TEST PLAN FOR

MR. TASKMASTER: ON-DEMAND HOME SERVICES SYSTEM

ChangeLog

Version	Change Date	By	Description
version number	Date of Change	Name of person who made changes	Description of the changes made
001	10/31/2023	Manvendra Kumar	Initial Draft

1 Introduction

An on-demand home service system is a technological platform that connects service providers with customers seeking various services for their homes. This innovative approach leverages digital technology to streamline and simplify the process of finding, booking, and receiving home services. Whether it's plumbing, electrical work, cleaning, gardening, or other household tasks, an on-demand home service system aims to make these services accessible, convenient, and efficient.
An on-demand home service system is a modern solution that brings together technology and traditional services to create a seamless and efficient experience for both service providers and consumers.

1.1 Scope

Functional Requirements:

- User Registration and Authentication:
  - Users should be able to create accounts with unique identifiers.
  - A secure authentication mechanism to verify the identity of users and service providers.
- Service Provider Registration:
  - Service providers should be able to create profiles, including information about their skills, experience, and services offered.
  - Verification process for service providers to ensure reliability.
- Service Listings:
  - A catalog of available services with detailed descriptions.
  - Filters and search functionality to help users find specific services or service providers.
- Real-Time Scheduling:
  - A calendar or scheduling system allowing users to select preferred dates and times for services.
  - Notifications to confirm or modify scheduled appointments.
- Booking and Confirmation:
  - A booking system that allows users to request services and receive confirmation from service providers.
  - Integration with a payment gateway for securing bookings.
- Payment Processing:
  - Secure payment options for users to pay for services.
  - Invoicing and receipts for both users and service providers.
- Rating and Review System:
  - A system that allows users to rate and provide reviews for services.
  - Aggregation of ratings to display an overall score for each service provider.
- Notifications:
  - Automated notifications for users and service providers regarding booking confirmations, cancellations, and other updates.
  - Push notifications through the mobile app (if applicable).
- GPS Integration:
  - GPS tracking for services that involve physical presence, enabling users to track the location of service providers.

- Integration with maps for navigation.
- User Profile Management:
  - Users should be able to manage their profiles, update personal information, and review their booking history.
  - Service providers should have access to tools for managing their profiles and schedules.
- Admin Dashboard:
  - An administrative interface to manage user accounts, service provider profiles, and overall system settings.
  - Tools for monitoring and resolving disputes or issues.
- Reporting and Analytics:
  - Generation of reports on service usage, user engagement, and financial transactions.
  - Analytics tools for continuous improvement and decision-making.
- Security Measures:
  - Encryption of user data and secure communication channels.
  - Measures to prevent unauthorized access and protect sensitive information.

Non-functional Requirement:

- Performance Testing: Verify system response times and scalability under varying loads, including stress testing.
- Reliability Testing: Test system availability, recovery, and failover mechanisms.
- Security Testing: Assess authentication, authorization, data encryption, and vulnerability assessments.
- Usability Testing: Evaluate accessibility and user interface usability.
- Compatibility Testing: Ensure compatibility with different browsers, devices, and integration with external systems.
- Compliance Testing: Verify regulatory compliance and assess environmental impact.
- Maintenance Testing: Evaluate update and upgrade processes and the accuracy of system documentation.
- Data Backup and Recovery Testing: Test backup and restore procedures for data integrity.
- Cost-Efficiency Testing: Assess operational cost efficiency. This condensed list still captures the essential non-functional requirements for your test plan while making it more concise and focused.

1.2 Quality Objective

The main objectives of our proposed system are as follows:

i. Convenience and Accessibility
ii. Efficient Service Booking
iii. Diverse Service Offerings
iv. Reliable Service Providers
v. Transparent and Trustworthy Transactions
vi. Secure and Seamless Payments
vii. Effective Communication
viii. Optimized Resource Utilization
ix. User Satisfaction

1.3 Roles and Responsibilities

Detail description of the Roles and responsibilities of different team members like

- QA Analyst: Manvendra Kumar
- Test Manager: Mr. Abhishek Goyal
- Configuration Manager: Prof. Akankskha
- Developers: Mayank Gupta, Manvendra Kumar, Naman Nagaria

2 Test Methodology

2.1 Overview.

Hybrid Approach is used.
It can be effective when your project has a mix of well defined and evolving requirements. You can combine aspects of both Agile and waterfall to suit your project's unique needs. This approach allows you to maintain a structured plan while accommodating changes and feedback as necessary. Careful planning and coordination between phases are crucial to make the hybrid model work effectively.

2.2 Test levels

Testing levels define the types of testing to be executed on the application under test(AUT). The testing levels primarily depends on the scope of the project, time and budget constraints.
Performance Testing:
Response Time: Define acceptable response times for critical functions, such as retrieving sensor data or generating reports.
Scalability: Verify the system's performance under varying loads, including the ability to handle a larger number of sensors or simultaneous users.
Stress Testing: Assess the system's response under extreme conditions to ensure it can handle peak loads without critical failures.

Reliability Testing:
Availability: Specify the desired level of system uptime and test for reliability to minimize unplanned downtime.
Recovery Testing: Evaluate the system's ability to recover from failures or crashes without data loss or corruption.
Failover Testing: Verify that the system can smoothly transition to a backup or redundant system in case of a failure.

Security Testing:
Authentication and Authorization: Ensure that only authorized users have access to sensitive features and data.
Data Encryption: Verify that sensitive data is encrypted during transmission and storage to protect against unauthorized access.

Vulnerability Assessment: Regularly conduct security assessments, such as penetration testing, to identify and address potential vulnerabilities.

Usability Testing:
Accessibility: Confirm that the user interface is accessible to individuals with disabilities, including support for screen readers and keyboard navigation.
User Interface Testing: Assess the usability and user-friendliness of the application, considering user satisfaction and ease of use.

Compatibility Testing:
Browser and Device Compatibility: Ensure that the system functions correctly on various web browsers and devices, including mobile devices and different operating systems.
Integration Testing: Validate the system's ability to integrate with other software, databases, or external systems without conflicts.

2.3 Test Completeness

Here you define the criterias that will deem your testing complete.
For instance, a few criteria to check Test Completeness would be

- 100% test coverage
- All Manual & Automated Test cases executed
- All open bugs are fixed or will be fixed in next release

3 Test Deliverables

3.1 USER REGISTRATION

i. Positive Test Cases:
   a. Valid registration
      - Input valid information (name, email, password).
      - Verify that the user is registered successfully.
   b. Email Verification
      - After registration, check the email for a verification link
      - Click the verification link and verify that the user account is activated.
   c. Password Strength
      - Ensure that the system enforces password strength requirements
      - Test registration with a strong password
   d. Unique email address
      - Attempt to register with an email address that is not already in use.
      - Confirm that the system accepts the registration
   e. Confirmation message
      - After successful registration, verify that the user receives a confirmation message.
   f. Login after registration
      - Use the registered credentials to login immediately after successful registration.
      - Confirm that login is successful.
   g. User profile completeness
      - Verify that the user's profile is created with correct information provided during registration.

ii. Negative test cases:

a. Invalid email format
   - Attempt to register with an invalid email format
   - Confirm that the system displays an appropriate error message.
b. Short password
   - Register with a password that does not meet the minimum length requirement
   - Verify that the system prompts for a stronger password
c. Existing email address
   - Attempts to register with an email address that is already registered.
   - Confirm that the system displays a message indicating the email already in use.
d. Missing required fields
   - Submit the registration form with one or more mandatory fields empty.
   - Verify that the system prompts to fill in all required fields.
e. Registration timeout
   - Take too long to complete the registration process.
   - Confirms that the system handles a timeout scenario appropriately.
f. Special characters
   - Register with a password containing special characters.
   - Confirm that the system accepts and stores the password correctly.
   -

Scenario	Input	Output	Expected output	Result
Valid registration	Name, Email, Username, Password	POPUP: "Registered successfully"	POPUP: "Registered successfully"	PASS
Email verification	Email	POPUP: "Account activated"	POPUP: "Account activated"	PASS
Password Strength of less than 15 characters	Password	ACTION: Strong password Suggestion	ACTION: Strong password suggestion	PASS
Unique email address	Email (already registered)	WARNING: "Already registered"	WARNING: "Already registered"	PASS
Login after registration	Username or email, password	ACTION: Direct to homepage	ACTION: Direct to homepage	PASS
Missing required fields	-	WARNING: "Fill all required field"	WARNING: "Fill all required field"	PASS

Decision table for login credentials

Email or username	Password	Output	Result
Invalid	-	WARNING: "Account does not exists"	PASS
Valid	Invalid	WARNING: "Incorrect password"	PASS
Valid	Valid	POPUP: "Successfully login" ACTION: Direct to homepage	PASS

Equivalence Partitioning & Boundary Value Analysis For password strength

Scenario	Password Strength	Output	Expected output	Result
0	0	“Enter a password”	“Enter a password”	PASS
MIN - 1	7	“Password is too short”	“Password is too short”	PASS
8<=N<=15	8	Suggest Strong password	Suggest strong password	PASS
More than 15	15	Depends on module	-	PASS

3.2 SERVICE BOOKING:

i. Positive Test Cases:
   a. Browse Services
      - Navigate to the services section.
      - Verify that all available services are displayed correctly.
   b. Select Service
      - Choose a specific service from the list.
      - Confirm that the service details and options are accurate.
   c. Service Customization
      - If the service allows customization, test the ability to customize and book.
      - Verify that customized details are accurately reflected in the booking.
   d. Choose Service Provider
      - If users can select a specific service provider, test the selection process.
      - Confirm that the chosen service provider is assigned to the booking.
   e. Date and Time Selection
      - Pick a date and time for the service.
      - Confirm that the selected date and time are correctly displayed in the booking details.
   f. Multiple Service Booking
      - Test the ability to book multiple services in a single transaction.
      - Verify that each service is accurately reflected in the booking summary.
   g. Booking Confirmation
      - Complete the booking process.
      - Confirm that users receive a booking confirmation message.
   h. View Booking History
      - Navigate to the booking history section.
      - Verify that the recent booking is listed with the correct details.

ii. Negative test cases
   a. Invalid Date/Time
      - Attempt to book a service for an invalid date or time.
      - Confirm that the system prompts for a valid schedule.
   b. Booking Timeout
      - Take too long to complete the booking process.
      - Confirm that the system handles a timeout scenario appropriately.
   c. Select Unavailable Service Provider
      - Attempt to select a service provider who is unavailable.
      - Confirm that the system provides an appropriate error message.
   d. Exceed Maximum Services
      - Try to add more services than the system allows in a single booking.
      - Confirm that the system enforces the maximum limit.
   e. Book with insufficient funds
      - If applicable, attempt to book a service without sufficient funds in the user's account.

- Confirm that the system prompts for payment or displays an appropriate error.

f. Cancel Booking

- Test the cancellation process for a booked service.
- Confirm that the booking is canceled successfully.

g. Modify Booking

- If the application allows modifying booked services, test the modification process.
- Confirm that the changes are reflected in the booking details.
- Verify that error messages are clear and provide guidance on resolving issues during the booking process.

Scenarios	Input	Output	Expected output	Result
Invalid Time and Date	Time, Date	WARNING: "Invalid Time and Date"	WARNING: "Invalid Time and Date"	PASS
Unavailable Service Provider	Service Provider	WARNING: "Time slot is unavailable for service Provider"	WARNING: "Time slot is unavailable for service Provider"	PASS
Booking timeout	(too long to book a service)	POPUP: "Session timeout"	POPUP: "Session timeout"	PASS
Unavailable Service	Service	WARNING: "Sorry, This service is unavailable"	WARNING: "Sorry, this service is unavailable"	PASS
Multiple services	"ADD MORE SERVICES"	-	-	PASS
Booking confirmation	Complete booking details	MESSAGE: "Booking confirmed"	MESSAGE: "Booking confirmed"	PASS
Cancel Booking	-	MESSAGE: "Booking cancelled"	MESSAGE: "Booking cancelled"	PASS

Decision table for booking a service

Service	Time slot	Service provider	Output	Result
Unavailable	-	-	WARNING: "Service unavailable"	PASS
Available	Unavailable	-	WARNING: "Choose other slot" ACTION: Suggest available time slot	PASS
Available	Available	Unavailable	WARNING: "Choose other service provider" ACTION: Suggest available service provider	PASS
Available	Available	Available	MESSAGE: "Booking confirmed"	PASS

4 Resource & Environment Needs

4.1 Testing Tools

SQLUnit: A tool for database testing that allows you to define and run SQL-based tests on your database.

Selenium: A popular tool for automated functional testing of web applications, which can be useful for testing user interfaces in your system.

Jenkins: An open-source automation server that can be used for building, testing, and deploying code continuously.

4.2 Test Environment

It mentions the minimum **hardware** requirements that will be used to test the Application.

Following **software's** are required in addition to client-specific software.

- Windows 8 and above
- Office 2013 and above
- MS Exchange, etc.

5 Terms/Acronyms

Make a mention of any terms or acronyms used in the project

TERM/ACRONYM	DEFINITION
API	Application Program Interface
AUT	Application Under Test

TESTING FACULTY SIGNATURE:
Mr. Abhishek Goyal

GUIDE SIGNATURE:
Mr. Sreesh Gaur

PROJECT COORDINATOR:
Prof. Neha Shukla