Line following robot System

Guided By: **Manas Patidar**

Dr.Shaligram Prajapat Sir   **IT-2k19-28**

BONAFIDE CERTIFICATE

Certified that this project titled " line following robot System " is a bonafide work of MANAS PATIDAR who carried out the researched and completed the project under my supervision. Certified further, that to the best of my knowledge, the work reported herein does not form part of any other project on the basis of which a

degree or award was conferred on an earlier occasion on this or any other candidate.

Internal examiner   External examiner

RECOMMENDATIONS

The Project wok entitled "line following robot System" submitted by MANAS PATIDAR is satisfactory account of the bona fide work under my supervision and is recommended towards the end of their III year of MTECH(I.T)-2024.

Guided By:

ACKNOWLEDGEMENT

I would like to express my gratitude toward staff of IIPS -DAVV as well as the honourable director Dr. B. K. Tripathi for providing us a great opportunity to complete a project on "line following robot System". My sincere thanks go to Dr. Shaligram Prajapat without his support and guidance for the completion of this project.

SYSTEM

TESTING

Introduction

Once source code has been generated, software must be tested to uncover (and correct) as many errors as possible before delivery to customer. Our goal is to design a series of test cases that have a high likelihood of finding errors. To uncover the errors software techniques are used. These techniques provide systematic guidance for designing test that

(1) Exercise the internal logic of software components, and

(2) Exercise the input and output domains of the program to uncover errors

In program function, behavior and performance.

Steps: Software is tested from two different perspectives:

(1) Internal program logic is exercised using ―White box‖ test case design Techniques.

(2) Software requirements are exercised using ―block box‖ test case Design techniques.

In both cases, the intent is to find the maximum number of errors with the Minimum amount of effort and time.

Testing Methodologies:

A strategy for software testing must accommodate low-level tests that are necessary to verify that a small source code segment has been correctly implemented as well as high-level tests that validate major system functions against customer requirements. A strategy must provide guidance for the practitioner and a set of milestones for the manager. Because the steps of the test strategy occur at a time when deadline pressure begins to rise, progress must be measurable and problems must surface as early as possible. Following testing techniques are well known and the same strategy is adopted during this project testing.

Unit testing:

Unit testing focuses verification effort on the smallest unit of software design-the software component or module. The unit test is white-box oriented. The unit testing implemented in every module of student attendance management System. by giving correct manual input to the system ,the datas are stored in database and retrieved. If you want required module to access input or get the output from the End user. any error will accrued the time will provide handler to show what type of error will accrued .

System testing:

System testing is actually a series of different tests whose primary purpose is to fully exercise the computer-based system. Below we have described the two types of testing which have been taken for this project. it is to check all modules worked on input basis .if you want change any values or inputs will change all information. so specified input is must.

6.2.4 Performance Testing

Performance testing is designed to test the run-time performance of software within the context of an integrated system. Performance testing occurs throughout all steps in the testing process. Even at the unit level, the performance of an individual module may be assessed as white-box tests are conducted.

This project reduce attendance table, codes. it will generate report fast.no have extra time or waiting of results .entered correct data will show result few millisecond. justused only low memory of our system. Automatically do not getting access at another software. Get user permission and access to other applications.

Test cases

Test case is an object for execution for other modules in the architecture does not represent any interaction by itself. A test case is a set of sequential steps to execute a test operating on a set of predefined inputs to produce certain expected outputs. There are two types of test cases:-*manual* and *automated.* A manual test case is executed manually while an automated test case is executed using automation.

In system testing, test data should cover the possible values of each parameter based on the requirements. Since testing every value is impractical, a few values should be chosen from each equivalence class. An equivalence class is a set of values that should all be treated the same.

Ideally, test cases that check error conditions are written separately from the functional test cases and should have steps to verify the error messages and logs. Realistically, if functional test cases are not yet written, it is ok for testers to check for error conditions when performing normal functional test cases. It should be clear which test data, if any is expected to trigger errors.

TEST CASE:

Agent and admin login form

Sno	Test case id	Test case name	Test case desc	Step	Expected result	Actual Result	Test case status pass/fail
1	Login admin	Validate login	To verify that login name on login page	Enter the login name and password and click submit button	Login successful or an error message "In valid login or password" must be displayed	Login successful	Pass
2	Login Staff	Validate login	To verify that login name on login page	Enter the login name and password and click submit button	Login successful or an error message "In valid login or password" must be displayed	Login successful	Pass
3	Password	Validate password	To verify that password on login page	Enter password and login name click submit button	An error message "password invalid" must be displayed	An error message "password invalid" must be displayed	fail