GENESIS - Learning Outcome & Mini-project Summary Report

LTTS GLOBAL ENGINEERING ACADEMY

L&T Technology Services

Details

Ver. Rel. No.	Release Date	Prepared. By	Reviewed By	To be Approved	Remarks/Revision Details
1	11-11-2020	Raj Shekhar Mishra			Selenium, javaScript and python
2					
3					

Contents

Table of Figures:

List of tables:

Miniproject -1(Web Automation and testing using Java based Selenium and Cucumber) [Team]

Modules

"Modules linked to the miniproject–SDLC, Java, Selenium and Cucumber"

Topic and Subtopics

Core Topic:

Web page automation

Automation of a web page using Java based selenium scripts run using Eclipse IDE

Sub Topics:

Testing the automated page using cucumber framework

Creation of feature files

Run configuration – Cucumber feature

Objectives & Requirements

Objective:

- To automate a sample webpage and to test if all the functionalities on the webpage is working correctly.
- A travel webpage, https://www.phptravels.net/home was chosen.
- Two selenium script were written in JAVA. First script is for entering the user's booking information like destination, check-in and checkout, number of members to accommodate.
- From this page, it has to navigate to the payment and hotel selection page which is written in the next script.

Following which cucumber based testing of these web automation was done.

- Understanding Java and selenium tool.
- Testing the automated web page using Cucumber framework

Requirements (High level and low level):

ID	Description
HL_01_L_01 HL_01_L_02	High level 01 – Automate a webpage using Selenium JAVA Low level 01 – Invoke Eclipse IDE Low level 02 – Adding the required Selenium and JAVA dependencies
HL_02_L_01 HL_02_L_01	High level 02 – Launching a chrome browser Low level 01 – Create a MAVEN repository Low level 02 – Create a driver folder and setup an executable chrome driver
HL_03_L_01	High level 03 – Automate a registration page

	Low level 01 – Find a suitable web page Low level 02 – Invoke the selenium automation framework through eclipse
HL_04_L_01	High level 04 – Feature file creation and BDD test case execution using cucumber framework Low level 01 – Invoke cucumber framework
HL_05_L_01	High level 05 – Execute and pass a few mentioned scenarios Low level 01 – Implement the test cases for following scenarios: • Given launching chrome • When launching URL • Then check functionality

Table 1 (Requirements)

Design

Behavioral Diagram

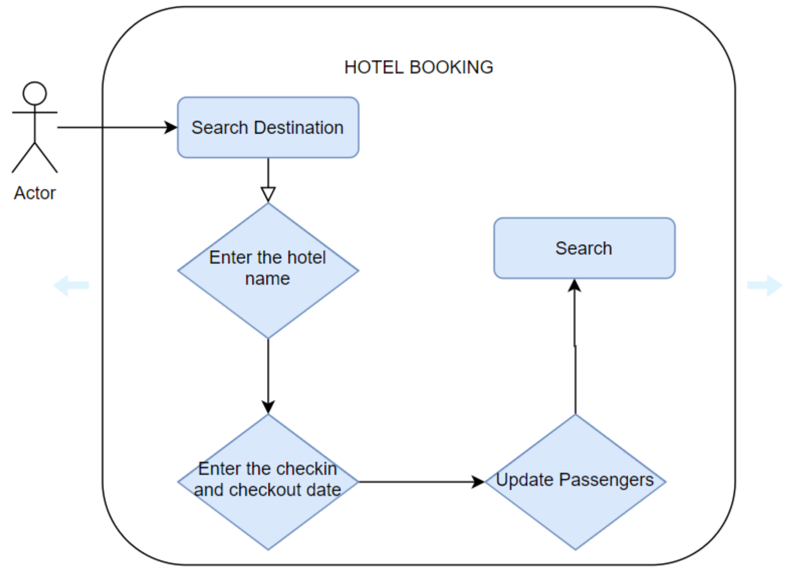


Figure 1 (Behavioral Diagram)

Structural Diagram

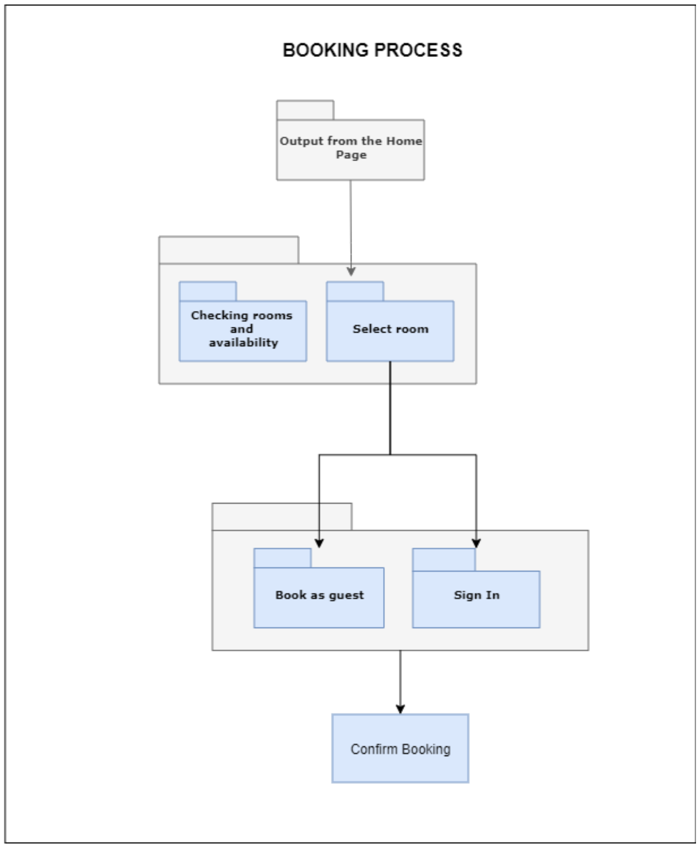


Figure 2 Structural Diagram

Test Plan

Unit Level Test Cases:

ID	Description	Precondition	Expected input	Expected output	Actual output
Tc1	Check if the chrome driver has launched the URL	An executable chrome driver is present in the driver folder	Hit the webpage using chrome driver	Browser launch	Browser launched
Tc2	Check whether destination can be chosen from the dropdown	Dropdown for destination has unique id or tags	Java code to navigate to the destination element and to select the desired destination	Destination should be selected	Destination selected

Tc3	Check whether check-in and checkout can be chosen from the dropdown	Dropdown for check-in and checkout can be accessed	Java code to navigate to the check-in and checkout element and to select the desired dates	Dates should be selected	Dates are selected
Tc4	Updating the number of adults and children should be possible	Button to increment the number of members should be accessible	Java code to access the update adult and children tag	Number of adults and children should be selected	Number of children and adults selected
Tc5	Check whether checkout details entered can be submitted	Button to submit present	Java code to navigate to checkout tag and click checkout	Redirects to the next page	Redirected to the next page after submission
Tc6	Select one of the rooms from the listed hotels	Option to select the room available	Java code to navigate to the select tag and select	Room should get selected	Room selected
Tc7	Navigate to the next section with the details entered	Book now option available	Code to click the book now button	Move to the payment page	Redirected to the payment page

Table 2 Unit Level Test Cases:

Integration level Test cases:

Tc1	Check whether the functionalities of all the pages are passing the cucumber tests	Launch chrome browser	Corresponding Java code and cucumber tags	Launch the corresponding page with all functionalities	Launched the corresponding page with all functionalities

Table 3 Integration level Test cases

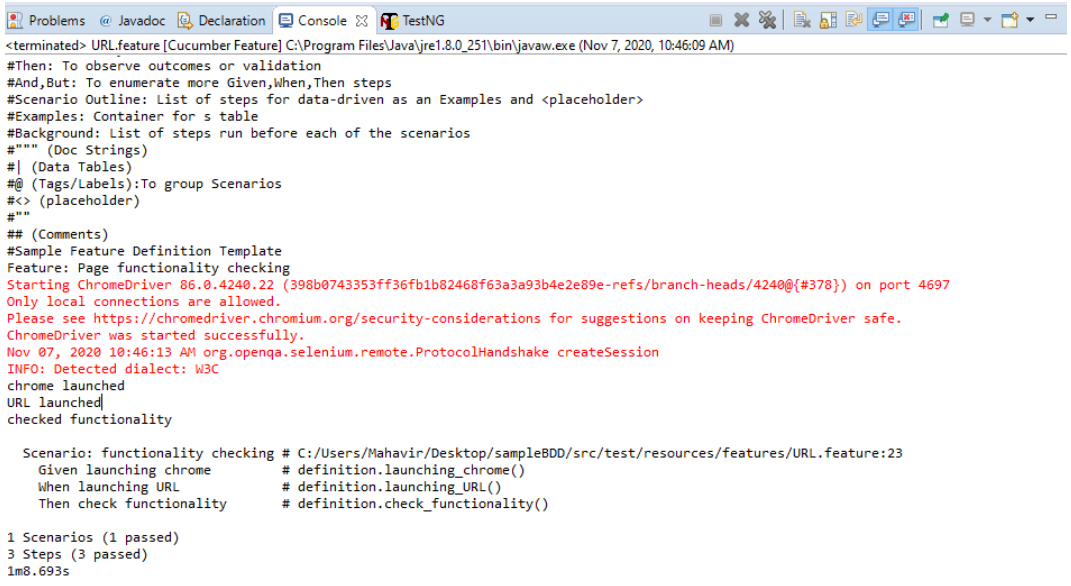
Implementation Summary

A travel webpage was automated using Selenium tool with and all the functionalities in the page was automated and Behavioral Driven Development (BDD) testing was done using cucumber framework. The following scenarios were tested using cucumber tags:

Feature: Page functionality testing
Scenario:

- Given launching chrome
- When launching URL
- Then check functionality

Output:



```
<terminated> URL.feature [Cucumber Feature] C:\Program Files\Java\jre1.8.0_251\bin\javaw.exe (Nov 7, 2020, 10:46:09 AM)
#Then: To observe outcomes or validation
#And,But: To enumerate more Given,When,Then steps
#Scenario Outline: List of steps for data-driven as an Examples and <placeholder>
#Examples: Container for s table
#Background: List of steps run before each of the scenarios
#""" (Doc Strings)
#| (Data Tables)
#@ (Tags/Labels):To group Scenarios
#<> (placeholder)
#""
## (Comments)
#Sample Feature Definition Template
Feature: Page functionality checking
Starting ChromeDriver 86.0.4240.22 (398b0743353ff36fb1b82468f63a3a93b4e2e89e-refs/branch-heads/4240@{#378}) on port 4697
Only local connections are allowed.
Please see https://chromedriver.chromium.org/security-considerations for suggestions on keeping ChromeDriver safe.
ChromeDriver was started successfully.
Nov 07, 2020 10:46:13 AM org.openqa.selenium.remote.ProtocolHandshake createSession
INFO: Detected dialect: W3C
chrome launched
URL launched
checked functionality

  Scenario: functionality checking # C:/Users/Mahavir/Desktop/sampleBDD/src/test/resources/features/URL.feature:23
    Given launching chrome         # definition.launching_chrome()
    When launching URL             # definition.launching_URL()
    Then check functionality       # definition.check_functionality()

1 Scenarios (1 passed)
3 Steps (3 passed)
1m8.693s
```

Figure 3 Output Eclipse

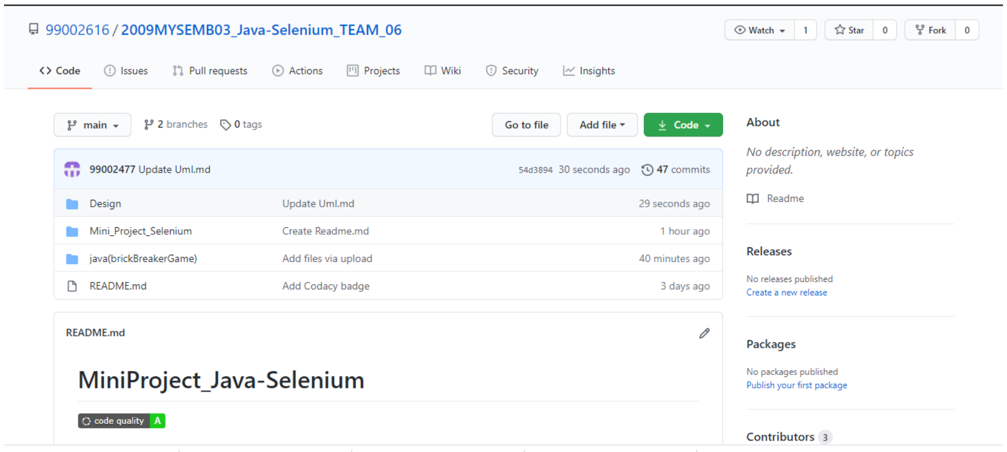
Video Summary

Video Link

Git Link

Git Repository

Git Dashboard



99002477 Update Uml.md 54d3894 30 seconds ago 47 commits

Design	Update Uml.md	29 seconds ago
Mini_Project_Selenium	Create Readme.md	1 hour ago
java(brickBreakerGame)	Add files via upload	40 minutes ago
README.md	Add Codacy badge	3 days ago

README.md

MiniProject_Java-Selenium

code quality A

About

No description, website, or topics provided.

Readme

Releases

No releases published
Create a new release

Packages

No packages published
Publish your first package

Contributors 3

Figure 4 Git Dashboard

Individual Contribution & Highlights

Sl.No	Name	Contributions
1	Shandhiya V.S (99002477)	Done with the selenium automation of web page. Documentation of report and ppt.
2	Raj Shekhar Mishra (99002616)	Done with the BDD testing of automated page with cucumber framework. Documentation of report and ppt.
3	Soniya Chandran (99002587)	Done with git repository creation and code quality checking. Documentation of report and ppt.

Table 4 Individual Contribution & Highlights

Challenges faced and how were they overcome

Challenge 1: Timeout issue and delay in loading the webpage.
Solution: This was handled by including Threadsleep() in the code

Challenge 2: Fetching a strong Xpath was difficult with certain tags
Solution: hence, gave relative Xpath with stronger tags at certain places.

Mini-Project -2 Java Script and Jasmine framework[Team]

Modules

- Java Script and Jasmine Framework

Topic and Subtopics

"Briefly list the core topics and subtopics being implemented and how"

- **JavaScript** is one of the 3 languages all web developers must learn:
  1. HTML to define the content of web pages
  2. CSS to specify the layout of web pages
  3. JavaScript to program the behavior of web pages
- **Jasmine Framework** is an open-source JavaScript system, fit for testing any sort of JavaScript application. Jasmine follows Behavior Driven Development (BDD) methodology to guarantee that each line of JavaScript articulation is appropriately unit tested. Jasmine provides a small syntax to test the smallest unit of the entire application instead of testing it as a whole.

Objectives & Requirements

Objectives:

To develop a digital alarm clock web page using Java Script and testing it with Jasmine Framework.

High level requirement:

ID	Description
HL_01	The HTML file is developed to give the structure of the web page
HL_02	The CSS file is developed to give the style of the web page
HL_03	The JS file is developed to give an interactive web page
HL_04	Jasmine framework is created for unit tests and passes

Table 5 High level requirement

Low Level Requirement:

ID	Description
HL_01_L_01	The CSS and JS file developed must be referenced in HTML file.
HL_01_L_02	wakeUpTimeSelector, lunchTimeSelector, napTimeSelector is implemented.
HL_01_L_03	partyTimeButton is developed.

HL_02_L_01	The color, width ,size of the web page is specified.
HL_02_L_03	The size and width of the image used in the web page is specified.
HL_02_L_04	The physical features of the party button is implemented.
HL_03_L_01	Variables are created.
HL_03_L_02	Function showCurrentTime is developed to show the current time
HL_03_L_03	The hours, minutes and seconds are set and put together in a string to display the current time in web page.
HL_03_L_04	Function updateClock is implemented for incrementing the clock on its own.
HL_03_L_05	The time is selected by the user and the corresponding image is shown in the web page
HL_04_L_01	The unit test cases corresponding to the clock is created and passed

Table 6 Low Level Requirements

Design

Structural Diagram

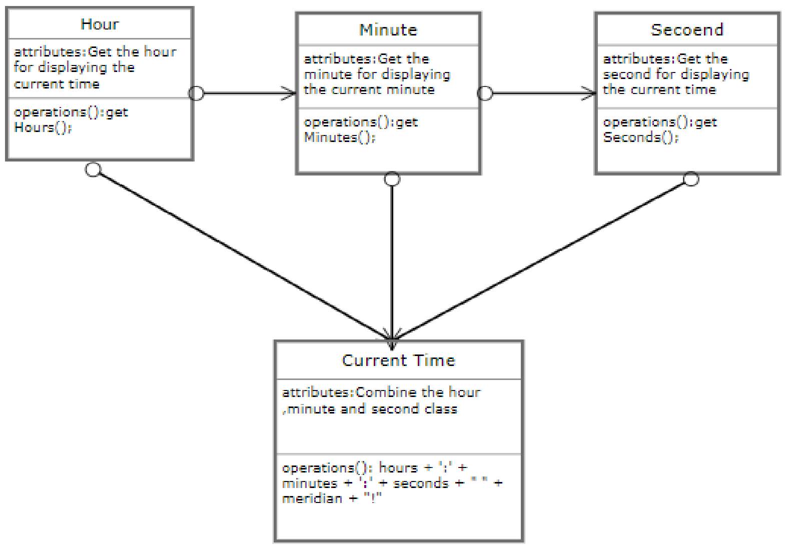

Hour
attributes:Get the hour for displaying the current time
operations():get Hours();
Minute
attributes:Get the minute for displaying the current minute
operations():get Minutes();
Secoend
attributes:Get the second for displaying the current time
operations():get Seconds();
Current Time
attributes:Combine the hour ,minute and second class
operations(): hours + ':' + minutes + ':' + seconds + " " + meridian + "!"

Figure 5 Structural Diagram

Behavioral Diagram

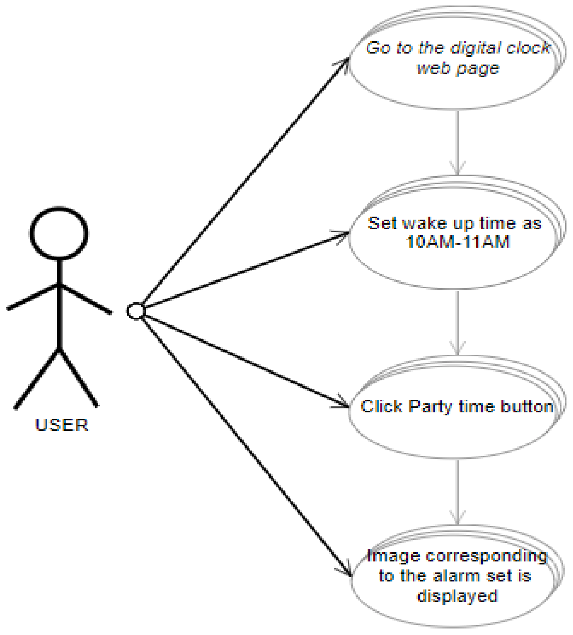

Go to the digital clock web page
Set wake up time as 10AM-11AM
Click Party time button
Image corresponding to the alarm set is displayed
USER

Figure 6 Behavioral Diagram

Test Plan

Unit level test case

ID	Description	Pre-condition	Expected input	Expected output	Actual output
TC_HL_01	Check whether the HTML file is able to display the output in the web page	The codes must not have errors.	Click the HTML file	The web page is displayed.	The web page is displayed.
TC_HL_02	Check whether the CSS file gives the structure to the web page	The physical parameters of the web page must be specified	Click the HTML file	The layout structure of the web page is shown	The layout structure of the web page is shown

TC_HL_03	Check whether the JS file is able to run	The functions and variables are specified	Click the HTML file	The functions are called and output is displayed in web page	The functions are called and output is displayed in web page
TC_HL_04	Check whether the unit test is created for the web page	Place the source and spec file in the	Run the spec runner file	Test cases are passes	Test cases are passes

Table 7 High level Requirements

Low Level Test Plan

ID	Description	Pre-condition	Expected input	Expected output	Actual output
TC_HL_01_L_01	Check whether the CSS and JS files are referenced in HTML file	CSS and JS file must not have errors	Referencing CSS and JS file	The web page is displayed	The web page is displayed
TC_HL_01_L_02	Check whether wakeUpTimeSelector,lunchTimeSelector,napTimeSelector works perfectly	The code must be error free	The alarm time is selected from the scroll	The corresponding time selected is displayed in the web page	The corresponding time selected is displayed in the web page
TC_ HL_02_L_01	Check whether the party time button displays image when clicked	Button id must be specified	Click the party time button	Image is displayed	Image is displayed
TC_ HL_02_L_02	Check whether the webpage has color width and margin	The CSS file must have these attributes	Click the HTML document	The attributes are shown in the web page	The attributes are shown in the web page

TC_ HL_02_L_03	Check whether the webpage image has the appropriate size and width	The CSS file must have these attributes	Click the HTML document	The attributes are shown in the web page	The attributes are shown in the web page
TC_ HL_02_L_04	Check whether the party time button has the physical attributes	The CSS file must have these attributes	Click the HTML file	The attributes are shown in the web page	The attributes are shown in the web page
TC_ HL_03_L_01	Check whether the variables of the project is declared	Open the JS file	JS file is examined	The code will not run without variable specification	The code will not run without variable specification
TC_ HL_03_L_02	Check whether showCurrentTime is specified in the code	The variables must be specified	Click the HTML file	The user gets hour, minute and second	The user gets hour, minute and second
TC_ HL_03_L_03	Check whether the hour, minute and second are put together in the string	The user must get the hour, minute and second	Click the HTML file	The current time is displayed in the web page	The current time is displayed in the web page
TC_ HL_03_L_04	Check whether the function updateClock increments on its own	The current time is specified	Click the HTML file	The current time increments on its own in the web page.	The current time increments on its own in the web page.
TC_ HL_03_L_05	Check whether the corresponding image is shown when the user selects the	The current time must be shown	The user chooses alarm time	The image is displayed in the web page	The image is displayed in the web page

	alarm time				
TC_ HL_04_L_01	Check whether the unit test cases in jasmine framework has passed	The JS ,CSS and HTML file must be ready	The JS file must be referenced in source and the unit test file must be referenced in spec	The test cases have passes	The test cases have passes

Table 8 Low level requirements

Individual Contribution & Highlights

S.No	Name	Contributions
1	Raj Shekhar Mishra 99002616	Developed JS FILE,HTML FILE and CSS file Documentation of report and ppt.
2	S.Gowsalya 99002470	Developed CSS file, Jasmine Framework and uploaded it in the git hub. Documentation of report and ppt.
3	R.Harine Parvathi 99002472	Developed JS file ,Jasmine Framework Documentation of report and ppt.

Table 9 Individual Contribution & Highlights

Implementation Summary

- We have put an effort to make a digital clock web page which will show the current time of India(IST). We have implemented this using html, java script and cascading style sheet(css).
- This digital clock can also be used as a alarm clock where the user can select the time for his/her sleep, wake up time, lunch time and party time.
- Along with this we have also implemented a testing framework using Jasmine. By checking the wakeup time, lunchtime, nape time are in correct format and minimum character's in it, when it is displayed. It also checks the format of the clock.

Output

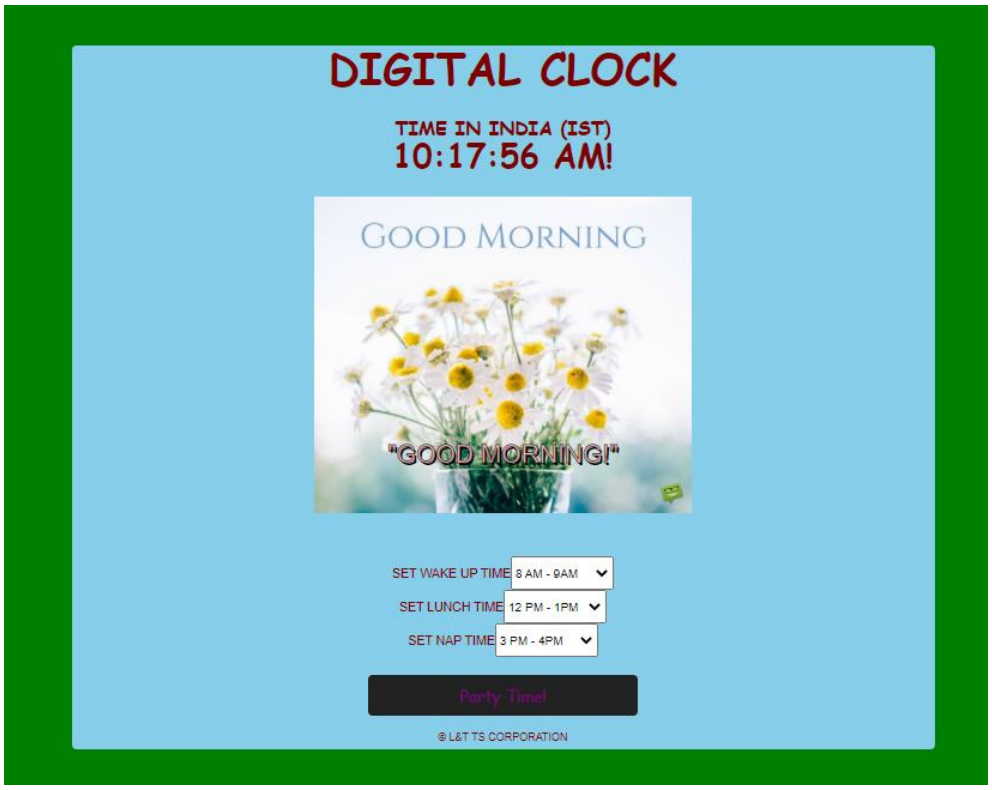
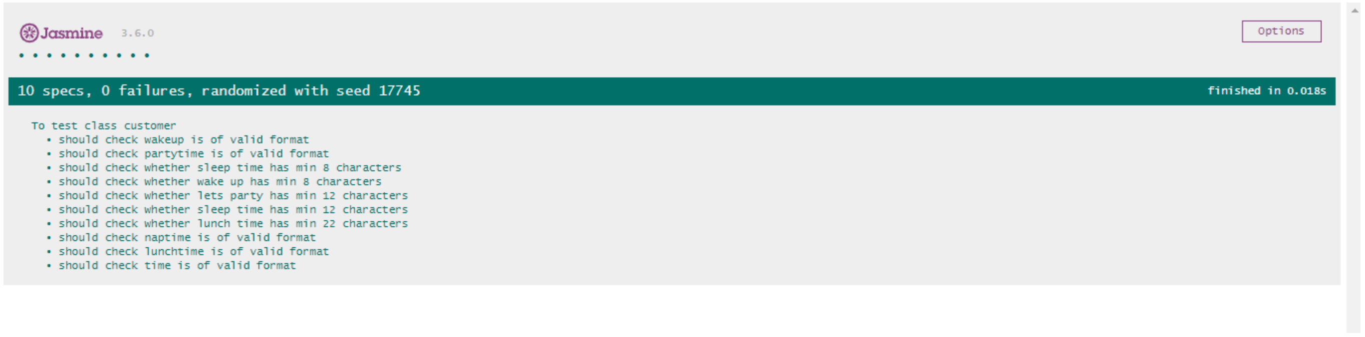


Figure 7 Output



```
Jasmine 3.6.0                                                          Options
• • • • • • • • • •
10 specs, 0 failures, randomized with seed 17745                finished in 0.018s

  To test class customer
    • should check wakeup is of valid format
    • should check partytime is of valid format
    • should check whether sleep time has min 8 characters
    • should check whether wake up has min 8 characters
    • should check whether lets party has min 12 characters
    • should check whether sleep time has min 12 characters
    • should check whether lunch time has min 22 characters
    • should check naptime is of valid format
    • should check lunchtime is of valid format
    • should check time is of valid format
```

Figure 8 Jasmine framework output

Video Summary

The video includes a small look into our miniproject workflow, first the code is shown followed by its working in chrome web browser. Here the user is able to see the current Indian time and along with that user can set a time or alarm as sleep/lunch/wake up and party time. Whenever the time reaches their set by the user the display will be changed.

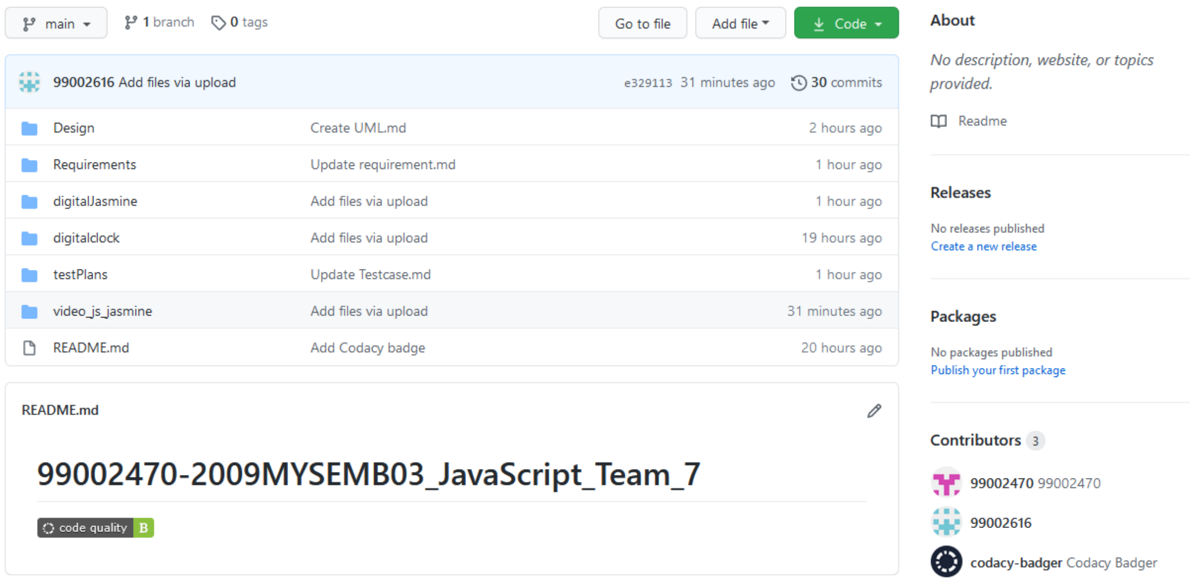
Video link

[Video Link]

Github link

[Github JavaScript link]

Git Dashboard



main 1 branch 0 tags Go to file Add file Code

99002616 Add files via upload e329113 31 minutes ago 30 commits

Design	Create UML.md	2 hours ago
Requirements	Update requirement.md	1 hour ago
digitalJasmine	Add files via upload	1 hour ago
digitalclock	Add files via upload	19 hours ago
testPlans	Update Testcase.md	1 hour ago
video_js_jasmine	Add files via upload	31 minutes ago
README.md	Add Codacy badge	20 hours ago

README.md

99002470-2009MYSEMB03_JavaScript_Team_7

code quality B

About

No description, website, or topics provided.

Readme

Releases

No releases published
Create a new release

Packages

No packages published
Publish your first package

Contributors 3

99002470 99002470
99002616
codacy-badger Codacy Badger

Figure 9 Git Dashboard

Summary

1. Developed a web page for displaying a digital alarm clock using JS, CSS and HTML file.
2. The unit test has been done using Jasmine Framework

Challenges faced and how were they overcome

Challenge 1: Jasmine Framework.
Solution: The JS file developed in this project was not able to incorporate in the jasmine framework. Therefore, we have created a separate JS file for jasmine framework and the unit test cases has passed.

Future Scope (If applicable)

- An automatic buzzer can be implemented if possible for the digital alarm clock.
- It can be further modified it to a digital fitness band
- It can also be used as daily routine chart helper

Miniproject -3(Advanced Python programming) [Team]

Modules

Modules linked to the miniproject–SDLC and Advanced Python programming.

Topic and Subtopics

Core Topic:

An application of python programming language.

Sub Topics:

A patient record monitoring using python

Objectives & Requirements

Objective:

1. To develop an application of python language for making a patient health record monitoring.
2. Understanding python programming and working with multi-file in python and building them.
3. Understanding with the file operations.

Requirements (High level and low level):

ID	Description
HL_01_L_01	Low level 01 – current production version of Python High level 01 –python compatibility with notepad++
HL_02_L_02	Low level 02 – command prompt and python shell High level 02 –Able to add new patient details
HL_03_L_03	Low level 03 – command prompt and python shell High level 03 – Able to edit existing patient details
HL_04_L_04	Low level 04– command prompt and python shell High level 04 – Able to search for existing patient details as requested
HL_05_L_05	Low level 05 – command prompt and python shell High level 05 – Able to display existing patient details as requested
HL_06_L_06	Low Level 06 – command prompt and python shell High Level 06 – Able to exit from the log on demand

Table 10 High level and low level requirement

Design

Behavioral Diagram (UseCase Diagram):

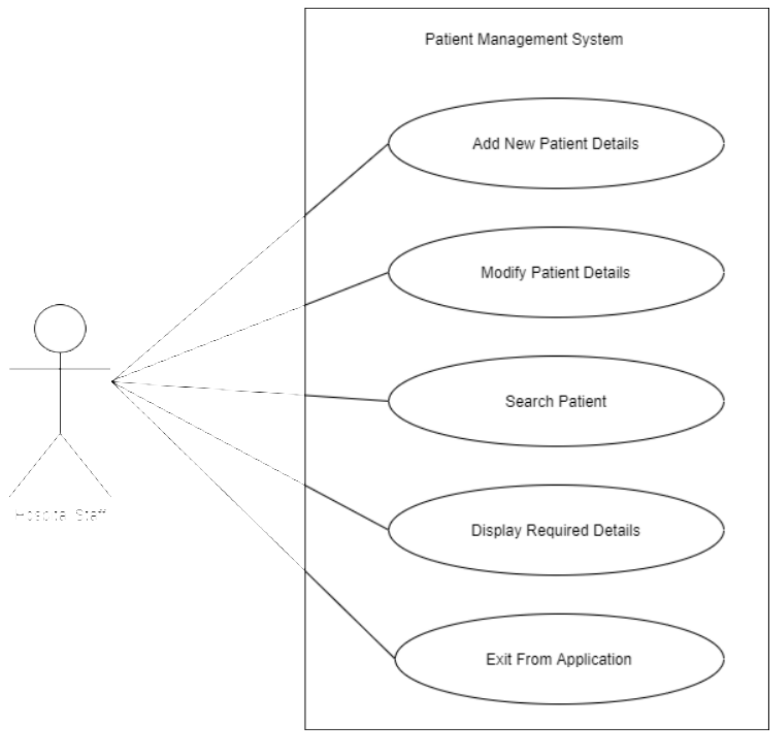


Figure 10 Behavioral Diagram

Structural Diagram (Component Diagram):

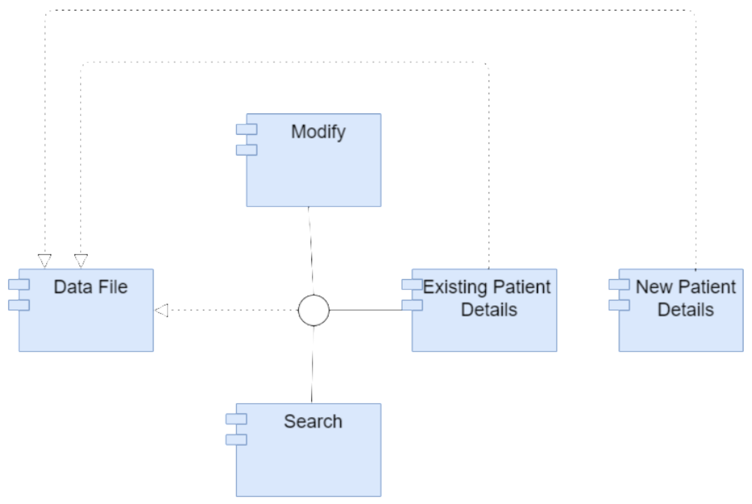


Figure 11 Structural Diagram

Test Plan

Unit Level Test Cases:

ID	Description	Precondition	Excepted input	Expected output	Actual output
Tc1	Check whether python environment is set, is compatible and the path is set.	current production version of Python	Setting up the path	Indication in command window	Indication in command window
Tc2	Check whether “add to patient list” function is executing.	Set required environment variables and path with current version of python.	Corresponding ID and Adding patient information to the list	List should be updated successfully	List has been updated successfully
Tc3	Check whether “search in existing patient list” function is executing.	Set required environment variables and path with current version of python.	Corresponding ID and Search for a patient with name	Display details of the requested patient	Displayed details of the requested patient
Tc4	Check whether “Display all existing patient list” function is executing.	Set required environment variables and path with current version of python.	Corresponding ID to display the patient details	Display details of the all the patients in the list	Displayed details of the all the patients in the list
Tc5	Check whether “modify a corresponding patient details” function is executing.	Set required environment variables and path with current version of python.	Corresponding ID and patient full name to modify	Able to modify existing details.	Existing details can be modified.
Tc6	Check whether ‘Exit’ function is executed.	Set required environment variables and path with current version of python.	Corresponding ID to exit from the log	Exit the application.	Exiting the application.

Table 11 Test Plan

Integration level Test cases:

Tc1	Check whether all the functions in unit test plan when integrated are executing correctly.	Set required environment variables and path with current version of python.	Corresponding ID to details	All functions executing correctly.	All functions executing correctly.

Table 12 Integration level Test cases

Implementation Summary

An application for Patient health record monitoring with python programming language is developed. Here the user is able to create a new patient detail contains Name. Address, Disease status and ID. The new list gets updated on adding data. This data can be modified, the list can be displayed and a search action can be performed by giving corresponding Full name of the patient.

These actions are performed with the help of corresponding ID for each functions.

Enter 1. To Add Contacts

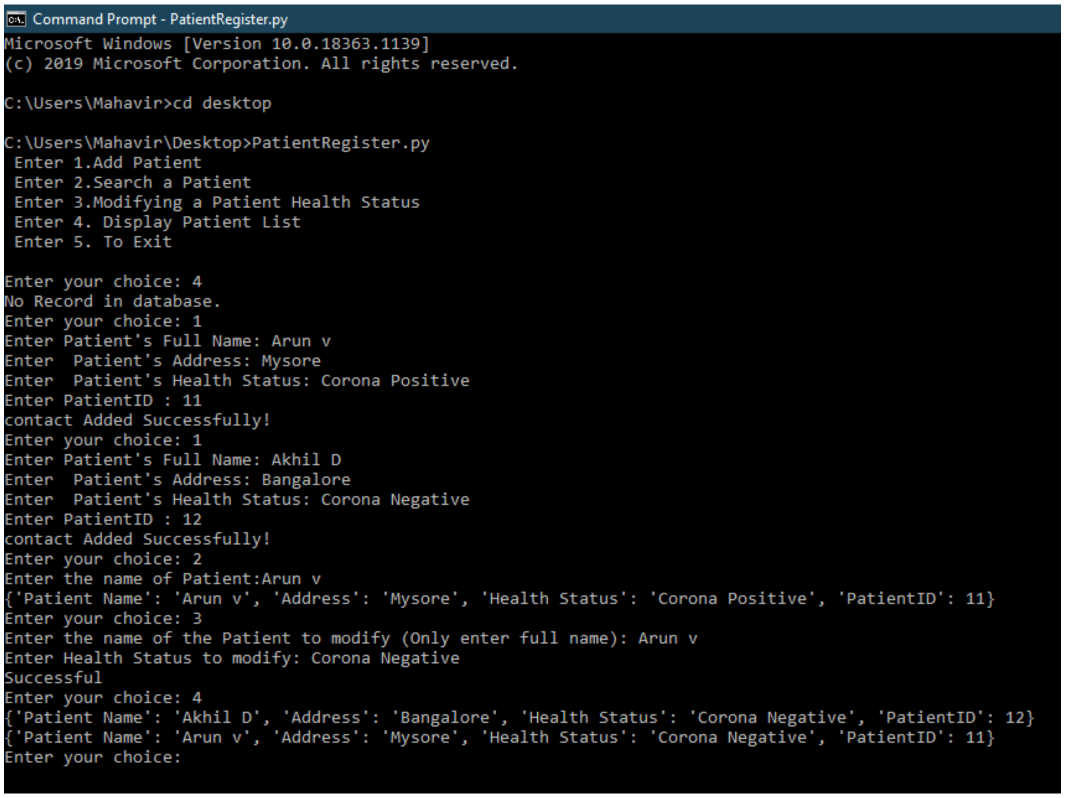
Enter 2. For Searching a Contact

Enter 3. For Modifying a Contact

Enter 4. To Display Contacts

Enter 5. To Exit

Output:



```
Command Prompt - PatientRegister.py
Microsoft Windows [Version 10.0.18363.1139]
(c) 2019 Microsoft Corporation. All rights reserved.

C:\Users\Mahavir>cd desktop

C:\Users\Mahavir\Desktop>PatientRegister.py
 Enter 1.Add Patient
 Enter 2.Search a Patient
 Enter 3.Modifying a Patient Health Status
 Enter 4. Display Patient List
 Enter 5. To Exit

Enter your choice: 4
No Record in database.
Enter your choice: 1
Enter Patient's Full Name: Arun v
Enter  Patient's Address: Mysore
Enter  Patient's Health Status: Corona Positive
Enter PatientID : 11
contact Added Successfully!
Enter your choice: 1
Enter Patient's Full Name: Akhil D
Enter  Patient's Address: Bangalore
Enter  Patient's Health Status: Corona Negative
Enter PatientID : 12
contact Added Successfully!
Enter your choice: 2
Enter the name of Patient:Arun v
{'Patient Name': 'Arun v', 'Address': 'Mysore', 'Health Status': 'Corona Positive', 'PatientID': 11}
Enter your choice: 3
Enter the name of the Patient to modify (Only enter full name): Arun v
Enter Health Status to modify: Corona Negative
Successful
Enter your choice: 4
{'Patient Name': 'Akhil D', 'Address': 'Bangalore', 'Health Status': 'Corona Negative', 'PatientID': 12}
{'Patient Name': 'Arun v', 'Address': 'Mysore', 'Health Status': 'Corona Negative', 'PatientID': 11}
Enter your choice:
```

Figure 12 Python console output

Video Summary

The video includes a small look into our miniproject workflow, first the code is shown followed by its working in command prompt. Here the user is able to create a new patient detail contains Name. Address, Disease status and ID. The new list gets updated on adding data. This data can be modified, the list can be displayed and a search action can be performed by giving corresponding Full name of the patient.

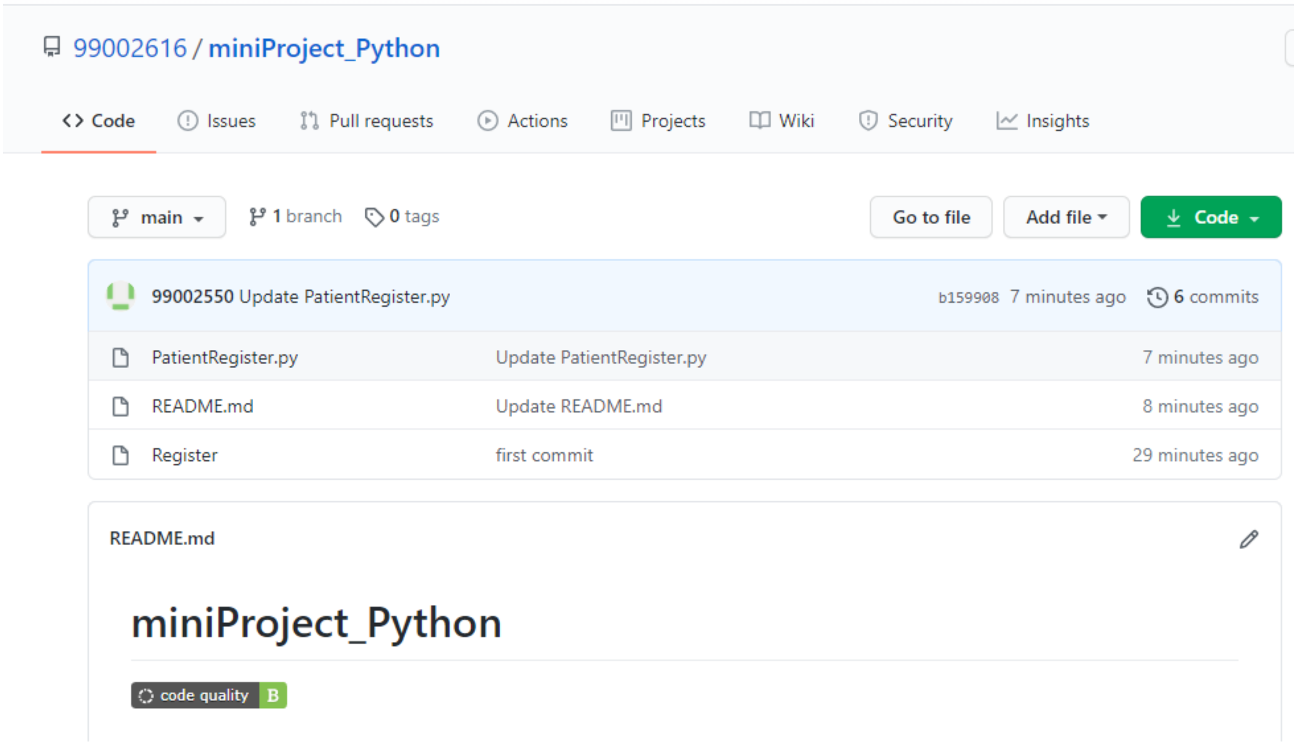
Video Link

[Video Link]

Git Link

[Git Repo]

Git Dashboard



99002550 Update PatientRegister.py b159908 7 minutes ago 6 commits

PatientRegister.py	Update PatientRegister.py	7 minutes ago
README.md	Update README.md	8 minutes ago
Register	first commit	29 minutes ago

README.md

miniProject_Python

code quality B

Figure 13 Git Repo Screenshot

Summary

An application for Patient health record monitoring with python programming language is developed. Here the user is able to create a new patient detail contains Name. Address, Disease status and ID. The new list gets updated on adding data. This data can be modified, the list can be displayed and a search action can be performed by giving corresponding Full name of the patient.

These actions are performed with the help of corresponding ID for each functions.

Enter 1. To Add Contacts
Enter 2. For Searching a Contact
Enter 3. For Modifying a Contact
Enter 4. To Display Contacts
Enter 5. To Exit

Individual Contribution & Highlights

S.No	Name	Contributions
1	Neema Zacharias (99002557)	Done with coding. Documentation of report and ppt.
2	Shahna S.S (99002550)	Done with coding. Documentation of report and ppt.
3	Raj Sekhar Mishra (99002616)	Done with coding, git repository creation, readme file creation and code quality checking. Documentation of report and ppt.

Table 13 Individual Contribution & Highlights

Challenges faced and how were they overcome.

Challenge 1: Indentation issues.
Solution: Use 4-space indents and avoiding all hard tab characters.

Challenge 2: Cross import of Modules.
Solution: Doing a selective import only in the functions where it is needed.