

|  |  |
| --- | --- |
| **ASSIGNMENT** | |
| **Course Code** | CSC210A |
| **Course Name** | Software Development Fundamentals |
| **Programme** | B. Tech. |
| **Department** | Computer Science and Engineering |
| **Faculty** | FET |

#### 

|  |  |
| --- | --- |
| **Name of the Student** | Satyajit Ghana |
| **Reg. No** | 17ETCS002159 |
| **Semester/Year** | 4th/2nd |
| **Course Leader/s** | Ms.Sahana.P.Shankar |

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Declaration Sheet** | | | | | | | | |
| Student Name | Satyajit Ghana | | | | | | | |
| Reg. No | 17ETCS002012 | | | | | | | |
| Programme | B. Tech | | | | | Semester/Year | 4th/2nd | |
| Course Code | CSC210A | | | | | | | |
| Course Title | Software Development Fundamentals | | | | | | | |
| Course Date |  | | to | |  | | | |
| Course Leader | Ms.Sahana.P.Shankar | | | | | | | |
| **Declaration**  The assignment submitted herewith is a result of my own investigations and that I have conformed to the guidelines against plagiarism as laid out in the Student Handbook. All sections of the text and results, which have been obtained from other sources, are fully referenced. I understand that cheating and plagiarism constitute a breach of University regulations and will be dealt with accordingly. | | | | | | | | |
| Signature of the Student | |  | | | | | Date |  |
| Submission date stamp  (by Examination & Assessment Section) | |  | | | | | | |
| Signature of the Course Leader and date | | | | Signature of the Reviewer and date | | | | |
|  | | | |  | | | | |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Faculty of Engineering and Technology** | | | | | | |
| **Ramaiah University of Applied Sciences** | | | | | | |
| Department | | Computer Science and Engineering | | | Programme | B.Tech |
| Semester/Batch | | 4th/2017 | | | | |
| Course Code | | CSC210A | | | Course Title | Software Development Fundamentals |
| Course Leader(s) | | Ms.Sahana.P.Shankar and Ms.Supriya M S | | | | |
| **Assignment – 2** | | | | | | |
| **Reg.No.** |  | | **Name of Student** |  | | |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | | | | | |
| **Sections** | **Marking Scheme** | | **Marks** | | |
| **Max Marks** | **First Examiner Marks** | **Moderator** |
| **Part A** |  | | | | |
| A **1.1** | Introduction to the topic | 1 |  |  |
| A **1.2** | Critical comparison | 3 |  |  |
| A **1.3** | Conclusion | 1 |  |  |
|  | **Part-A Max Marks** | **5** |  |  |
| **Part B.1** |  | | | | |
| B **1.1** | Introduction to state chart | 2 |  |  |
| B **1.2** | State chart design | 7 |  |  |
| B **1.3** | Conclusion | 1 |  |  |
|  | **B.1 Max Marks** | **10** |  |  |
| **Part B.2** |  | | | | |
| B **2.1** | Introduction to ER diagram | 2 |  |  |
| B **2.2** | ER diagram design | 7 |  |  |
| B **2.3** | Conclusion | 1 |  |  |
|  | **B.2 Max Marks** | **10** |  |  |
| **Total Assignment Marks** | | | **25** |  |  |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Course Marks Tabulation** | | | | |
| **Component-1 (B) Assignment** | **First Examiner** | **Remarks** | **Moderator** | **Remarks** |
| A |  |  |  |  |
| B.1 |  |  |  |  |
| B.2 |  |  |  |  |
| **Marks (Max 25 )** |  |  |  |  |
| **Signature of First Examiner Signature of Moderator** | | | | |

# Part-A (05 Marks)

Software testing is the process of finding the defects in software, those bugs / defects once found can be fixed to improve the quality of the end product. Quick and frequent software development cycles affect the test engineer’s ability to execute the test cases within the project deadlines. The main aim of test automation is to do more work in less time. Software test engineers play a vital role in developing the test cases and automation tools.

In this context, debate on the statement: ***“Software Test Engineer’s job is at stake with increase in emphasis on automation testing”.***

Your debate should emphasize on:

**A1.1 Introduction to the topic:**

Automation testing is a technique uses an application to actualize whole life cycle of the software in less time and provides efficiency and effectiveness to the testing software.

Automation testing is an Automatic technique where the tester writes scripts by possess and uses suitable software to test the software. It is basically an automation process of a manual process. Like regression testing, Automation testing also used to test the application from load, performance and stress point of view.

**A1.2 Critical comparison between manual and automation software testing:**

**AUTOMATION TESTING**

The automation software can also enter test information into the System Under Test, look at expected and real results and create itemized test reports. Test Automation demands considerable investments of cash and resources.

Successive improvement cycles will require execution of same test suite more than once. Using a test automation device, it's possible to record this test suite and re-play it as required. When the test suite is mechanized, no human mediation is required. This improved ROI of Test Automation. The objective of Automation is to diminish the quantity of test cases to be run manually and not to dispose of Manual Testing altogether.

* 70% faster than the manual testing
* More extensive test inclusion of application features
* Dependable in results
* Ensure Consistency
* Saves Time and Cost
* Improves exactness
* Human Intervention is not required while execution
* Increases Efficiency
* Better speed in executing tests
* Re-usable test scripts
* Test Frequently and thoroughly
* More cycle of execution can be accomplished through automation
* Early time to showcase

**MANUAL TESTING**

Manual testing is the process of manually testing software for defects. It requires a tester to assume the job of an end user whereby they use most of the application's features to ensure correct behavior. To ensure completeness of testing, the tester often follows a composed test plan that leads them through a set of important test cases.

A key step in the process is testing the software for correct behavior prior to release to end users.

For small scale building efforts , exploratory testing might be sufficient. With this informal methodology, the tester does not pursue any rigorous testing technique, yet rather explores the user interface of the application using as huge numbers of its features as possible, using information picked up in prior tests to instinctively infer extra tests. The success of exploratory manual testing relies intensely on the space expertise of the tester, because an absence of information will prompt incompleteness in testing. One of the key advantages of an informal methodology is to pick up an instinctive insight to how it feels to use the application.

Expansive scale building projects that depend on manual software testing pursue a more rigorous procedure so as to augment the quantity of defects that can be found. A systematic methodology focuses on foreordained test cases and for the most part involves the accompanying steps.

**COMPARISON**

Test automation might probably lessen or wipe out the cost of genuine testing. A PC can pursue a repetition sequence of steps more rapidly than a person, and it can run the tests medium-term to present the results in the morning. Nonetheless, the labor that is saved in real testing must be spent instead authoring the test program. Contingent upon the sort of application to be tested, and the automation tools that are chosen, this may require more labor than a manual methodology. What's more, some testing tools present an extremely expansive measure of information, possibly making a time consuming task of deciphering the results.

Things such as gadget drivers and software libraries must be tested using test programs. What's more, testing of substantial numbers of users (performance testing and load testing) is commonly simulated in software rather than performed practically speaking.

Conversely, graphical user interfaces whose format changes as often as possible are exceptionally hard to test naturally. There are test frameworks that can be used for regression testing of user interfaces. They depend on recording of sequences of keystrokes and mouse gestures, then playing them back and observing that the user interface responds similarly without fail. Unfortunately, these recordings may not work properly when a catch is moved or relabeled in a subsequent release. A programmed regression test may also be tricked if the program output varies significantly.

**A1.3 Justification with stance taken and conclusion**

In the race of "Faster Time to Market", Automation testing has turned into the device to accomplish it. On the off chance that the real skill of tester is lost, then this present reality QA criticism will be lost. It is because when the Software undergoes automation testing, it will be a testing process of executing test scripts created via automation test engineers which are more towards covering positive test scenarios because an automation engineer thinks like a developer. Covering in general negative test cases will be a test.

Automation test engineers influence automation testing tools and their specialized skills to make automation test scripts however they need real space learning and testing skills which are the most essential elements for quality software conveyance.

Genuine testers, who are passionate about testing and who are capable at testing, use all their space information to distinguish all possible test scenarios which

address basic business scenarios. Moreover, they are all around experienced and sufficiently skilled to make compelling test cases to surround all the recognized test scenarios.

# Part B (20 Marks)

Scenario: An online reservation system aids in the efficient management of activities such as reservation and cancellation of train, bus, flight and movie tickets, and hotel rooms. You are required to develop an online reservation system in consultation with your course leader:

Based on the previously completed design assignment (Assignment 1: B.1 and B.2), you are to:

* 1. **(10 Marks)**

Design the state chart for the system developed. Your report should include the following:

**B1.1 Introduction to state chart :**

A State chart diagram depicts a state machine. State machine can be characterized as a machine which characterizes diverse states of an object and these states are constrained by external or internal events.

State chart diagram is one of the five UML diagrams used to show the dynamic idea of a system. They characterize distinctive states of an object amid its lifetime and these states are changed by events. State chart diagrams are valuable to demonstrate the reactive systems. Reactive systems can be characterized as a system that reacts to external or internal events.

State chart diagram portrays the flow of control starting with one state then onto the next state. States are characterized as a condition in which an object exists and it changes when some occasion is activated. The most important reason for State chart diagram is to demonstrate lifetime of an object from creation to end.

State chart diagrams are additionally utilized for forward and reverse engineering of a system. Nonetheless, the principle reason for existing is to demonstrate the reactive system.

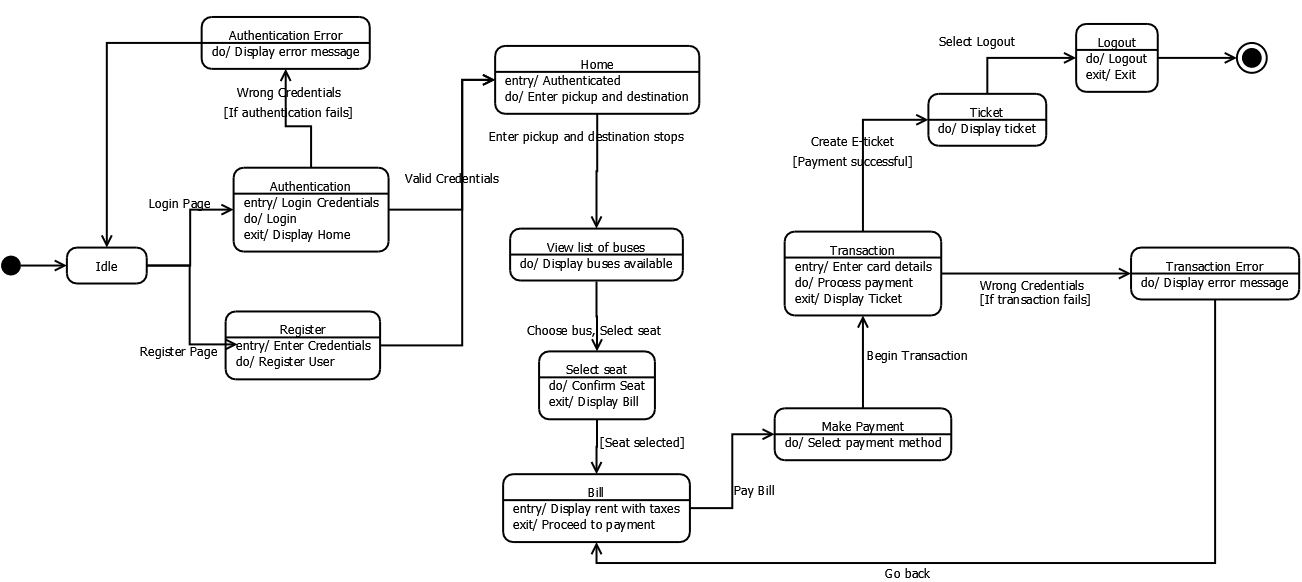
Following are the primary motivations behind utilizing State chart diagrams −

* To demonstrate the dynamic part of a system.
* To demonstrate the existence time of a reactive system.
* To portray distinctive states of an object amid its life time.
* Characterize a state machine to show the states of an object.

**B1.2 State chart design:**

The state chart diagram is based on online bus ticket booking system.

Initially, a user login a software (on the off chance that the user is new, at that point sign in). After login check the seat availability in the following options(if seats are not available at that point go final state). After that,payment for booking tickets . This software allows distinctive sorts of payment, for example, wallets and so forth. Now, e-ticket is displayed.

****

**B1.3 Conclusion:**

I reasoned that state chart diagram is the most valuable strategy In the designing of software.

State diagrams are the ideal way to display object life cycles. State diagrams enable you to portray the behavior of articles during as long as they can remember span. In addition, the distinctive states and state changes as well as occasions causing transitions can be depicted.

# (10 Marks)

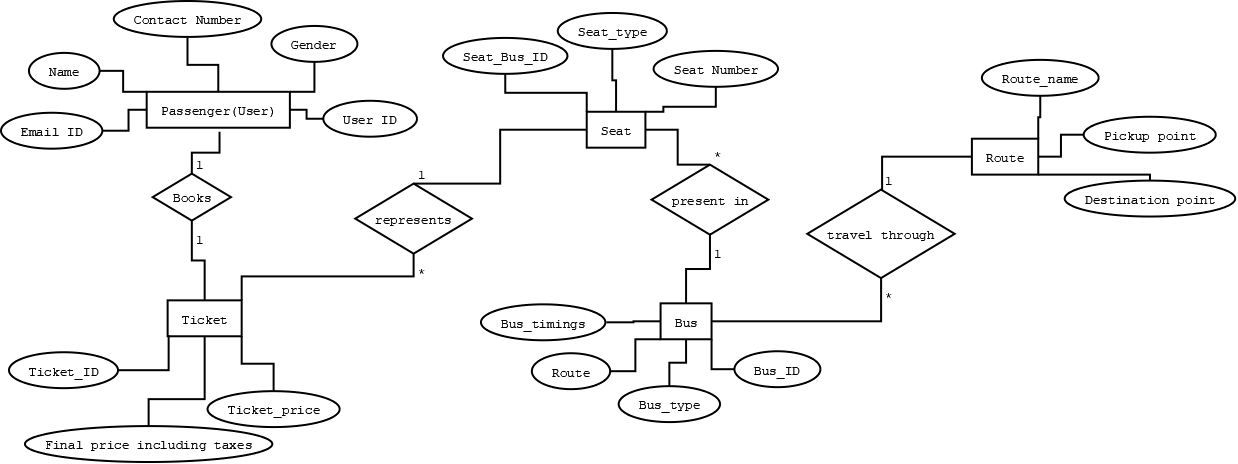
Design the Entity-Relationship diagram for the entire online reservation software. Your report should include the following:

**B2.1 Introduction to Entity-Relationship diagram:**

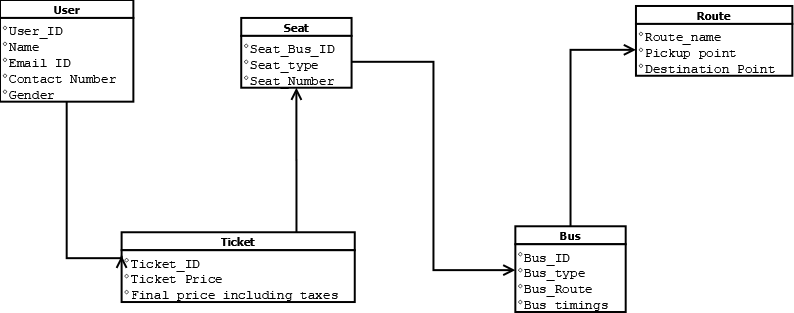
An entity– relationship model depicts interrelated things of enthusiasm for a particular domain of learning. A fundamental ER model is made out of entity types and determines relationships that can exist between entities .

In software engineering, an ER model is generally formed to speak to things a business needs to recollect so as to perform business forms. Therefore, the ER model turns into an abstract data model, that characterizes a data or information structure which can be actualized in a database, normally a social database.

**B2.2 Entity-Relationship design diagrams:**

****

Database Table:

****

**B2.3 Conclusion:**

I finished up ENTITY RELATIONSHIP DIAGRAM displays the relationships of entity set put away in a database .The ER diagrams are easy change to any data model and it is conceptually straightforward as compare to all different diagrams, for example, state chart diagram, activity diagrams and so on.

Disadvantages of ER diagram are

1. Loss of information content

2. restricted relationship representation

3. No representation of data manipulation.