Department of Computer Science Engineering

SRM IST, Kattankulathur – 603 203

18CSC206J – SOFTWARE ENGINEERING AND PROJECT MANAGEMENT

Experiment No	11
Title of Experiment	Testcases
Name of the Candidate	Sai Rohit P
Team Members	Sai Rohit (RA2111003010806) Pavan Sagar (RA2111003010809)
Date of Experiment	

Mark Split Up			
S.No	**Description**	**Maximum Mark**	**Mark Obtained**
1	Exercise	5	
2	Viva	5	
Total		**10**	

Staff Signature with date

Aim:

To develop the testcase manual for TimetableSOS

Team Members:

S. No.	Register Number	Name	Role
1	RA2111003010806	Sai Rohit	Rep
2	RA2111003010809	Pavan Sagar	Member

Project Title: TimetableSOS

Test Cases

Functional Test Cases

ID	Test Scenario	Test Case	Execution Steps	Expected Outcome	Actual Outcome	Status	Remarks
1	Verify user login functionality	1. Open the login page and enter valid username and password. 2. Click on the Login button. 3. Ensure that the system redirects the user to the dashboard page.	1. Login to the system as a user with the appropriate permissions. 2. Select the subject details to know the details. 3. Verify that the subject attendence up to date	The user should be able to log in to the system successfully and be redirected to the page.	The user was able to log in successfully and was redirected to the dashboard page.	Pass	Success

2	Verify subject details and functionality	1. Open the dashboard page 2. Select the sub code. 3. Ensure that the system displays the current subject details	1. Login to the system as a user with the appropriate permissions. 2. Set up a notification for a specific bus or route. 3. Wait for the bus to reach a designated location. 4. Verify that the notification is triggered at the appropriate time. 5. Repeat steps 2-4 for multiple notifications to ensure that the functionality is working for all notifications.	The system should be able to track the bus in real- time and display the current location on the map	The system was able to track the bus in real-time and display the current location on the map, and the location updated automatically.	Pass	None

3	Verify parent notification functionality	1. Open the dashboard page. Select the bus to track. 2. Set up a notification for a specific stop. 3. Ensure that the system sends a notification to the parent when the bus arrives at the stop.	1. Login to the system as a user with the appropriate permissions. 2. Select the report that you want to generate, such as a bus route report or a driver activity report. 3. Input the necessary parameters, such as the date range or bus route number. 4. Generate the report. 5. Verify that the report data is accurate and up-to-date	The parent should receive a notification when the bus arrives at their child's stop.	The system successfully sent a notification to the parent when the bus arrived at the stop.	Pass	None

4	Verify emergency alert functionality	Open the dashboard page.	1. Login to the system as an administrator. 2. Add a new user with the appropriate permissions. 3. Edit the permissions of an existing user. 4. Delete a user from the system. 5. Verify that the changes are reflected in the system and that the appropriate access is granted or revoked.	The system should be able to send emergency alerts to parents and school authorities.	The system successfully sent emergency alerts to parents and school authorities.	Pass	None

5	Verify Reporting Functionality	1. Open the reporting page. 2. Select the report type. 3. Enter the required parameters. 4. Verify that the system generates the report.	1. Login to the system as a user with the appropriate permissions. 2. Navigate to the map view. 3. Verify that the map is responsive and easy to use. 4. Verify that the and out smoothly and 5. Type your name are displayed clearly.	The system should be able to generate reports and student attendance.	The system was able to generate and student attendance.	Pass	None

Non-Functional Test Cases

ID	Test Scenario	Test Case	Execution Steps	Expected Outcome	Actual Outcome	Status	Remarks
1	Performance testing	1. Simulate a large number of concurrent users accessing the system. 2. Monitor the system response time. 3. Verify that the system can handle the load without slowing down or crashing.	1. Use a load testing tool to simulate a large number of concurrent users accessing the system. 2. Monitor the system response time using performance monitoring tools such as New Relic or AppDynamics. 3. Gradually increase the load until the system reaches its maximum capacity. 4. Observe the system behaviour, including response time, errors, and CPU/memory usage.	The system should be able to handle a large number of concurrent users and requests without slowing down.	The system was able to handle the load without slowing down or crashing.	Pass	None

2	Security testing	1. Attempt to access the system using incorrect login credentials. 2. Attempt to access another user's data. 3. Verify that the system blocks unauthorized access and provides appropriate error messages.	1. Attempt to log in using incorrect credentials, such as an incorrect username or password. 2. Attempt to access another user's data by changing the URL or modifying the request parameters. 3. Use tools such as OWASP ZAP or Burp Suite to perform security scans and identify vulnerabilities. 4. Verify that the system blocks unauthorized access and provides appropriate error messages. 5. Implement security fixes for any identified vulnerabilities.	The system should be secure and protect user data from unauthorized access.	The system successfully blocked unauthorized access and provided appropriate error messages.	Pass	None

3	Usability testing	1. Ask users to perform common tasks such as tracking a bus, setting up notifications, and generating reports. 2. Observe users as they complete the tasks and note any difficulties or confusion. 3. Gather feedback from users on the system's usability.	1. Create a list of common tasks that users are likely to perform on the system, such as tracking a bus, setting up notifications, and generating reports 2. Recruit a group of representative users and ask them to perform the tasks while being observed 3. Observe the users as they complete the tasks, noting any difficulties or confusion. 4. Gather feedback from users on the system's usability, using surveys or interviews. 5. Use the feedback to improve the system's usability.	The system should be easy to use and navigate, and users should be able to complete common tasks without difficulty.	Users found the system easy to use and navigate, and were able to complete common tasks without difficulty.	Pass	None

4	Compatibility testing	1. Access the system using different devices such as desktops, laptops, tablets, and mobile phones. 2. Use different browsers such as Chrome, Firefox, Safari, and Edge. 3. Verify that the system works correctly on each device and browser.	1. Test the system on different devices such as desktops, laptops, tablets, and mobile phones 2. Use different browsers such as Chrome, Firefox, Safari, and Edge. 3. Verify that the system works correctly on each device and browser, including functionality and appearance. 4. Use responsive design techniques to ensure that the system is optimized for different screen sizes.	The system should be compatible with different devices and browsers.	The system worked correctly on all tested devices and browsers.	Pass	None

5	Availability testing	1. Monitor the system uptime and availability. 2. Test the system backup and recovery procedures. 3. Verify that the system can recover from a disaster or outage.	1. Monitor the system uptime and availability using tools such as Nagios or Pingdom. 2. Test the system backup and recovery procedures by simulating a disaster or outage. 3. Verify that the system can recover from the disaster or outage within a reasonable amount of time. 4. Implement improvements or upgrades to improve system availability as needed.	The system should be available and accessible to users at all times, and should be able to recover from a disaster or outage.	The system was available and accessible to users at all times, and was able to recover from a simulated disaster or outage.	Pass	None

Result: Thus, the test case manual has been created for the TimetableSOS