The University of Nevada, Reno

Department of Computer Science and Engineering

CS 426: Senior Projects in Computer Science

Project Part #3: Acceptance Criteria and Testing Strategy and Plan

Team 5B

Joanna Lopez and Araam Zaremenhjardi

Instructor: David Feil-Seifer and Devin Lee

Advisor: Vihn Le

March 14th, 2022

Table of Contents

1.0 Abstract

The current dormitory maintenance requests system at the University of Nevada, Reno (UNR) provides a system that is flawed and susceptible to complications. Some issues the current system faces are paper trails, leading to lost reports, a lack of maintenance request progress tracking for the students, and deters students from submitting issues that occur within the property. Moreover, Optimum Property Fix (OPF) aims to revamp UNR's current system through a new and improved maintenance request system, accessible to dormitory students as well as facilities and service members. OPF's functionality fully eliminates the paper-based system and replaces it with a web application that retains all maintenance records in one place.

2.0 Project Updates and Changes

The progress made towards the development of OPF would be the implementation of application features with further redesigns slated for frontend and database. Implementation of application features is for user authentication, account creation, additional web pages, and the development of a communication pipeline to add student user tickets to the database. There are more plans developed to incorporate major changes to the interface design and database technology.

Project Updates

Implementation of application features is session functionality, giving users the ability to create a new account, create form pages for admin users, and build a basic pipeline for communication between the frontend and backend subsystems of OPF. Session functionality allows for users to sign in and sign out of OPF with the frontend having a JSON web token used by the backend to authenticate access to database information. Account creation was implemented within OPF allowing users to have a portal to create their own accounts with user information and credentials being saved to a database. Additional web pages added are for admin users being forms for adding new announcements and frequently asked questions for student users. The form webpages do not communicate to the backend and are currently static pages in OPF. Changes done to one of the application's ticket functionality are the further development of a communication pipeline to the database. Student users are now able to create tickets that are saved to the backend with their accounts tied to the ticket.

Future Project Changes

Further plans to change OPF is the redevelopment of the frontend user interface design and switching of database technology. The frontend user interface design will be further adapted to work with mobile screens. Elements of the interface to primarily be focused on for adaptation are form elements and redesign of the application's side navigation to a simplified layout. These

changes are proposed to further make the application accessible to students which use mobile devices as their primary devices. Database technology will be changed to MySQL with dbForge as a helper tool to interface with the MySQL database. The reason for this major change of database technology is to further refine the schema of the database and its associated models. Using external tools for schema definition of the database would allow finer database models with sensible relationships and thus allow code within the backend to primarily for querying rather than database schema definitions and querying.

3.0 User Stories and Acceptance Criteria

User stories describe a situation that explains how a system or systems might be used and the interactions with the systems that might take place. Acceptance tests are tests written by a collaboration of the stakeholders and the programmers in order to define when a requirement is done. The customer tests to decide if it is adequate to meet their needs and so should be accepted by a supplier. The following describes OPF's user stories and their associated acceptance criteria.

1. As a *student user*, I expect to *log in & log out* of the Optimum Property Fix website.
    a. Acceptance criteria: Given that I am a student user when I go to enter a username and password and click on sign up, then I am successfully registered and able to log in with my chosen credentials.
    b. Acceptance criteria: Given that I am a student user, when I am logged into OPF I can click on the logout button, then successfully log out of OPF with my credentials cleared.
2. As an *administrative user*, I expect to *log in & log out* of the Optimum Property Fix website.
    a. Acceptance criteria: Given that I am an admin user when I go to enter a username and password and click on sign up, then I am successfully registered and able to log in with my chosen credentials.
    b. Acceptance criteria: Given that I am an admin user, when I am logged into OPF I can click on the logout button, then successfully log out of OPF with my credentials cleared.
3. As a *student user*, I expect to *create and submit* a maintenance ticket with the following: title, description, severity, location, building, unit number, and additional notes to fix an issue in my dormitory.
    a. Acceptance criteria: Given that I am a student user when I create a maintenance ticket, then that ticket is successfully saved on submit.
4. As a *student user*, I expect to *view* the recently created maintenance ticket to ensure its creation.
    a. Acceptance criteria: Given that I am a student user when I create a ticket then I can successfully see the ticket with the information I supplied.

5. As an *administrative user*, I expect to *view* the maintenance tickets created by a user with a status of ‘received’.
    a. Acceptance criteria: Given that I am an administrator user, I can successfully see a ticket registered by student users with status as ‘received’ in a table.
6. As a *student user*, I expect to *view* the tickets with the information I supplied such as title, status, identification number, severity, date requested, and location in a table for organization and transparency.
    a. Acceptance criteria: Given that I am a student user, I can view the information for a ticket in a table.
7. As an *administrative user*, I expect to *view* tickets with the information supplied by the user such as title, status, identification number, severity, date requested, and location in a table for organization and transparency.
    a. Acceptance criteria: Given that I am an administrator user, I can view all tickets supplied by all student users in a table with the ticket data.
8. As a *student user,* I expect to *search and filter* for a specific ticket I've created and view that result in the table.
    a. Acceptance criteria: Given that I am a student user, I can search for a specific ticket in the table view with a textbox and be able to filter by ascending or descending order.
9. As an *administrative user,* I expect to *search and filter* for a specific ticket created by any user and view that result in the table.
    a. Acceptance criteria: Given that I am an administrative user, I can search for a specific ticket in the table view with a textbox and be able to filter through the data different actions displayed on the table.
10. As a *student user,* I expect to *submit* feedback such as rating the overall Optimum Property Fix experience, if we were satisfied, and allowing additional comments/concerns to communicate with the facilities and services team.
    a. Given that I am a student user, I can give feedback to the maintenance and services team through a survey about their completed job in a form.
11. As a *student user,* I expect to *view* my maintenance ticket and supply feedback directly from the completed ticket.
    a. Acceptance criteria: Given that I am a student user, I can view the completed maintenance ticket and be able to fill out a survey to give feedback on the facilities and services completed.
12. As an *administrative user,* I expect to *view* feedback from users after completing the maintenance request.
    a. Acceptance criteria: Given that I am an administrative user, I can view the feedback supplied by student users on the feedback page via an analytics graph.

4.0 Testing Workflow

The testing workflow will overview the happy and unhappy paths of the basic functionality of OPF. The basic functionality of OPF consists of frontend to backend communication, user authentication, and database reading and writing operations. Frontend to backend communication entails OPF's features such as pipelines to create, modify, and update tickets, appointments, and other data. User authentication features of OPF entail the ability for users to log in and log out of the application with the frontend and backend able to properly respond to events that require authentication such as changes to accounting settings and database interactions. Reading and writing operations to the database within the application must ensure data inputs are correctly validated such as ensuring data types are correct and strings may not contain illegal data.

Happy Workflow Paths

Happy workflow paths in OPF are the sequence of events that correctly respond to user actions without any errors within the system. The steps of workflow steps are overviewed with the proper response to user action being denoted as validation criteria. The ideal case scenarios overviewed for happy workflow paths are the submissions of new tickets, users signing into OPF, and student users viewing ticket metadata.

Table 1: Submitting New Ticket

Happy Workflow Path Steps	Validation Criteria
1. Student users login into the application. 2. Click the "Maintenance Requests" button from the navigation bar. 3. Click the "Create New Ticket" button from the page's sub-navigation. 4. Fill in the required forms with completed information. 5. Press the "Submit Ticket" button. 6. See the newly created ticket in the previous tickets table as the most recent ticket.	The validation criteria for the submission of a new ticket is to view the ticket listed within the database and listed in the user-facing tickets table.

Table 2: User Signing into OPF

Happy Workflow Path Steps	Validation Criteria
1. User while logged into OPF is at any screen in the application. 2. User clicks the “logout” button in the navigation of the application. 3. User is taken to OPF’s login screen.	The validation criteria for the user logging out of a current session is the removal of the stored access token in the user’s browser and backend successfully reporting the current JSON web token removed.

Table 3: Student Sees Single Ticket Metadata

Happy Workflow Path Steps	Validation Criteria
1. Student users login into the ticket. 2. Click the “Maintenance Requests" button from the navigation bar. 3. Views a table of all current and previous tickets. 4. The student user clicks the title of any ticket in the table. 5. The student user views the individual ticket view. 6. The student user views the following metadata: description, severity, location, building, unit number, and additional notes.	The validation criteria for student users viewing ticket metadata is to see the display of the ticket view screen. Confirmation of the details of the ticket data is confirmed through the viewing of the database record using external tools such as dbForge.

Unhappy Workflow Paths

Unhappy workflow paths in OPF are the sequence of events that attempt to break the system in unintentional ways. The steps to reproduce situations that may cause the system to function abnormally are overviewed with the validation criteria being how well the system responds to these errors. The unforeseen scenarios for unhappy workflow paths are when data related to a ticket does not delete upon deletion of a ticket, when unauthorized users attempt to access restricted systems of OPF, and when database systems may fail unknowingly to other systems.

Table 4: Appointment Persisting after Ticket Deletion

Unhappy Workflow Path Steps	Validation Criteria
1. The admin clicks the “Maintenance Requests" button from the navigation bar. 2. The admin user clicks the title of any ticket in the table. 3. The admin user views the individual ticket view. 4. The admin user clicks the “delete” button from the ticket options. 5. The admin user clicks the “Appointments” button from the navigation bar. 6. The admin user views the table of appointments. 7. The admin user sees the appointment for the deleted ticket persisting after ticket deletion.	The validation criteria for the existence of the error is by viewing both the existence of the ticket and related appointment in the database using dbForge. Taking note of the related appointment, following the workflow path the related appointment should be deleted due to ticket deletion. The persistence of the appointment after deletion indicates a system issue has occurred. The issue in the system should be reported as a server error when the user attempts to access the ticket metadata from the appointments page.

Table 5: Student Attempts to Enter Administrator Account by Modifying Frontend URL

Unhappy Workflow Path Steps	Validation Criteria
1. Student user while logged into OPF is at any screen in the application. 2. The student user enters an admin user-only URL into the browser. 3. The student user views the admin user experience.	The validation criteria for the system error is the access of a restricted page. Student users should not be able to access admin user pages and vice versa. The error is to be handled by automatically logging the user out and taking them to the login screen.

Table 6: When The Backend System Communicates to offline Database Server

Unhappy Workflow Path Steps	Validation Criteria
1. Any user (Student & Admin) while logged into OPF. 2. User attempts to do any read and write operation in OPF (seeing their tickets, creating tickets, etc.) 3. OPF attempts to contact the backend system to receive data for REST API calls. 4. Users are notified by a connection issue with a “connection issue” error as floating dialog.	The validation criteria for the system error is the failure of the backend to fulfill a request from the frontend. The error is to be handled gracefully by the frontend by allowing some other non-server-related functionality to continue functioning while for other operations the user is notified of a connection issue.

5.0 Testing Strategy

The benefits of testing our codebase include confidence in our product, OPF, to stakeholders such as dormitory users and administrators to ensure reliable software. The benefit of testing allows developers to find and fix bugs early in the lifecycle and address them early on. Additionally, the code quality can be enhanced by refactoring without the fear of breaking the program and simultaneously enhancing the quality of the codebase.

Team 05B will be using Jest, a JavaScript testing framework, to test our front-end code. Jest is fast, can perform snapshot testing, and provides everything out of the box. Due to time constraints, we won’t have a high percentage of code coverage. We will be implementing at least one of the following: acceptance tests, user tests, unit testing, and integration testing.

Table 7: The overall responsibility of Testing for the OPF web application.

Tests	Responsibility
Acceptance Tests & User Tests	Araam
Unit Tests & Integration Tests	Joanna

Table 8: The table outlines the user story, who it is developed by, who will test it, and the date of when it will be tested.

User Story #	Developed By	Testing Role	Date
1	Araam	Joanna	April 19th, 2022

2	Araam	Joanna	April 19th, 2022
3	Araam	Joanna	April 19th, 2022
4	Araam	Joanna	April 19th, 2022
5	Araam	Joanna	April 19th, 2022
6	Araam	Joanna	April 19th, 2022
7	Joanna	Araam	April 19th, 2022
8	Joanna	Araam	April 19th, 2022
9	Joanna	Araam	April 19th, 2022
10	Joanna	Araam	April 19th, 2022
11	Joanna	Araam	April 19th, 2022
12	Joanna	Araam	April 19th, 2022

Defects

Defects in software engineering are a result of expected and divergent results of the software application such as an error or a bug. The irregularities are defects from the project specification that are caused by the developer in development. Defects include logic, syntax, interface, and performance defects. Team 05B will be responsible for reporting defects through identification, description, detailed steps for replication, date of bug reported, the severity of the defect, and the priority. Figure 1, outlines how OPF will handle defects and how they will be reported and distributed amongst the team.

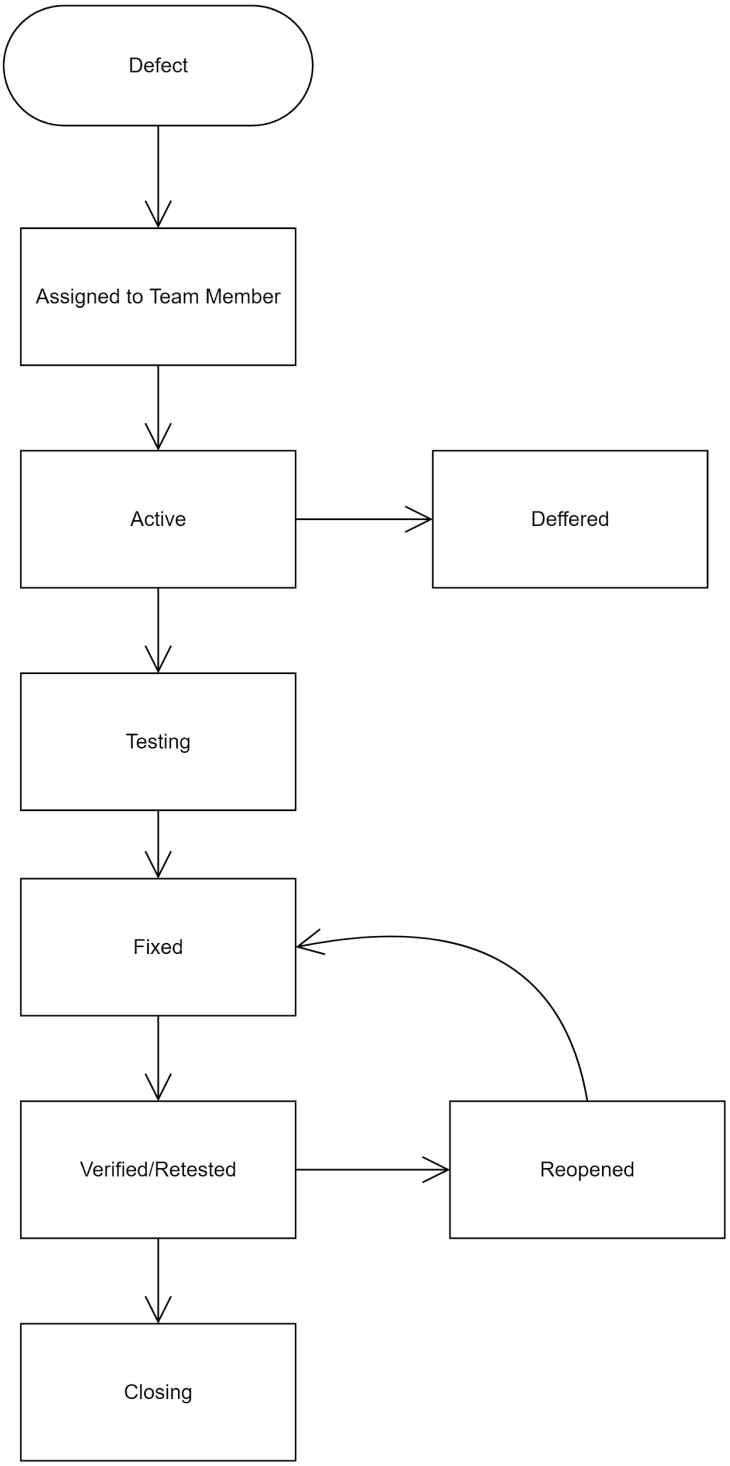

Figure 1: The plan to handle defects and how they will be reported and distributed amongst the team.

Table 9: The OPF testing plan for the acceptance criteria and workflow items.

Test Plan								
No.	**Type**	**Target File or Screen**	**Name**	**Purpose**	**Test Data or Situation**	**Expected Result**	**Expected Result**	**Outcome and Actions Required**
1	Login	Login Screen	Credential Validation	Test that it logins in based on entered user credentials.	Enter the credentials used to register an account.	Directed to correct account with their personalized settings.	1. Expected	The login renders are based on user credentials. The user is directed to the respective page.
2	Logout	Logout Screen	Credential Deactivation	Tests that cleared the access token alongside the backend.	The user is able to log out of their account upon selecting the sign-out button.	Directed out of OPF account and directed back to OPF home page.	1. Expected	The user is logged out of OPF and will need to re-enter credentials.
3	Ticket View	Ticket Screen	Rendering of the ticket table.	Test the ticket table and renders correctly.	The user will be able to view the ticket table when selecting the ticket screen.	The table will render 10 rows and 7 columns with alternating colors of white and light grey. The columns will contain title, description, severity, location, building, unit number, and additional notes.	1. Expected 2. Expected	The ticket table renders the information as expected. No actions are required.
4	Ticket Filtering	Ticket Screen	Filtering Data Tables	Test that ticketing data is filtered and rendered on	Select the data to filter in ascending or descending	The data will be sorted based on what the	1. Expected	The ticket table renders the information

				the table.	order.	user selects and reflected on the ticket table.		as expected.
5	Feedback Submission	Feedback Screen	Rendering of feedback data.	Test that feedback user data is rendered on the administrative feedback screen via a graph and/or table.	Select the feedback of the maintenance ticket in either.		1. Expected	The feedback graph and/or table will render the information as expected.

6.0 Contributions of Team Members

Araam Zaremehrjardi's Contribution

Araam Zaremehrjardi's total time worked on the revised specification and design totals 5 hours. Contributions on the revised specification and design were on project updates and future project changes and testing workflow.

Joanna Lopez's Contribution

Joanna Lopez's total time worked on the revised specification and design totals 5 hours including the user stories and acceptance criteria, testing strategy, and editing the documentation with Araam.