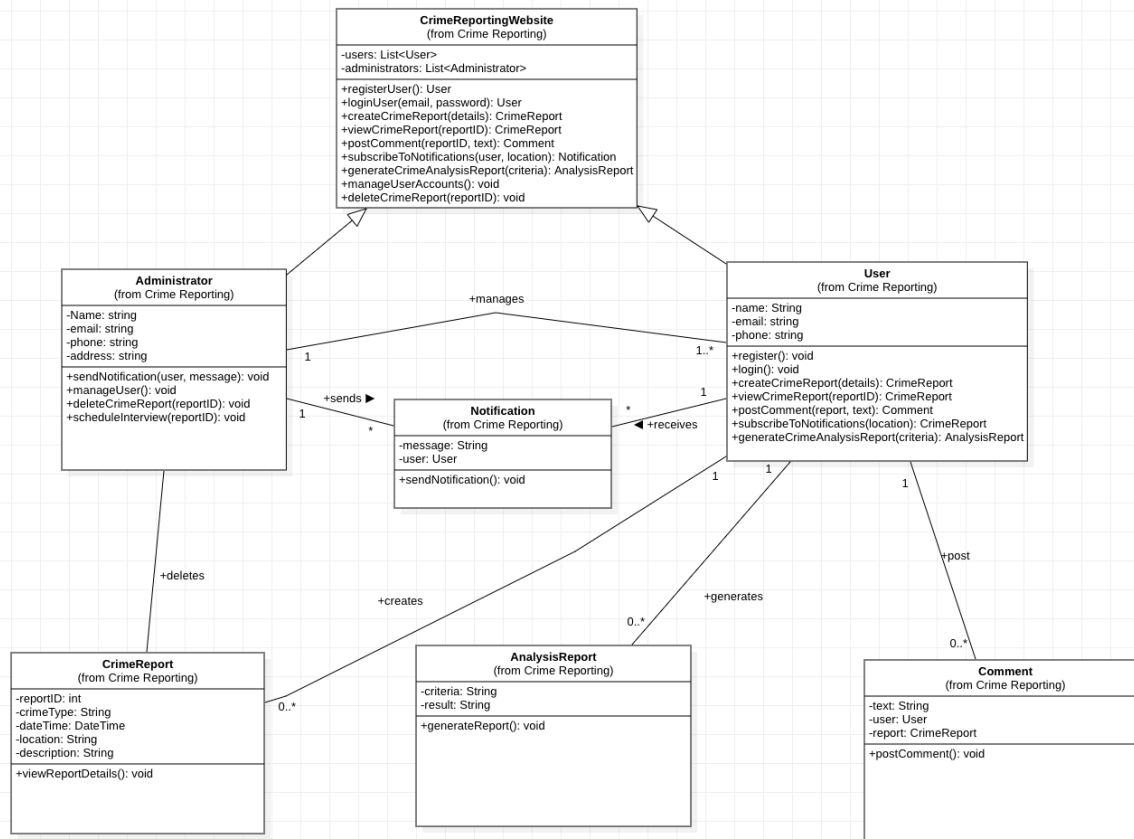9. Administrators can delete crime reports.

Features:
1. User registration: After entering their name, email address, and password, users can create an account.
2. User Login: Admins and registered users can safely log in with their email addresses and passwords.
3. Crime Reporting: Registered users can report crimes by including information about the crime's nature, date, time, location, description, and supporting documentation.
4. Viewing of Crime Reports: Users get access to detailed crime reports that include the kind of offence, the date, the time, the place, a description, and user comments.
5. Commenting: By leaving comments on crime reports, users can participate in conversations.
6. Notification Subscription: Users can sign up to receive notifications and real-time crime updates for particular places.
7. Crime Analysis: Registered users can create crime analysis reports based on a variety of criteria, including the type of incident, the time period, and the location of the crime.
8. User Account Management: Users' accounts, including user registration, account deletion, and user profile management, are under the administration's authority.
9. Users and administrators both have the option to erase criminal reports if necessary.
10. User Roles: With varying levels of access and rights, the system distinguishes between regular users and administrators.

Relations:
1) Users and administrators are system participants. As a result, they can be classified as a "User" class.
2) Users have the option of reporting crimes or not.
Users can choose to view or not to view criminal reports.
4) Administrators have the ability to notify users of additional information or actions in connection with reported occurrences.
5) Users have the option to check their notifications.
6) To handle reported occurrences, administrators might form numerous interview panels.
7) Administrators have the option of scheduling multiple interviews on reported instances.
8) Reports on crimes may be produced by one or more administrators for analysis.
9) A crime incidence may be reported by one or more users.
10) Crime reports can be accessed and viewed by one or more people.

Class Diagram:

CrimeReportingWebsite
(from Crime Reporting)

-users: List<User>
-administrators: List<Administrator>

+registerUser(): User
+loginUser(email, password): User
+createCrimeReport(details): CrimeReport
+viewCrimeReport(reportID): CrimeReport
+postComment(reportID, text): Comment
+subscribeToNotifications(user, location): Notification
+generateCrimeAnalysisReport(criteria): AnalysisReport
+manageUserAccounts(): void
+deleteCrimeReport(reportID): void

Administrator
(from Crime Reporting)

-Name: string
-email: string
-phone: string
-address: string

+sendNotification(user, message): void
+manageUser(): void
+deleteCrimeReport(reportID): void
+scheduleInterview(reportID): void

User
(from Crime Reporting)

-name: String
-email: string
-phone: string

+register(): void
+login(): void
+createCrimeReport(details): CrimeReport
+viewCrimeReport(reportID): CrimeReport
+postComment(report, text): Comment
+subscribeToNotifications(location): CrimeReport
+generateCrimeAnalysisReport(criteria): AnalysisReport

+manages   +sends ▶   +receives ◀

Notification
(from Crime Reporting)

-message: String
-user: User

+sendNotification(): void

+deletes   +creates   +generates   +post

CrimeReport
(from Crime Reporting)

-reportID: int
-crimeType: String
-dateTime: DateTime
-location: String
-description: String

+viewReportDetails(): void

AnalysisReport
(from Crime Reporting)

-criteria: String
-result: String

+generateReport(): void

Comment
(from Crime Reporting)

-text: String
-user: User
-report: CrimeReport

+postComment(): void

Code Snippets:

```
public class User {
    private String name;
    private String email;

    public User(String name, String email) {
        this.name = name;
        this.email = email;
    }

    public void register() {
        // Code for user registration
    }

    public void login() {
        // Code for user login
    }
```

```java
    public void subscribeToNotifications() {
        // Code for subscribing to crime notifications
    }

    // Other user-related methods
}

public class Administrator {
    private String name;
    private String email;

    public Administrator(String name, String email) {
        this.name = name;
        this.email = email;
    }

    public void sendNotification(User user, String message) {
        // Code for sending notifications to users
    }

    public void manageUserAccounts() {
        // Code for managing user accounts
    }

    // Other administrator-related methods
}

public class CrimeReport {
    private int reportID;
    private String crimeType;
    private DateTime dateTime;
    private String location;
    private String description;

    public CrimeReport(int reportID, String crimeType, DateTime dateTime, String location,
String description) {
        this.reportID = reportID;
        this.crimeType = crimeType;
        this.dateTime = dateTime;
        this.location = location;
        this.description = description;
    }

    public void viewReportDetails() {
```

```java
        // Code for displaying crime report details
    }

    // Other crime report-related methods
}

public class Comment {
    private String text;
    private User user;

    public Comment(String text, User user) {
        this.text = text;
        this.user = user;
    }

    public void postComment(CrimeReport report) {
        // Code for posting comments on crime reports
    }

    // Other comment-related methods
}

public class Notification {
    private String message;
    private User user;

    public Notification(String message, User user) {
        this.message = message;
        this.user = user;
    }

    public void sendNotification() {
        // Code for sending notifications to users
    }

    // Other notification-related methods
}
```

Conclusion:

This class diagram represents the core entities and their relationships in the Crime Reporting Website. It defines the classes User, Administrator, CrimeReport, Comment, and Notification, along with their attributes and methods. The diagram serves as a foundation for the development of the website's functionality, allowing users to report crimes, view reports, engage in discussions, and receive notifications, while administrators can manage user accounts and reported incidents.