Mateen Bashir
B-25028
Assignment # 01

Q.no.1:

Imagine that a government wants a software program that helps to keep track of the utilization of the country's vast mineral resources. Although the requirements put forward by the government were not very clear, a software company was tasked with the development of a prototype. The government found the prototype impressive, and asked it be extended to be the actual system that would be used. Discuss the pros and cons of taking this approach.

Ans:

Taking the approach of extending the prototype to become the actual system used for tracking the country's mineral resources has both pros and cons:

Pros:

1. Easy to use: People already like the prototype, so they'll find the final system easy to understand and use.

2. Saves money: It's cheaper to build on what we already have rather than starting from scratch.

3. Saves time: We've already done a lot of work on the prototype, so we can finish the system faster.

4. Listens to feedback: We can make sure the final system meets what the government wants by listening to their feedback on the prototype.

Cons:

1. Might not be enough: The prototype might be missing important things we need for the final system to work well.

2. Could cause problems later: We might have taken shortcuts when making the prototype, which could cause problems down the road.

3. Might not handle big changes: The prototype might not be able to handle lots of data or lots of people using it.

4. Needs to match what's needed: The government's needs might change, and we need to make sure the final system still matches what they want.

Q.no.2 :

Discover functional and nonfunctional requirements of the school management system, and draw a UML use case for the school management system and which technique do you suggest for requirement elicitation and why.

Ans:

Functional Requirements of a School Management System:

1. Student Management:
- Add new students
- Update student information
- Remove student records
- Make student ID cards

2. Grade Management:
- Record student grades
- Calculate GPA and create transcripts

3. Attendance Management:
- Keep track of student and staff attendance
- Generate attendance reports

4. Staff Management:
- Add new staff members
- Update staff details
- Assign roles
- Track staff attendance

5. Course Management:
- Create new courses
- Edit course details
- Assign courses to students
- Schedule classes and exams

6. Fee Management:
- Record fee payments
- Issue fee receipts
- Send payment reminders

Non-functional Requirements:

1. Performance: The system should work fast even when many people use it at the same time.
2. Security: Only authorized people should access sensitive information, and data should be protected from hackers.
3. Usability: The system should be easy to use for everyone, like teachers, students, and parents.
4. Reliability: The system should work all the time and should not lose data.
5. Scalability: The system should be able to handle more users and data as the school grows.
6. Compatibility: The system should work on different devices and systems.

For requirement gathering, I suggest using a mix of interviews and prototyping.

1.Interviews:

Talk directly to people like teachers, students, and parents to understand what they need from the system. It helps to ask specific questions and get detailed answers about what they want.

2.Prototyping:

Create simple versions of the system to show how it might work. This allows people to see and try out the ideas in action. It's like making a rough sketch before drawing the final picture.

Using both interviews and prototyping gives a good balance. Interviews help understand people's needs deeply, while prototyping helps to visualize and test ideas quickly. It's like talking and then showing, making sure everyone's ideas are heard and considered.