**Exam Project Softwareudvikling 2. Semester**

**Test Plan**

**Introduction**

This test plan outlines the testing approach for MTOGO, a food delivery application. It describes the scope, testing approach, schedule and tools needed to ensure quality, functionality and performance.

**Scope**

Features to be Tested:

* User registration, login, and account management.
* Menu browsing, item selection, and order placement.
* Order status notifications (app).
* Delivery and feedback systems, including customer and delivery agent ratings.
* Fees and bonus calculations for restaurants and delivery agents.
* Management dashboard with order statistics and performance metrics.

Features Out of Scope:

* Payment gateway integration.
* Order status notifications (SMS).

**Test Items**

Account: Account creation and login.

Search bar: Searching for restaurants and menu display.

Ordering: Selecting items from menu, placing order and restaurant notification.

Delivery: Selecting delivery driver and updating status.

Feedback: Sending feedback requests to customers and receiving a response.

Restaurant: Restaurant fee and editing menu.

Admin: Statistics on order metrics, delivery performance and delivery driver bonus.

**Testing Approach**

Unit Testing: TDD will be used for all individual components and functions in all features mentioned in the scope.

Integration Tests: Will be implemented for all services mentioned in the scope.

System Tests: End-to-end testing will be done to test customers’ experience from login to order placement, delivery and feedback.

Acceptance Tests: Will be implemented based on acceptance criteria from the most important user stories.

Load Tests: This will be done using JMeter once the application is finished to test that it fulfills the requirement from the test strategy.

Mutation Test: Will be done once the application is finished to ensure that all tests are working as intended.

**Test Data**

Types of Data:

* User Accounts: Test accounts with varying profiles for ordering.
* Restaurants and Menus: Sample data to simulate restaurants, menu items and pricing.
* Orders: Test data to validate different order sizes and bonus calculations.

Data Privacy Considerations:

* Anonymize or use synthetic data to ensure compliance with data protection standards.
* Automated data reset scripts to maintain a consistent state in the test environment.

**Test Schedule**

Test planning: 05-11-2024 to 10-11-2024.

Unit Testing: Created at the same time as the service.

Integration Testing: Created at the same time as the service.

Performance Testing: After the entire application has been created.

**Test Automation**

Unit tests and integration tests will be automated using GitHub actions, so any code pushed to GitHub will have to pass all tests.

**Test Tools**

Moq to simulate the applications behavior so that tests can be performed in the same environment.

Use XUnit for writing and running unit tests, integration tests and acceptance tests.

Postman for testing the API.

CI/CD pipeline for testing the application with GitHub Actions.

Load tests will be done using JMeter to simulate a large number of users.

**Code coverage**

Code coverage will be documented using Coverlet to ensure the coverage requirements in the testing strategy.