Test Scenarios for Nextay

Introduction

This document lists the test scenarios for various functionalities of the Nextay hotel management system. Each scenario validates specific aspects of the system to ensure comprehensive testing and reliability.

Test Scenarios

Login Functionality

1. Verify that each role (Manager, Staff, Finance) can log in with valid credentials.
2. Check login behavior with incorrect credentials (e.g., wrong password).
3. Ensure fields for username and password are mandatory.
4. Confirm password input is masked.
5. Validate that each role is navigated to their respective dashboard upon login.
6. Verify new password functionality if the user forgets the password and tries to reset it.
7. Verify that all required fields (Username, Password) are enforced.
8. Ensure character limits are enforced for fields like Username (20 character limit) and Password (6 character limit).

Add Customer Feedback

9. Verify that all required fields (Customer Name, Rating, and Feedback) are enforced.
10. Verify that feedback can be submitted successfully when all fields are filled correctly.
11. Confirm that appropriate error messages are shown if mandatory fields are left empty.
12. Ensure character limits are enforced for fields like Customer Name (15 character limit).
13. Check that a confirmation message is displayed after successful submission.
14. Ensure the submitted feedback is immediately visible in the feedback list.
15. Verify that only authorized roles (e.g., Staff) can access the Add Feedback form.
16. Ensure that error messages do not expose system details that could aid attackers.
17. Confirm that excessive input in text fields (Customer Name) is restricted to avoid buffer overflow attacks.
18. Verify that submitted feedback appears in the feedback list promptly.
19. Check if the database efficiently handles large numbers of feedback entries.
20. Confirm that all fields (Customer Name, Feedback, Type) are clearly labeled and instructions are provided where needed.
21. Ensure error messages are understandable and guide users on correcting issues.

View Feedback

22. Verify that Manager can log in successfully to view customer feedback.

23. Check that correct details of feedback are fetched from the database based on the correct customer ID.

24. Verify all feedbacks of a customer display on the screen.

25. Check that customer feedback information remains accurate after reloading the page.

26. Manager can view all feedback from the same customer on different dates if feedback was provided at different times.

27. Ensure that all labels are clearly shown with corresponding information like Feedback Type, Customer Name.

View Attendance

28. Verify that Manager can log in successfully to view the attendance of employees (Staff, Finance).

29. Verify attendance of all employees is displayed on the screen.

30. Manager can view all attendance records datewise or monthwise.

31. Absent, Present, Late status of attendance must be displayed in the UI with specific colors for easy comprehension.

32. System prevents unauthorized users from viewing attendance records.

33. Manager views real-time attendance for employees currently checked in.

34. System highlights attendance records with missing check-outs.

Mark Attendance

35. Ensure that authorized employees access the mark attendance feature.

36. Verify Employee checks in and checks out at the scheduled times, and the system records attendance correctly.

37. System also marks attendance status for Late Entry employees.

38. System prevents multiple check-ins within the same shift to avoid duplicate records.

39. System restricts check-out attempts if a prior check-in is missing, prompting the user to check in first.

40. Only authorized roles (e.g., managers) can edit attendance records, while employees (Finance) can only view their own.

41. Ensure that all data related to attendance is accurately stored in the database.

Add Employee Functionality

42. Add employee with all required fields correctly filled.
43. Attempt to add an employee with missing mandatory fields.
44. Add employee with invalid data in fields (e.g., invalid email).
45. Attempt to add a duplicate employee profile.
46. Verify system logging and audit trail for added employees.
47. Add employee with minimum required information.
48. Test system behavior when adding multiple employees in rapid succession.
49. Validate max character limits on the employee name field.

Delete Employee Functionality

50. Delete an existing employee.
51. Attempt to delete a non-existent employee.
52. Confirm the deletion process includes an additional verification step.
53. Verify system logging for deleted employees.
54. Check system response for deletion without internet connection.

Update Employee Functionality

55. Update employee profile with valid data.
56. Attempt to update with invalid data.
57. Attempt to update a non-existent employee.
58. Verify update reflects immediately in the employee list.
59. Verify system logging for updated employees.
60. Validate character limits for each editable field.
61. Attempt to update while another user edits the same profile.

Manage Shift for Employees

62. View employees scheduled for the current shift.
63. Validate the employee shift change after updating.

Pay Salary

64. Verify Finance Manager can log in and access the salary management page.

65. Check that employee details with ”Pay” and ”Bonus” options are displayed.

66. Confirm Finance Manager can select a date to view or pay salaries.

67. Validate that a bonus can be added and is displayed as ”Bonus Added.”

68. Ensure salary is marked as ”Paid” without a bonus and logged in expenses.

69. Confirm salary with bonus is marked as ”Paid” and recorded in expenses.

70. Verify all paid salaries on the selected date are summed in the expense table.

71. Ensure ”isPaid” resets to ”unpaid” for all employees in the new month.

72. Check that ”Pay” cannot be clicked for already paid salaries.

73. Validate that an error is shown for invalid bonus entries.

74. Confirm Finance Manager can navigate dates to view historical payments accurately.

75. Verify ”Paid” or ”Pay” status displays correctly on reload.

View Salary for Employee

76. Verify that employees can log in successfully to access their salary details.

77. Check that the correct salary details are fetched from the database based on the logged-in employee’s ID.

78. Ensure that the salary amount and designation are displayed correctly for the employee.

79. Verify that ”Bonus Added” is shown if a bonus has been added; otherwise, display ”null.”

80. Confirm that the status displays as ”Paid” when the salary has been paid for the month.

81. Ensure the salary status resets to ”unpaid” at the beginning of each month.

82. Check that the salary information and status remain accurate upon reloading the page.

83. Validate that an appropriate error message is shown if salary data is missing.

84. Verify that employees can view previous months’ salary details if allowed.

85. Ensure employees cannot view other employees’ salary details.

View Salary for Manager

86. Verify that the manager can log in successfully to access the salary view functionality.
87. Ensure that the manager can select a specific date to view employees' salary details for that month.
88. Confirm that the manager can view the salary details of all employees for the selected month.
89. Check that each employee's salary and designation are displayed correctly for the selected date.
90. Ensure that invalid or future dates cannot be selected when viewing salary data.
91. Confirm that an appropriate message is shown if salary data for the selected month is unavailable.
92. Ensure that only managers can access the full salary view of all employees.
93. Validate that the manager can access salary details for previous months.

Employee's Promotion

94. Verify that the manager can access the Employee Promotion page.
95. Ensure that each employee's name, designation, and salary are displayed correctly.
96. Verify that clicking the "Promote" button opens a new window for changing the employee's designation and salary.
97. Ensure that the new window displays the current designation and allows the manager to update it.
98. Confirm that clicking the "Change Salary" button opens a popup for salary modification.
99. Verify that the popup displays the current salary and provides an input field for entering the new salary.
100. Check that the salary input field only accepts valid numerical values.
101. Ensure that clicking the "Save" button in the popup successfully updates the employee's salary in the system.
102. Verify that clicking the "Cancel" button in the popup closes it without making changes.
103. Ensure that the employee's updated designation or salary is saved in the system database.
104. Verify that the Employee Promotion page reflects the updated designation and salary after a successful promotion.

105. Confirm that only users with manager roles can access the promotion functionality.

106. Ensure that the Employee Promotion page loads quickly and without lag when displaying multiple employees.

107. Verify that the changed salary is greater than 20000 and consistent with the designation.

View Promotion History

108. Verify that the "Promotion History" page loads correctly with a list of employees who were promoted.

109. Ensure that each employee's name, designation, and salary are displayed accurately on the Promotion History page.

110. Verify that clicking the "View Promotion History" button opens a new window with promotion history details for the selected employee.

111. Ensure that the promotion history window displays fields for the previous designation, promotion date, and new salary.

112. Date Format Validation: Ensure that the promotion date is displayed in a consistent and readable format (e.g., MM/DD/YYYY).

113. Empty History Handling: Verify that a message, such as "No promotion history available," is displayed if an employee has no previous promotions.

114. Authorization Check: Verify that only authorized users (e.g., manager) can access and view promotion histories.

115. Responsive Design: Verify that the promotion history window adapts properly on different screen sizes and resolutions.

Add Room

116. Valid Room Addition: Verify that a room is successfully added when all required fields (room type, amenities, price, etc.) are filled correctly.

117. Missing Required Fields: Check that the system shows an error if any required fields (e.g., room type or price) are left blank.

118. Duplicate Room Entry: Verify that the system prevents adding a room with the same room number as an existing room.

119. Invalid Price Format: Test that the system validates and rejects prices entered in an invalid format (e.g., negative price or non-numeric value).

120. Maximum Room Capacity: Test the system's behavior if the maximum number of rooms allowed in the hotel is reached.

121. Add Room with Amenities: Verify that all selected amenities are correctly saved and displayed after adding a new room.

122. Large Text Input for Description: Test adding a room with a large text input for the description field to check for input length restrictions or truncation.

123. Boundary Price Value: Test adding a room with a price value of 0 or a high boundary price to ensure the system handles boundary conditions.

Delete Room

124. Valid Room Deletion: Verify that a room is deleted successfully when the user selects an existing room.

125. Non-Existent Room Deletion: Test the system's response when trying to delete a room that does not exist.

126. Delete Room with Active Booking: Check if the system prevents deleting a room that currently has active bookings.

127. Delete Room Without Confirmation: Verify that a room is not deleted if the user does not confirm the delete action.

128. Bulk Room Deletion: Test if multiple rooms can be selected and deleted in one action and confirm that all selected rooms are deleted.

129. Room Deletion by Unauthorized User: Verify that users without delete permissions cannot delete rooms.

Update Room

130. Valid Room Update: Test that a room's information can be updated successfully when all fields are modified and submitted.

131. Update Non-Existent Room: Verify that the system shows an error if attempting to update a room that no longer exists in the database.

132. Update with Invalid Price: Check that the system prevents updating the room if an invalid price is entered.

133. Partial Room Update: Test updating only some fields (e.g., updating amenities only) to verify partial updates work.

134. Update Room Status (Available/Unavailable): Verify that the system allows toggling a room's status between available and unavailable.

135. Update with Special Characters in Fields: Ensure that special characters in text fields are handled correctly during updates.

136. Exceeding Room Description Character Limit: Verify that the system restricts room description text length to the specified limit and provides feedback.

Check-In

137. Valid Check-In with Customer Data: Verify that check-in is successful when all required customer details (name, ID, contact) are entered correctly.

138. Check-In with Missing Customer Data: Test that the system prevents check-in if mandatory customer data is missing.

139. Check-In for Unavailable Room: Verify that check-in is not allowed for rooms marked as unavailable or under maintenance.

140. Check-In with Invalid Check-In Date: Test that the system does not allow check-in for dates in the past or dates that conflict with current bookings.

141. Automatic Room Status Update on Check-In: Verify that the room's status automatically updates to "Occupied" upon successful check-in.

142. Check-In with Invalid Customer ID: Test that the system validates the customer's ID format and prevents check-in if it is incorrect.

143. Check-In by Unauthorized User: Ensure only authorized users (e.g., front desk staff) can perform check-ins.

Check-Out

144. Valid Check-Out with Bill Calculation: Verify that check-out is successful when bill calculation (based on room rates and stay duration) completes without errors.

145. Check-Out for Unchecked-In Room: Test that the system prevents check-out if no check-in was recorded for the room.

146. Check-Out Without Bill Clearance: Check if the system prevents check-out if the bill is not cleared or marked as unpaid.

147. Room Cleaning Status Update on Check-Out: Verify that the room status is marked for cleaning after a successful check-out.

148. Invalid Check-Out Date: Test if the system handles invalid check-out dates (e.g., before check-in date) correctly.

149. Check-Out by Unauthorized User: Ensure only authorized users (e.g., front desk staff) can perform check-out.

150. Automatic Room Status Update on Check-Out: Verify that the room's status updates to "Available" after successful check-out.

151. Bill Calculation: Test that the system calculates the final bill accurately.

Additional Edge Cases for Room Check-In/Check-Out

152. Simultaneous Check-In and Check-Out: Test that the system handles simultaneous check-in and check-out operations on multiple rooms without errors.

153. System Response for Network Failure During Room Operations: Check how the system handles a network failure during room add/update/check-in/check-out operations.

Performance Testing Test Scenarios

154. Check if the page load time is within the acceptable range.

155. Check if the page loads on slow connections.

156. Check the response time for any action under light, normal, moderate, and heavy load conditions.

157. Check the performance of database stored procedures and triggers.

158. Check for stress testing of the application.

159. Check for load testing of the application.

Database Testing Test Scenarios

160. Check values for columns that are not accepting null values.

161. Check for data integrity. Data should be stored in single or multiple tables based on the design.

162. Check if data is committed to the database only when the operation is successfully completed.

163. Check numeric fields with minimum, maximum, and float values.

164. Check numeric fields with negative values (for both acceptance and non-acceptance).

165. Check if the radio button and drop-down list options are saved correctly in the database.

166. Check if all table constraints like Primary key, Foreign key, etc. are implemented correctly.

167. Input field leading and trailing spaces should be truncated before committing data to the database.

Security Testing Test Scenarios

168. Check for SQL injection attacks.

169. Page crash should not reveal application or server info. The error page should be displayed for this.

170. All credentials should be transferred over an encrypted channel.

171. Check for role-based authentication.

172. Passwords and other sensitive fields should be masked while typing.

173. Test unauthorized application access by manipulating variable values in the browser address bar.