Explain 2 *example issues* each for different phases of app development discussed: What can go wrong?

Project Planning:

complex requirements - stakeholders giving unclear or incomplete goals, leading to misaligned expectations and potential project failure.

unrealistic timelines - overly promising on deadlines, without considering complexity of features or testing needs, may result in rushed and faulty app development.

User Interface (UI) Design:

complex designs: trying to incorporate too many features or flashy elements can overwhelm users and hinder usability.

lack of accessibility: ignoring accessibility standards, such as font sizes for visually impaired users or proper contrast, can alienate certain user groups.

Programming Languages specific to the target platform (like Java for Android or Swift for iOS):

incompatibility: some language or frameworks chosen may not be optimized for the target platform (e.g., attempting advanced animations in Java for Android that don’t perform well)

learning gap: developers may lack sufficient experience with platform-specific languages like Swift for iOS, leading to suboptimal coding practices.

Database Management:

query issues: database queries that are written poorly can slow down app performance, especially for apps with large datasets or high user demand.

security vulnerabilities: weak database security (e.g., not encrypting sensitive user data) can expose the app to data breaches.

Application Logic:

overcomplicated logic: writing overly complex or inefficient algorithms can make the app prone to bugs and difficult to maintain.

API integration is poor: incorrect implementation of third-party APIs or backend services can lead to failures in app functionality.

Testing Methodologies:

lacking test coverage: relying only on a few test cases can leave critical bugs undetected, especially for edge cases.

ignoring cross-platform testing: failing to test how the app functions across different devices and operating systems can result in poor user experiences on specific platforms.

Deployment Processes:

app store denials: apps may fail to meet store guidelines (e.g., Apple’s strict rules on user privacy), delaying the launch.

deployment rollbacks: errors during deployment, such as server misconfigurations or incomplete code uploads, can cause app crashes or downtime.

Maintenance Strategies:

lacking regular updates: failing to provide updates for bug fixes, new features, or platform requirements (e.g., iOS version changes) can render the app obsolete.

poor monitoring: without monitoring tools, issues like server outages or performance degradation can go unnoticed, frustrating users