Software Engineering

Case Study: Student Attendance System
A school wants a system that:
1.Records daily attendance.
2.Generates monthly attendance reports
3.Notifies parents when a student is absent

Question 1:
Explain 6 main phases of waterfall model for this system.

Answers:
**Six Main Phases of the Waterfall Model**

**1.Requirement Analysis:**

- Collect and understand all requirements from the school specially from faculties.
- Decide what the system should do any task.
- Requirements are clearly analyzed.

**In this Attendance System the Requirements are:**

1. Attendance recording
2. Monthly reports
3. Parent notifications

**Example task:**
Meeting with teachers and admin to document attendance rules.

**2. System Design:**

- Plan how the system will work technically.
- Decide database structure, user roles, and system flow.
- This design is based on the user requirements.

**In this system**:

1. Design tables for students, attendance, parents
2. Decide UI screens (login, attendance page, report page)

Example task:
Creating ER diagrams or system architecture diagrams.

**3.Implementation** :(Coding / Development)

- Developers write actual code based on the design.
- Each module will be developed according to the system design.
- Each module is tested individually during development.

**For this system:**

1. Write code to mark attendance of students.
2. Generate reports.
3. Send SMS/email notifications to Parents.

Example task:
Writing Java/Python/JS code for attendance entry.

**4.Testing:**

- Check whether the system works correctly or not.
- Find and fix errors (bugs).
- This ensure that weather the system meets the user requirements or not.

**For this system:**

1. Test attendance marking
2. Test report generation
3. Test notification sending to the parent or not

Example task:
Testing if parents receive notification when a student is absent.

**5.Deployment:**

- The system is installed and made live for real users.
- System is made ready for actual use.
- The customer starts using the Software.

**For this system:**

1. Install system on school servers
2. Give access to teachers and admin

Example task:
Hosting the system and creating user accounts.

**6.Maintenance :**

- Fix issues and update the system after deployment.
- New features may be added if required.
- Performance improvements and updates are done in this phase.

For this system:

1. Fix bugs found later
2. Update features if school rules change

Example task:
Updating notification format or fixing report errors.

**Question 2:**
Give one example of task that would be done in each phase of the waterfall model for this system.

Answer:

| Phase name | Example |
|---|---|
| 1. Requirements Analysis | Collecting the requirement for this Attendance System from School. |
| 2. System Design | Design the data flow and interface of the students Attendance records logically. |
| 3. Implementation (Development) | Implementation of actual code means Students reports to convert that requirement into a Software Application. |
| 4. Testing | Testing the absent student's notification is sanded or not to their parents.<br>Daily records are maintained in this system.<br>And it correctly generates the monthly report to the Teachers and administration. |
| 5. Deployment | Publish or hosting the software in the school.<br>The teachers will start this application for a daily basis. |
| 6. Maintenance | Fixing the bugs and may update for adding the new rule of attendance system. |

**Question 3:**
Advantage and disadvantage of Waterfall model for this project:

**Answer:**

**Advantages:**

1. Clear structure and sequence
   Each phase is completed step by step, making the system easy to manage the Attendance system.

2. Easy documentation
   Detailed documentation is prepared at every phase, which helps schools understand system rules and reports.

3. Good for small and medium school systems

4. Attendance systems usually have fixed features, so Waterfall fits well.

**Disadvantages:**

1. No flexibility for new features
   If the school later wants biometric attendance or mobile app support, changes are difficult.

2. User feedback comes late
   Teachers and parents can test the system only after development is complete.

3. Testing happens late
   Errors in attendance calculations may be discovered only at the end.

4. Difficult to go back
   Once a phase is completed, revisiting It is complex.