Prepared by:

Nicholas Chu

Nicholas Kuhaneck

Benjamin Mizell

10/21/2022

OBJECTIVES AND TASKS

Objectives

The objectives of this project are to develop and implement a banking application that allows a user to view their transaction history on bank accounts as well as apply for loans. To accomplish this will be managing the project using Jira. Nicholas Chu will focus on developing the testing at both the automated end to end and the unit level. Benjamin Mizell will be developing the front-end HTML and CSS for the application. Nicholas Kuhaneck will be implementing the database schema for this project as well developing the back-end functionality to include the API testing with Postman.

Tasks

* User can login with correct credentials
* User can not login with incorrect credentials
* User can check their account transactions
* User can navigate with navigation links
* User can submit a loan application
* User can view a loan application

SCOPE

General

All of the methods in the Service layer will be tested with unit tests. API testing will occur using Postman using basic status code tests. Selenium will be used to test the end to end product.

Tactics

We will monitor progress on all taskings by doing daily stand-ups. We will communicate as a team using Discord and Microsoft teams to coordinate the order and timely completion of all testing.

TESTING STRATEGY

Using a TDD/BDD strategy as our approach to development and testing we will start by outlining the features using feature files developed with Cucumber and Gherkin. Once the feature files are written we will move on to outlining the unit tests for the applications business requirements. We will then develop the back-end using the outlined unit testing and feature files so we can then implement the Postman API tests.

Unit Testing

Definition:

Unit testing will be implemented using Mockito in which we will mock the methods in the service layer in order to mock and test each method individually.

Participants:

Nicholas Chu.

Methodology:

Nicholas Chu. will be developing the unit tests based on the outlines previously developed by the whole team.

System and Integration Testing

Definition:

System testing will be written using Selenium, Cucumber, Gherkin and Junit. Integration testing will occur with Postman.

Participants:

Nicholas Chu will develop System testing.

Nicholas Kuhaneck will develop the API testing

Methodology:

Integration testing will be implemented after the API has been implemented.

System testing feature files will be written before code development. The completion of the automated testing will occur after the front-end has been developed.

User Acceptance Testing

Definition:

User Acceptance Testing will occur as the front-end is being developed. This will be accomplished with manual testing.

Participants:

Benjamin Mizell.

Methodology:

Manual testing

HARDWARE REQUIREMENTS

A computer that has internet access.

ENVIRONMENT REQUIREMENTS

IntelliJ IDE, Maven, Java, Windows OS, Chrome browser and drive, Cucumber, Gherkin, Junit, Jira, and AWS RDS.

Where will you acquire these resources?

The internet.

TEST SCHEDULE

Given that this is a two-week timeline the testing schedule will be divided into two sprints, each ending on the Wednesday of the following week.

CONTROL PROCEDURES

Problem Reporting

When a defect is detected, it will be logged into the defect report and assigned to a developer.

Change Requests

All of the changes will be unanimously agreed to by the full team.

FEATURES TO BE TESTED

The user can login with correct credentials.

The user cannot login with incorrect credentials.

The user can check their checking and savings account balances.

The user can see their account transaction histories.

The user can apply for a loan.

The user can view their loan applications.

The user can conduct transfers between accounts.

RESOURCES/ROLES & RESPONSIBILITIES

All testing design will be contributed to by the entire 3 person team on this project.

SCHEDULES

Deliverables

⦁ Test Plan

⦁ Test Cases

⦁ Defect report

⦁ Requirements Traceability Matrix