First Part:

Creating a test plan for a mobile application (LinkedIn) involves defining high-level test scenarios and prioritizing them based on their potential business impact. Here are 10 high-level test scenarios with prioritization:

\*\*1. User Registration and Profile Creation\*\*

- Scenario: A new user registers and creates a LinkedIn profile, providing all required information.

- Business Impact: High (Critical for user acquisition)

\*\*2. Profile Editing and Updating\*\*

- Scenario: An existing user edits and updates their LinkedIn profile information, including adding new skills, updating job experience, and adding a profile picture.

- Business Impact: High (Critical for user engagement)

\*\*3. Connection Requests\*\*

- Scenario: A user sends a connection request to another user, and the other user accepts it.

- Business Impact: High (Key networking feature)

\*\*4. Messaging\*\*

- Scenario: Users exchange messages within the LinkedIn platform, including text, links, and attachments.

- Business Impact: High (Crucial for communication)

\*\*5. Job Search and Application\*\*

- Scenario: A user searches for job listings using various filters (location, industry, etc.) and applies for a job.

- Business Impact: High (Core feature for job seekers)

\*\*6. Content Posting and Sharing\*\*

- Scenario: Users create and share posts, articles, or updates on their LinkedIn feed, including text and multimedia content.

- Business Impact: Medium (Important for user engagement)

\*\*7. Notifications\*\*

- Scenario: Users receive and interact with notifications (e.g., connection requests, messages, job updates).

- Business Impact: Medium (Enhances user engagement)

\*\*8. Privacy Settings\*\*

- Scenario: Users configure privacy settings for their profile and content, controlling who can view their information.

- Business Impact: Medium (Critical for user data protection)

\*\*9. Search and Networking\*\*

- Scenario: Users search for other LinkedIn members using different criteria (e.g., name, industry, skills) and connect with them.

- Business Impact: Medium (Core functionality for networking)

\*\*10. Mobile App Security\*\*

- Scenario: Evaluate the security of the mobile app, including data encryption, secure login, and protection against common security vulnerabilities.

- Business Impact: High (Critical for user trust and data security)

\*\*Test Case Prioritization:\*\*

1. User Registration and Profile Creation

2. Connection Requests

3. User Registration and Profile Creation

4. Messaging

5. Job Search and Application

6. Mobile App Security

7. Privacy Settings

8. Profile Editing and Updating

9. Content Posting and Sharing

10. Search and Networking

This prioritization is based on the potential business impact of each scenario. High-priority scenarios are essential for user acquisition, engagement, and core functionality. Medium-priority scenarios are important but may not have as immediate and direct an impact on the business. Prioritizing test scenarios helps ensure that critical functionality is thoroughly tested and that resources are allocated effectively during the testing process.

\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\

Second Part:

1. Below are six bug reports for LinkedIn, including the required details:

\*\*Bug 1: Profile Picture Upload Issue\*\*

- \*\*Title\*\*: Unable to Upload Profile Picture

- \*\*Reproducible Steps\*\*:

1. Open LinkedIn mobile app.

2. Navigate to profile settings.

3. Click on "Edit Profile."

4. Attempt to upload a new profile picture.

- \*\*Attachments\*\*: Screenshots of the error message.

- \*\*Affected Devices\*\*: iPhone X, Samsung Galaxy

- \*\*Network\*\*: Wi-Fi and cellular

- \*\*Severity\*\*: Major

- \*\*Priority\*\*: High

- \*\*Impact\*\*: Users cannot update their profile pictures, impacting the completeness of their profiles and professional appearance.

\*\*Bug 2: Connection Request Error\*\*

- \*\*Title\*\*: Connection Request Not Sent

- \*\*Reproducible Steps\*\*:

1. Log in to LinkedIn mobile app.

2. Search for a user.

3. Click "Connect."

4. Observe that the request is not sent.

- \*\*Attachments\*\*: None

- \*\*Affected Devices\*\*: Various Android and iOS devices

- \*\*Network\*\*: Various networks

- \*\*Severity\*\*: Minor

- \*\*Priority\*\*: Medium

- \*\*Impact\*\*: Users cannot connect with others, affecting networking opportunities.

\*\*Bug 3: Inconsistent Notification Behavior\*\*

- \*\*Title\*\*: Inconsistent Notification Behavior

- \*\*Reproducible Steps\*\*:

1. Receive a connection request.

2. Click on the notification.

3. Sometimes, it opens the app directly; other times, it opens a browser window.

- \*\*Attachments\*\*: None

- \*\*Affected Devices\*\*: Various iOS devices

- \*\*Network\*\*: Wi-Fi and cellular

- \*\*Severity\*\*: Major

- \*\*Priority\*\*: High

- \*\*Impact\*\*: Inconsistent behavior disrupts the user experience and can lead to privacy concerns.

\*\*Bug 4: Job Application Error\*\*

- \*\*Title\*\*: Unable to Apply for Jobs

- \*\*Reproducible Steps\*\*:

1. Search for a job.

2. Click "Apply Now" on a job listing.

3. Observe that the application does not go through.

- \*\*Attachments\*\*: Error message screenshot.

- \*\*Affected Devices\*\*: iPhone 12, Google Pixel 5

- \*\*Network\*\*: Wi-Fi

- \*\*Severity\*\*: Critical

- \*\*Priority\*\*: High

- \*\*Impact\*\*: Users cannot apply for job opportunities, potentially missing out on career advancements.

\*\*Bug 5: Blank Feed Issue\*\*

- \*\*Title\*\*: Blank Feed After Scrolling

- \*\*Reproducible Steps\*\*:

1. Open LinkedIn mobile app.

2. Scroll through the feed.

3. Observe that the feed becomes blank after a few scrolls.

- \*\*Attachments\*\*: Screenshot of the error

- \*\*Affected Devices\*\*: Various Android devices

- \*\*Network\*\*: Cellular

- \*\*Severity\*\*: Major

- \*\*Priority\*\*: Medium

- \*\*Impact\*\*: Users cannot access content and updates, hindering engagement.

\*\*Bug 6: Unwanted Notifications\*\*

- \*\*Title\*\*: Unwanted Notifications for Declined Connection Requests

- \*\*Reproducible Steps\*\*:

1. Send a connection request.

2. Have the recipient decline the request.

3. Continue to receive notifications about the declined request.

- \*\*Attachments\*\*: None

- \*\*Affected Devices\*\*: Various iOS devices

- \*\*Network\*\*: Various networks

- \*\*Severity\*\*: Minor

- \*\*Priority\*\*: Low

- \*\*Impact\*\*: Annoying and irrelevant notifications may lead to user frustration.

2. These bug reports clearly describe each issue, steps to reproduce, device and network information, severity, priority, and impact, making it easier for the development team to identify and resolve the problems efficiently.

\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\

Third Part: Automated tests for Facebook registration and login scenarios

1. Attached the code.
2. The code is written using Selenium based on Java and page object model design patterns.

It’s divided into 3 pages, the first is the home page where the user can press ‘create account’, the second is the register page where the user can enter the details of the account and the third is the Facebook login page (this is not provided).

There is also a testing framework to divide the code into:

1. @BeforeClass: where the setup before the test is there and functions like setting the chrome driver, opening the Facebook homepage and maximizing the window.
2. @Test: where there are the tests of creating the account.
3. @AfterClass: where the setup after the test is there and has the function of closing the browser.

\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\

Fourth Part

1. A list of test cases Proposed for Automation:
2. Verify that the /products endpoint returns a valid response (status code 200).
3. Verify that the /stores endpoint returns a valid response (status code 200).
4. Verify that the /services endpoint returns a valid response (status code 200).
5. Verify that the /categories endpoint returns a valid response (status code 200).
6. Create a product, get it by the id returned from the post creation, and then use this id to update the product and in final delete the product by this id.
7. Create a store, get it by the id returned from the post creation, and then use this id to update the store and in final delete the store by this id.
8. Create a service, get it by the id returned from the post creation, and then use this id to update the service and in final delete the service by this id.
9. Create a category, get it by the id returned from the post creation, and then use this id to update the category and in the final delete the category by this id.
10. Attached the collection.
11. The automation is based on workflows of creating, getting, updating, and deleting every one of products, stores, services, and categories and this is divided into folders.

Each one of these folders has a test script to validate the pass response in each of their requests. Also, for every folder there is a script to take the ID from the post creation and use it in upcoming requests of (get, patch, and delete).