問1　安全な Web アプリケーションの開発に関する次の記述を読んで，設問 1～5 に答えよ。

X 社は従業員数 500 名のソフトウェア会社で，10 本程度の人事関連のソフトウェアパッケージ（以下，パッケージという）を開発，保守している。X 社には，顧客から，パッケージを購入せず，利用料を払ってアプリケーションサービス（以下，サービスという）を受けられるようにしてもらえないかという問合せが増えている。また，競合他社の中には，そのようなサービスを提供するところも出てきている。このような状況を受けて，X 社の経営陣は，パッケージに加え，サービスも商品ラインナップに加える方針を決定した。

X 社は，手始めに，既存のパッケージではなく，新規の分野でサービスを提供することにした。具体的には，既存顧客から要望が出ていた，店舗で働くパートタイマやアルバイトの勤怠管理を行うシステム（以下，勤怠管理システムという）でサービスを提供することにした。また，システムの稼働率を高めるために，単一のシステムを複数の顧客で共用するマルチテナントのシステムとする方針を決定した。

一方，X 社はこれまでイントラネット向けのパッケージ開発を専業としてきたので，インターネットに公開する安全な Web アプリケーションの開発ノウハウがなかった。X 社の経営陣は，今後，サービス提供を拡大していくためには，安全な Web アプリケーション開発のノウハウ獲得が不可欠であると考えた。そこで，今回の開発プロジェクトを通じて，そのノウハウを身につけ，今後のほかの開発プロジェクトにも適用していくことにした。特に，今回のシステムはマルチテナントのシステムなので，顧客間でデータが漏えいしないよう，十分に検討して対策を行うことにした。

X 社のシステム関連部署は四つあり，パッケージ開発部，システム技術部，システム運用部及びプロジェクト管理部である。パッケージ開発部はパッケージの開発・保守を行っている。システム技術部は，社内のネットワーク，サーバ，ミドルウェアなどのインフラ（以下，インフラという）の設計・構築を担当している。システム運用部は社内のインフラの運用を担当しており，セキュリティパッチ適用などの情報セキュリティ運用も行っている。プロジェクト管理部は，社内の開発プロジェクトを管理する部署である。

X 社は勤怠管理システムの開発プロジェクトを立ち上げた。プロジェクト体制として，プロジェクトマネージャをプロジェクト管理部の P 課長が務め，その下に，開発チーム及びインフラチームが作られた。開発チームは，アプリケーションの設計・開発を担当し，開発の一部は複数の外部委託先に委託した。そして，開発チームのリーダはパッケージ開発部の B さんが務めることになった。また，アプリケーション開発のセキュリティ対策を支援するために，システム技術部から D 課長と F 主任が開発チームに参加することになった。インフラチームは，システム技術部のメンバで構成され，インフラの設計・構築及び本番稼働までの保守・運用を担当し，インフラチームのリーダはシステム技術部の E さんが務めることになった。本番稼働後は，勤怠管理システムの運用をシステム運用部に引き継ぐことになっている。

〔勤怠管理システムの概要〕

勤怠管理システムは，店舗ごとの従業員情報（氏名，従業員番号など）の管理機能，勤務予定の作成及び勤務実績の記録と参照のための機能，並びに給与システムへの勤怠データ出力機能をもつ。勤怠データは店舗ごとに管理される。また，複数店舗の情報を集約する機能があり，システムを利用する 1 顧客当たり 50 店舗まで管理することができる。

〔勤怠管理システムのネットワーク構成〕

勤怠管理システムのネットワーク構成を図 1 に，ファイアウォール（以下，FW という）のアクセス制御ルールを表に示す。Web サーバ上ではミドルウェアとアプリケーションが稼働する。

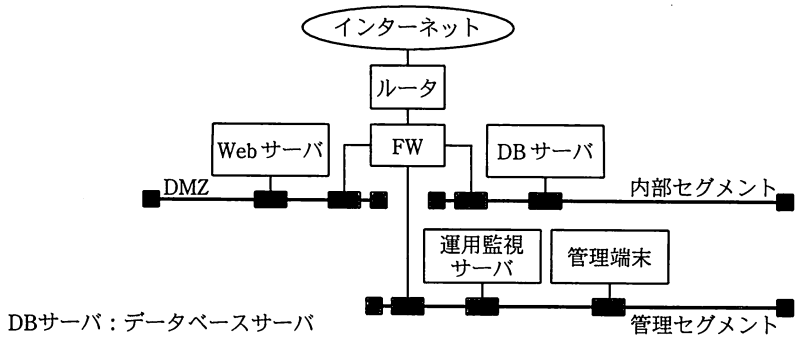

図 1　勤怠管理システムのネットワーク構成

表　FW のアクセス制御ルール

項番	プロトコル	送信元	あて先	あて先ポート番号	動作
1	TCP	インターネット	Web サーバ	80，443	許可
2	TCP	Web サーバ	DB サーバ	DB(1)	許可
3	すべて	管理セグメント	Web サーバ	すべて	許可
4	すべて	管理セグメント	DB サーバ	すべて	許可
5	すべて	すべて	すべて	すべて	拒否

注(1)　“DB” は，データベースアクセスに必要なポート番号である。
注　項番の昇順に，最初に一致したルールが適用される。

〔プロジェクト計画の策定〕

本開発プロジェクトの開始に当たって，F 主任は P 課長から提示されたプロジェクト計画書を確認した。

図 2 は，プロジェクト計画書中で示された，開発プロジェクトのスケジュール概要（開発チームの関連部分を抜粋）である。要件定義には，セキュリティ要件の定義も含まれている。また，システムテスト期間の後半で本番稼働前に，ルータ，FW，各サーバ及び Web アプリケーションの脆弱性テストを外部の専門家に依頼する計画となっていた。

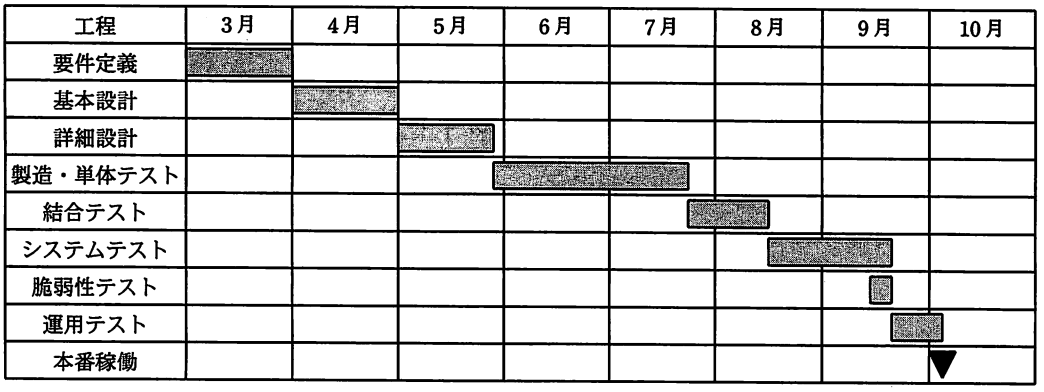

図 2　開発プロジェクトのスケジュール概要（抜粋）

F 主任は，①本番稼働直前の脆弱性テストだけで安全な Web アプリケーションを開発しようとする計画には，プロジェクトマネジメント上，問題があると考え，P 課長に計画の変更を提案した。P 課長は，F 主任の提案を受けて，②レビューを設計工程（基本設計，詳細設計）及び製造・単体テスト工程に追加する形で計画を変更した。

なお，外部委託先については，X 社から開発ガイドライン準拠チェックリスト（以下，チェックリストという）を提供し，外部委託先からは工程ごとにチェック結果を提出してもらうことにした。

〔コーディングルールの作成〕

次に，F 主任が本開発プロジェクトでのガイドラインを確認したところ，開発チームでは X 社の標準である開発ガイドラインを適用する計画となっていた。X 社では開発ガイドラインを制定し，各パッケージの開発・保守に適用している。図 3 は開発ガイドラインの SQL 処理に関する記載の抜粋である。

〔SQL 処理〕
SQL 文の組立てには，プレースホルダ及びバインド機構を使用すること

図 3　開発ガイドラインの SQL 処理に関する記載（抜粋）

F 主任は，基本設計に先立って，開発ガイドラインを基に，開発言語として使用する Java の具体的なコーディングルールも作成した。図 4 は，F 主任の作成したコーディングルールのうち SQL 処理に関する記載の抜粋である。

〔SQL 処理〕
1. SQL 文の組立てには，プレースホルダ及びバインド機構を使用すること
（サンプル）

```
String param = "suzuki"; //検索クエリの入力値
// SELECT * FROM atable WHERE name="suzuki"; を実行する。
java.sql.Connection con = java.sql.DriverManager.getConnection(url, userName, password);
String sql = "SELECT * FROM atable WHERE name=?";
[ a ]
stmt.setString(1, param);
[ b ]
```

図 4　コーディングルールのうち SQL 処理に関する記載（抜粋）

F 主任は，コーディングルールを作成後，プロジェクトメンバが参照できるようにファイルサーバに配置した。その後，開発プロジェクトは基本設計に入り，勤怠管理システムの設計が開発チームを中心に進められた。

〔勤怠管理システムのデータベース構成〕

勤怠管理システムはマルチテナントのシステムなので，一つのシステムの中で複数の顧客のデータを扱うが，顧客間でデータは分離する。顧客間でデータを分離する方法として，開発チームは，③データベースの各テーブルで，顧客を識別するカラムを追加する方式を採用した。具体的には，顧客単位で区別するために，顧客 ID を各テーブルに入れることにした。詳細設計におけるデータベース論理設計結果のうち，主なテーブルの概要を図 5 に，その E-R 図を図 6 に示す。

テーブル名：顧客テーブル

列名	データ型	内容	備考
CUST_ID	文字列	顧客 ID	（省略）
CUST_NAME	文字列	顧客名	（省略）
⋮	⋮	⋮	⋮

テーブル名：従業員テーブル

列名	データ型	内容	備考
CUST_ID	文字列	顧客 ID	（省略）
USR_ID	文字列	利用者 ID	（省略）
STORE_ID	文字列	店舗 ID	（省略）
PASSWORD	文字列	パスワード	ハッシュ値
⋮	⋮	⋮	⋮

テーブル名：店舗テーブル

列名	データ型	内容	備考
CUST_ID	文字列	顧客 ID	（省略）
STORE_ID	文字列	店舗 ID	（省略）
STORE_NAME	文字列	店舗名	（省略）
⋮	⋮	⋮	⋮

テーブル名：勤務実績テーブル

列名	データ型	内容	備考
CUST_ID	文字列	顧客 ID	（省略）
USR_ID	文字列	利用者 ID	（省略）
K_DATE	日付	日付	（省略）
START_TIME	時刻	開始時刻	（省略）
END_TIME	時刻	終了時刻	（省略）
⋮	⋮	⋮	⋮

注　下線は主キーを表す。

図 5　主なテーブルの概要

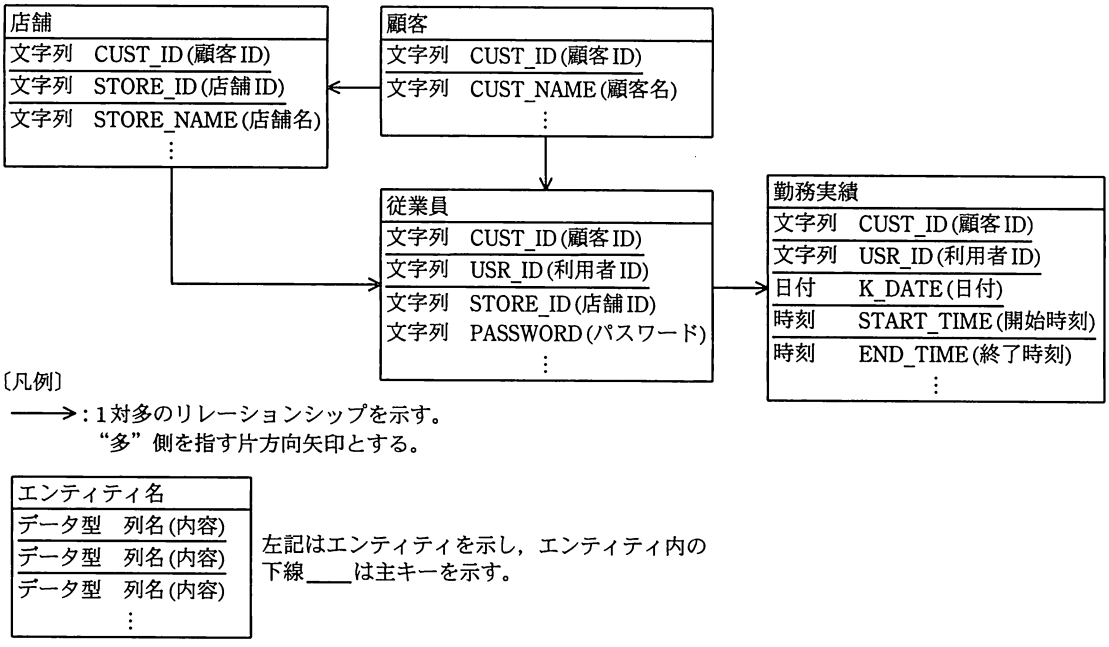

図6　主なテーブルのE-R図

〔アプリケーション設計のレビュー〕

詳細設計工程に進んだところで，社内メンバにセキュリティの観点でのレビュー経験がないことから，社内でのレビューではセキュリティ上の設計ミスやプログラム中の脆弱性を発見しきれないのではないかとBさんは考えた。そこで，セキュリティコンサルティング会社のH社の支援を受けることを提案し，承認された。H社は，アプリケーションセキュリティの専門家であるGさんを担当者としてアサインした。

詳細設計終了後，Gさんは，Bさん，F主任とともに詳細設計結果に対するレビューを行ったところ，勤務実績表示の処理に問題があることを発見した。問題があった勤務実績表示の処理を図7に示す。従業員選択画面には，ログイン中の利用者が所属している店舗の全従業員がリストに表示される。従業員をリストから選択すると，選択された従業員の勤務実績が勤務実績表示画面に表示される処理となっている。図7の勤務実績表示のSQL文は，図中の勤務実績表示画面を表示するためのSQL文で，【選択した従業員の利用者ID】には，従業員選択画面のリストで選択した従業員の利用者IDが指定される。

〔従業員選択画面〕

ログイン：○○店舗　鈴木　良子

勤務実績を表示する従業員を
選択してください。

従業員選択

▼

鈴木　良子
山田　太郎
⋮

〔勤務実績表示画面〕

ログイン：○○店舗　鈴木　良子

2010年10月

山田　太郎 さんの勤務実績

10/1	10:05～16:30
10/2	9:45～16:00
10/3	10:00～17:00

〔勤務実績表示の SQL 文〕

```
SELECT USR_ID, K_DATE, START_TIME, END_TIME
FROM 勤務実績テーブル
WHERE CUST_ID = 【ログイン中の利用者の顧客 ID】
    AND USR_ID = 【選択した従業員の利用者 ID】
    AND K_DATE >= '2010/10/01' AND K_DATE < '2010/11/01';
```

図７　問題があった勤務実績表示の処理

Ｇさん：勤務実績表示画面では，他人の勤務実績を参照することができますね。

Ｆ主任：はい，この機能では，同じ店舗に所属しているほかの従業員の勤務実績が参照できるようになっています。

Ｇさん：同じ店舗だけならばよいのですが，このままではほかの店舗の従業員の勤務実績も参照することができます。具体的には，従業員選択画面のリストから従業員を選択して表示する処理に問題があります。

Ｆ主任：リストには，店舗に所属している従業員の一覧を表示し，HTML の OPTION タグの VALUE 属性には，選択した従業員の利用者 ID を指定するようになっています。従業員の一覧では，ログイン中の利用者と同じ店舗に所属している従業員だけが表示されますので，ほかの店舗の従業員を選択することはできないのではないでしょうか。

Ｇさん：いいえ，可能です。選択された従業員の勤務実績を表示する処理では，VALUE 属性として送信されてきた利用者 ID に対し，データの型をチェックした後，図７中の SQL 文にそのまま使用する設計なので，これでは<u>④ほかの店舗の従業員の勤務実績を権限なく閲覧される可能性があります</u>。

Ｆ主任：なるほど。分かりました。早速，対策を検討して対応します。

Ｆ主任はＧさんのアドバイスを基に対策を考え，同様の問題がほかにないかどうかの確認も含めて，Ｂさんを通して対策を開発チームに依頼した。

〔SQL インジェクション脆弱性の発見と対策〕

製造・単体テスト，結合テストが終了し，システムテストに入った。一方，F 主任はプロジェクト計画に基づき，外部の専門家による脆弱性テストを実施した。

脆弱性テストの結果，ある処理で SQL インジェクション脆弱性の作り込みが発見された。開発チームリーダの B さんに確認したところ，開発プロジェクトの途中で，外部委託先 Z 社に新規に開発を委託しており，その開発部分で発見されたことが分かった。

Z 社から提出を受けていたチェック結果には“コーディングルールに適合”と記入されていたので，F 主任は Z 社に事実関係を確認した。Z 社は，X 社のコーディングルールの内容は十分に理解していたが，コーディングルール違反をレビューで見逃したままチェック結果を X 社に提出していたと回答した。F 主任は B さんに報告した。B さんは，Z 社の開発部分全体についてコーディングルールの遵守を確認するよう，開発チームに指示した。

〔ミドルウェアの脆弱性〕

システムテストが終了し，システム運用部が加わり勤怠管理システムをインターネットに接続して運用テストを進めていた。本番稼働の 2 日前に，システム運用部のメンバがインターネットのニュースサイトを見ていたところ，勤怠管理システムで使用しているミドルウェアにおける脆弱性が 1 か月前に公表されていることを，偶然見つけた。

X 社のこれまでのパッケージ開発では，テスト期間中にミドルウェアのバージョンを見直すことはなかった。システム運用部は，インフラチームのリーダである E さんに相談した。E さんはベンダが発表している脆弱性情報と勤怠管理システムのミドルウェアのバージョンを確認し，その内容と影響を報告書にまとめた。図 8 は E さんのまとめた報告書である。

○概要

9 月 1 日に，勤怠管理システムで使用しているミドルウェアにおける脆弱性情報及びセキュリティパッチが公表された。

○脆弱性の内容

ミドルウェアには管理画面があり，TCP ポート番号 8080 で利用できるようになっている。ネットワーク経由でこの管理画面にアクセスすることによって，ログインしなくても DoS 攻撃が可能である。ただし，ベンダはどのようなパケットで DoS 攻撃が可能か公表しておらず，ベンダのその後の報告では，10 月 1 日現在，脆弱性による不正な動作を再現する [c] は公開されていない。また，任意のコード実行が可能となる脆弱性ではない。

○影響

勤怠管理システムで使用しているミドルウェアは，この脆弱性の影響を受けるバージョンである。ただし，セキュリティ対策として，当初から次の対策を実施していたので，インターネットから直接に攻撃を受けるわけではない。

実施済対策：[d]

○根本対策

ベンダからのセキュリティパッチを適用すること。ミドルウェアのセキュリティパッチ適用時の動作検証には 3 日掛かる。

○監視方法

IDS（侵入検知システム）を導入していないが，現在公開されている情報によれば，この脆弱性が悪用された場合には，勤怠管理システムにおいてサービス停止や性能低下が発生するので，通常のサービス稼働監視によって検知することができる。

図 8　E さんのまとめた報告書

X 社内で，関係者で検討した結果，E さんのまとめた報告書にある脆弱性の内容及び影響に変化がなければ，サービスの本番移行は予定どおり実施するとの判断をした。また，根本対策として，次回の定期メンテナンスでセキュリティパッチを適用することにした。

その後，本番稼働までの 2 日間，脆弱性情報の更新を確認したが，特に更新はなかった。また，サービス稼働監視でも異常は発見されず，無事，本番稼働を迎えた。

〔本番稼働後の報告〕

本番稼働後，D 課長はプロジェクト管理部に対して，今回の開発プロジェクトで発生したセキュリティ上の問題点と対応について報告を行った。プロジェクト管理部は，SQL インジェクション脆弱性の作り込みを踏まえた⑤開発委託先管理の問題点と，開発プロジェクトにおける本番稼働前のシステムに対する⑥脆弱性情報の管理の問題点を指摘し，今回の開発プロジェクトで得られた経験を，今後のシステムの開発，運用，保守に生かすため，D 課長に具体的な解決策の提案を依頼した。

D 課長は P 課長，B さん，F 主任にも解決策の検討を依頼し，検討結果を取りまとめ，プロジェクト管理部に提案を行った。X 社の経営陣は提案を承認し，X 社のプロセスとしてシステムの開発，運用，保守に適用することにした。

設問1　〔コーディングルールの作成〕について，図4中の［　a　］，［　b　］に入れる適切な字句を解答群の中から選び，記号で答えよ。

解答群

ア　`java.sql.PreparedStatement stmt = con.prepareStatement(sql);`

イ　`java.sql.ResultSet rs = stmt.executeQuery();`

ウ　`java.sql.ResultSet rs = stmt.executeQuery(sql);`

エ　`java.sql.Statement stmt = con.bind(sql);`

オ　`java.sql.Statement stmt = con.doFilter(sql);`

設問2　〔プロジェクト計画の策定〕について，(1)～(3)に答えよ。

(1) 本文中の下線①でF主任が，問題があると指摘した理由を，35字以内で述べよ。

(2) 本文中の下線②でP課長は，設計工程（基本設計，詳細設計）にレビューを追加した。その内容を，40字以内で具体的に述べよ。

(3) 本文中の下線②でP課長は，製造・単体テスト工程にレビューを追加した。その内容を，50字以内で具体的に述べよ。

設問3　〔アプリケーション設計のレビュー〕で発見された，勤務実績表示の処理の問題について，(1)，(2)に答えよ。

(1) 本文中の下線④の指摘は，具体的にどのようなHTTPリクエストによって閲覧されると指摘したものか。55字以内で述べよ。

(2) この問題を避けるためには，データの型をチェックした後，どのようなチェック処理を行うべきか。60字以内で述べよ。

設問4　SQLインジェクション脆弱性の対策について，(1)，(2)に答えよ。

(1) SQLインジェクションが発生したときでも，ほかの顧客のデータをSQLで操作されにくいようにするためには，本文中の下線③の方式をどのように変更すべきか。45字以内で述べよ。

(2) 本文中の下線⑤の開発委託先管理の問題点を解決するために，X社が行うべきことを50字以内で述べよ。

設問5　ミドルウェアの脆弱性発見時の対応について，(1)～(3)に答えよ。

(1)　図8中の［　c　］に入れる適切な字句を解答群の中から選び，記号で答えよ。

解答群

ア　エクスプロイトコード　　イ　セキュリティパッチ

ウ　セキュリティポリシ　　エ　ゼロデイ攻撃手法

オ　バッファオーバフロー

(2)　図8中の［　d　］に入れる適切な実施済対策の内容を，50字以内で述べよ。

(3)　本文中の下線⑥の問題点に対する解決策を，今後の開発プロジェクトの在り方の観点から，40字以内で述べよ。