HW6: Appointment Reservation System

COMPUTER SCIENCE & ENGINEERING
UNIVERSITY of WASHINGTON

Agenda

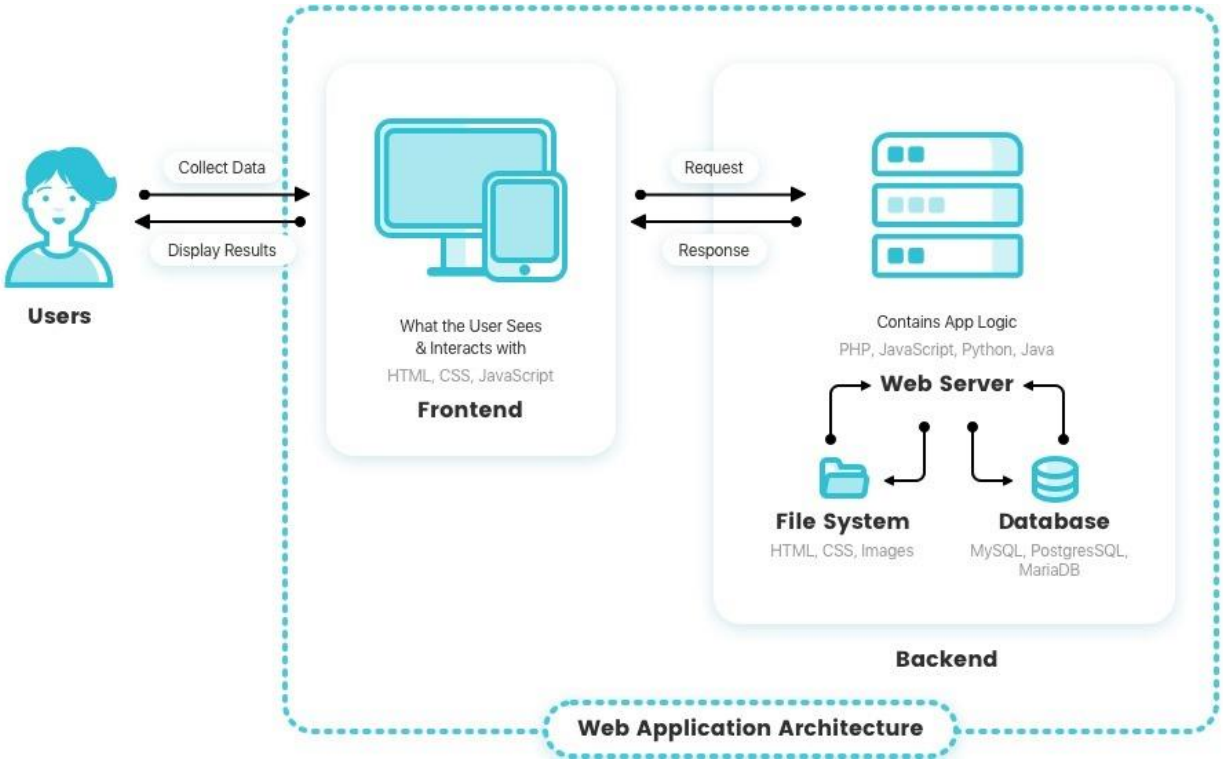
- **Assignment Structure**
- **Assignment Introduction**
- **Exceptions**
- **Running SQL Queries**
- **Handling Passwords**
- **Starter Code Walkthrough**

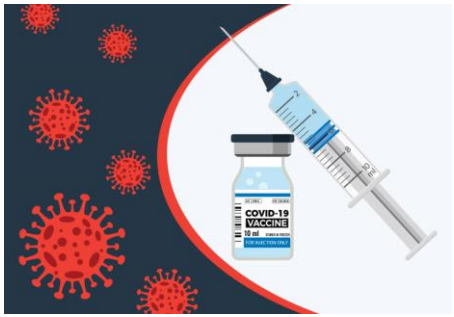
HW6 Structure

- **Setup:**
  - **Starter code, dependencies, and database connection;**
  - **due May 25 (screenshot);**
- **Part 1:**
  - **Design and implementation;**
  - **due May 27 (suggested, no submission is required);**
- **Part 2:**
  - **Implementation;**
  - **due June 3 (Part 1 & Part 2);**

Assignment Introduction

- **Appointment scheduler application for vaccinations;**
  - **With a command-line interface (easier to implement);**
  - **Data hosted on Azure.**





Assignment Introduction

- **Gain experience with application development:**
  - **(Planning), Analysis, Design, Implementation, Testing, (Support);**
- **Use database knowledge learned in class to solve real-world challenges:**
  - **Database design (ERD, (De)normalization);**
  - **Querying information (SQL queries, transactions);**
  - **Performance optimization (Cost estimation);**
- **Two versions available:**
  - **Java (JDBC);**
  - **Python (pymssql);**
- **Our solution is about 600 lines of code;**
  - **Don't wait, start ASAP!**
  - **Including ~300 lines of starter code.**
- **Place this project on your resume!**
  - **UW Daily: Where is the 'world-class' UW education?**

Not all “errors” should be failures

- **Some “error” cases:**
  1. **Misuse of your code;**
     - **E.g., precondition violation.**
     - **Expecting a String, got an integer instead.**
     - **Should be a failure.**
  2. **Errors in your code vs reasoning;**
     - **E.g., Representation Invariant fails to hold.**
     - **(Representation Invariant: A condition that must be true over all valid concrete representations of a class.)**
     - **Bank balance should never be negative but got negative somehow after a transaction.**
     - **Should be a failure.**
  3. **Unexpected resource problems;**
     - **E.g., missing files, server offline.**
     - **Should not be a failure.**

Errors vs. Exceptions

- **Error: an illegal operation performed by the user which results in the abnormal working of the program.**
  - **Compile-time error;**
  - **Runtime error;**
  - **Logical error;**
- **Exceptions: an unexpected event that disrupts the normal flow of the program's instructions.**
  - **Checked exceptions: exceptions that are checked at compile time.**
    - **E.g., IOException, SQLException.**
    - **A program can recover and programmers are expected to check for those exceptions.**
    - **Required by language to handle the exception.**
  - **Unchecked exceptions: those are "basically" runtime errors.**
    - **E.g., ArrayIndexOutOfBoundException.**

Java: JDBC

```
String selectUsername = "SELECT * FROM Caregivers WHERE Username = ?";
try {
    PreparedStatement statement = con.prepareStatement(selectUsername);
    statement.setString( parameterIndex: 1, username);
    ResultSet resultSet = statement.executeQuery();
    // returns false if the cursor is not before the first record or if there are no rows in the ResultSet.
    return resultSet.isBeforeFirst();
} catch (SQLException e) {
    System.out.println("Error occurred when checking username");
    e.printStackTrace();
} finally {
    cm.closeConnection();
}
```

- **Try-Catch block: Catch and handle exception.**
- **Finally: code inside the finally clause will always be executed.**

Java: JDBC

- **API library that allows Java programs to access database management systems.**
- **PreparedStatement: prevent SQL Injection attacks.**

```
Statement withoutPlaceholder = con.createStatement();
withoutPlaceholder.execute( sql: "INSERT INTO students VALUES('\" + userInput + \"')");

PreparedStatement withPlaceholder = con.prepareStatement( sql: "INSERT INTO student VALUES(?)");
withPlaceholder.setString( parameterIndex: 1, userInput);
withPlaceholder.execute();
```

- **What if the user input was**
  “Robert'); DROP TABLE students; --”?

Password Hashing and Salt

- **Instead of user password, store Hash(password).**
  - **System does not store actual passwords.**
  - **When user enters password, compute its hash and compare with entry in password file.**
- **Not entirely safe: Dictionary Attack.**
  - **Many passwords come from a small dictionary.**
  - **Attacker pre-compute Hash(password) for all words in the dictionary.**
- **Password salting**