PROG8060
TEST STRATEGY

10/11/2023

{TEAM MEMBERS AND STUDENT IDS}

SHIVAM MAHAJAN (8856622)

GURPREET KAUR (8911147)

SULIKA SHARMA (8925953)

POOJA THAKKAR (8799270)

1. Purpose

The major goal of risk-based test strategies is to verify the quality, dependability, and security of the car rental Chatbot, which is a crucial component of our car rental platform. This method is designed to provide a consistent user experience while managing and mitigating associated risks.

Our users may engage, make bookings, and inquire about services using the car rental chatbot, which serves as a significant interface for them. To match user expectations and organizational goals, the chatbot's functionality, accuracy, and security must be maintained.

The following are the overarching goals of this test strategy:

- Identifying and categorizing potential risks related to the chatbot project.

- Establishing testing priorities based on a thorough risk assessment.

- Assigning particular roles and tasks to ensure effective risk management.

- Specifying multiple levels of testing, environmental conditions, and acceptance criteria.

- Implementation of open defect and change management processes.

- Clearly defined exit criteria for each testing level.

- Managing dependencies to ensure they do not impede progress.

- Create contingency strategies for risk minimization.

- Obtaining consent and signatures from key stakeholders, indicating preparation for the next phase.

2. Traceability Matrix

A traceability matrix will be rigorously created to map test cases to the chatbot's specific requirements. This guarantees that test coverage extends across several functionalities, ensuring a thorough examination.

S.NO	Requirement/User Story	Test Case(s)	Risk Factors/Priorities
1.	The user has the option of inquiring about rental pricing.	TC001, TC002	Low
2.	A reservation can be made by the user.	TC003, TC004	Medium
3.	The chatbot is integrated with the booking system.	TC005, TC006	High
4.	User information is securely saved.	TC007, TC008	High
5.	Uncertain user queries are handled by the chatbot.	TC009, TC010	Medium
6.	High user load performance	TC011, TC012	Medium
7.	Pricing calculations that are precise	TC013, TC014	High
8.	Compliance with user data privacy	TC015, TC016	High
9.	The user has the ability to submit comments and ratings.	TC017, TC018	Medium
10.	Support in multiple languages	TC019, TC020	Medium
11.	Payment gateway integration	TC021, TC022	High
12.	Managing Reservation Changes	TC023, TC024	Medium
13.	The user can see the rental terms and conditions.	TC025, TC026	Low
14.	Handle password recovery and account recovery	TC027, TC028	Medium
15.	Provide contact information for customer service.	TC029, TC030	Low

3. Risks

3.1 Identification of risks

1. User Request Misinterpretation: Users may enter ambiguous or complex questions, which may result in chatbot misinterpretation.

Explanation: This danger is linked to the chatbot's natural language processing ability. Thorough testing and validation of the chatbot's knowledge of multiple user inputs are required to address this.

2. Booking System Integration Issues: Failures in the integration of the chatbot with the booking system may cause the booking process to be disrupted.

Explanation: The chatbot's integration with the booking system is critical for real-time updates and reservations. To discover and resolve any flaws, rigorous integration testing is required.

3. Risks to Data Security: Because the chatbot processes and keeps sensitive user data, it is a possible target for security breaches.

Explanation: User data security is critical. The focus of security testing will be on finding vulnerabilities and guaranteeing data security through encryption and access restriction.

4. Third-Party Service Unavailability: The chatbot relies on third-party services for location data and information.

Explanation: To address probable service outages, a contingency plan will be developed. This includes putting fallback procedures in place to guarantee the chatbot keeps working.

5. Incorrect Pricing Calculations: The chatbot is responsible for calculating rental costs, and mistakes in this procedure can result in financial inconsistencies.

Explanation: Testing pricing calculations thoroughly is essential for assuring accurate estimates and minimizing potential financial losses for the organization.

3.2 Categorization of Risks

S.NO	Risk	Description	Categorization
1.	User Request Misinterpretation	Users may send ambiguous or complex queries, which may result in misinterpretation by the chatbot.	Medium Risk
2.	Booking System Integration Issues	Failures in chatbot integration with the booking system may cause the booking process to be disrupted.	Medium Risk
3.	Data Security Threats	The chatbot processes and maintains sensitive user data, making it a possible target for security breaches.	High Risk
4.	Third-Party Service Unavailability	For location data and information, the chatbot relies on third-party sources.	Medium Risk
5.	Incorrect Pricing Calculations	The chatbot is responsible for calculating rental costs, and mistakes in this procedure can result in financial differences.	High Risk

4. Testing Priorities

Testing efforts will be prioritized based on risk classification. High-priority areas, such as data security and price computations, will see increased testing resources.

Feature/User Story	Priority	Reason for Priority	Not in Scope	Reason for Exclusion
1. Misinterpretation of User Requests	High	The impact on the user experience is critical.	N/A	N/A
2. Booking System Integration Issues	Medium	The booking procedure may be disrupted.	N/A	N/A
3. Risks to Data Security	High	The security of sensitive user information is critical.	N/A	N/A
4. Third-Party Service Unavailability	Medium	Effect on chatbot functionality.	N/A	N/A

5. Incorrect Pricing Calculations	High	Financial consequences and organizational risk.	N/A	N/A
6. Unexpected Chat Session Termination	Medium	Negative influence on user happiness and experience.	N/A	N/A
7. Peak Usage Performance Degradation	High	Assuring that the chatbot can handle heavy traffic flows.	Changes to the database schema	Technical difficulty and limited resources
8. Mismatch with User Expectations	Medium	Issues with user displeasure and trust.	Language translation in real time	Dependencies on other parties and licensing concerns
9. Mobile Device Compatibility	Medium	Mobile users are being added to the user base.	Integration of voice recognition	a scarcity of appropriate testing tools.
10. Support in multiple languages	High	Meeting the needs of a wide range of user populations.	Regression testing is done automatically.	Time and financial restrictions.
11. Changing Context	Medium	Transitions between themes are smooth.	User education	Outside the scope of the project.
12. Dealing with Emergencies	High	Ensure user security and compliance.	Cross-browser evaluation	This is not a necessary feature.
13. Chatbot Resource Management	Medium	Resource allocation that is efficient.	Error handling in poor connectivity situations	Resources are limited, and there are technical limits.
14. Offline Usability	High	Service continuity in the absence of internet access.	Regional language NLP in depth	Technical difficulty and limited resources
15. Recognizing Emotions	Medium	Improving user interaction and personalization.	Conversion of spoken words to text	This is not a primary function.

5. Risk owner Roles and Responsibilities

The following are some risk owners and their responsibilities in the SDLC for the car rental application project(WheelEye) :

Development team

- Risk owner: Project manager
- Responsibilities: Identify, assess, mitigate, and monitor risks related to the development process, such as schedule delays, budget overruns, and technical challenges.

Customers

- Risk owner: Product manager
- Responsibilities: Identify, assess, mitigate, and monitor risks related to the customer experience, such as usability issues, performance problems, and security vulnerabilities.

Regulatory authorities

- Risk owner: Compliance officer
- Responsibilities: Identify, assess, mitigate, and monitor risks related to regulatory compliance, such as data privacy and security requirements.

Technology partners

- Risk owner: IT manager
- Responsibilities: Identify, assess, mitigate, and monitor risks related to the use of third-party technology, such as compatibility issues and security vulnerabilities.

In addition to these specific risk owners, all team members have a responsibility to identify, assess, and report risks to the project manager.

Here are some examples of risks and how the different team members might be involved in managing them:

Risk: The customer-facing application experiences performance problems during peak season.

- Development team: Identify and fix the performance bottlenecks.
- Product manager: Communicate with customers about the problem and provide updates on the fix.

Risk: The application is not compliant with a new data privacy regulation.

- Compliance officer: Identify the specific requirements of the new regulation and develop a plan to ensure compliance.
- Development team: Implement the necessary changes to the application.
- Product manager: Communicate with customers about the changes to the application and how they will be impacted.

Risk: A third-party software library used in the application contains a security vulnerability.

- IT manager: Assess the severity of the vulnerability and develop a mitigation plan.
- Development team: Apply the necessary security patch to the application.
- Product manager: Communicate with customers about the vulnerability and the steps that have been taken to mitigate it.

6. Test Levels

Unit testing

Unit testing is the lowest level of testing and focuses on testing individual units of code, such as functions and classes. Unit testing is typically performed by the developers themselves.

Integration testing

Integration testing focuses on testing how different units of code work together. Integration testing is typically performed by a team of testers who are independent of the developers.

System testing

System testing focuses on testing the entire system as a whole. System testing typically includes testing the following:

- Functionality: Does the system perform all of its intended functions?
- Performance: Can the system handle the expected load?
- Security: Is the system secure from unauthorized access and attack?
- Usability: Is the system easy to use?

Performance testing

Performance testing focuses on testing the system's ability to handle a certain workload. Performance testing is typically performed by a team of testers who are specialized in performance testing.

Security testing

Security testing focuses on testing the system's security against unauthorized access and attack. Security testing is typically performed by a team of testers who are specialized in security testing.

Regression testing

Regression testing focuses on testing the system after changes have been made to ensure that the changes have not introduced any new bugs. Regression testing is typically performed by a team of testers who are independent of the developers.

User acceptance testing (UAT)

UAT is a type of testing that is performed by the end users of the system to ensure that it meets their needs. UAT is typically performed after the system has been tested by the development team and QA team.

7. Environmental Requirements

The environmental requirements for testing the car rental booking application will depend on the specific types of testing that are performed and the specific tools and services that are used. However, some general environmental requirements include:

- Hardware: The test environment should have the same or similar hardware to the production environment. This will help to ensure that the application performs as expected in the production environment.
- Software: The test environment should have the same or similar software to the production environment. This includes the operating system, database, and web server.
- Data: The test environment should have the same or similar data to the production environment. This data can be used to test the application with realistic data and to identify any potential data quality issues.
- Network: The test environment should have a network connection that is similar to the production environment. This will help to ensure that the application performs as expected in the production environment.

Here are some specific tools and services that can be used for the different test activities associated with the car rental booking project:

Test planning:

- Test management tools: These tools can be used to create, manage, and track test plans. Some popular test management tools include Jira.
- Requirement management tools: These tools can be used to manage and track requirements. Some popular requirement management tools include Jira.

Test development:

- Test case management tools: These tools can be used to create, manage, and track test cases. Some popular test case management tools include Jira.
- Automated testing tools: These tools can be used to automate the execution of test cases. Some popular automated testing tools for web applications include Selenium.

Test execution:

- Automated testing tools: As mentioned above, automated testing tools can be used to execute test cases.
- Manual testing tools: Manual testing tools can be used to execute test cases that cannot be automated. Some popular manual testing tools include Katalon Studio and Postman.

Test reporting:

- Test reporting tools: These tools can be used to generate test reports. Some popular test reporting tools include Jira, TestLink.

8. Acceptance Criteria

The acceptance criteria for a car rental software component or feature will vary depending on the specific component or feature being tested. However, some general acceptance criteria for car rental chatbot include:

- **Functionality:** The component or feature must perform all of its intended functions.
- **Performance:** The component or feature must be able to handle the expected load.
- **Security:** The component or feature must be secure from unauthorized access and attack.
- **Usability:** The component or feature must be easy to use.
- **Localization:** The component or feature must be localized for the target market(s).

9. References

- Software Testing Fundamentals by Rex Black
- The Art of Software Testing by Glenford Myers
- Car Rental Chatbot Requirements Document