./

GENESIS - Learning Outcome & Mini-project Summary Report



|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **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 |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |

**Details**

Contents

[Contents 3](#_Toc56002869)

[Miniproject -1(Web Automation and testing using Java based Selenium and Cucumber) [Team] 5](#_Toc56002870)

[**Modules** 5](#_Toc56002871)

[Topic and Subtopics 5](#_Toc56002872)

[**Objectives & Requirements** 5](#_Toc56002873)

[**Design** 6](#_Toc56002874)

[**Test Plan** 7](#_Toc56002875)

[**Implementation Summary** 9](#_Toc56002876)

[Video Summary 10](#_Toc56002877)

[Git Link 10](#_Toc56002878)

[Git Dashboard 10](#_Toc56002879)

[**Individual Contribution & Highlights** 10](#_Toc56002880)

[Challenges faced and how were they overcome 10](#_Toc56002881)

[Mini-Project -2 Java Script and Jasmine framework[Team] 11](#_Toc56002882)

[Modules 11](#_Toc56002883)

[**Topic and Subtopics** 11](#_Toc56002884)

[**Objectives & Requirements** 11](#_Toc56002885)

[Objectives: 11](#_Toc56002886)

[High level requirement: 11](#_Toc56002887)

[Low Level Requirement: 11](#_Toc56002888)

[**Design** 12](#_Toc56002889)

[Structural Diagram 12](#_Toc56002890)

[Behavioral Diagram 13](#_Toc56002891)

[**Test Plan** 13](#_Toc56002892)

[Unit level test case 13](#_Toc56002893)

[Low Level Test Plan 14](#_Toc56002894)

[**Individual Contribution & Highlights** 16](#_Toc56002895)

[**Implementation Summary** 16](#_Toc56002896)

[**Output** 17](#_Toc56002897)

[Video Summary 17](#_Toc56002898)

[Video link 18](#_Toc56002899)

[**Github link** 18](#_Toc56002900)

[Git Dashboard 18](#_Toc56002901)

[**Summary** 18](#_Toc56002902)

[**Challenges faced and how were they overcome** 18](#_Toc56002903)

[**Future Scope (If applicable)** 18](#_Toc56002904)

[Miniproject -3(Advanced Python programming) [Team] 19](#_Toc56002905)

[**Modules** 19](#_Toc56002906)

[Topic and Subtopics 19](#_Toc56002907)

[**Objectives & Requirements** 19](#_Toc56002908)

[**Design** 20](#_Toc56002909)

[**Test Plan** 21](#_Toc56002910)

[**Implementation Summary** 22](#_Toc56002911)

[Video Summary 23](#_Toc56002912)

[Video Link 23](#_Toc56002913)

[Git Link 23](#_Toc56002914)

[Git Dashboard 23](#_Toc56002915)

[Summary 24](#_Toc56002916)

[**Individual Contribution & Highlights** 24](#_Toc56002917)

[**Challenges faced and how were they overcome.** 24](#_Toc56002918)

**Table of Figures:**

[Figure 1 (Behavioral Diagram) 6](#_Toc56002919)

[Figure 2 Structural Diagram 7](#_Toc56002920)

[Figure 3 Output Eclipse 9](#_Toc56002921)

[Figure 4 Git Dashboard 10](#_Toc56002922)

[Figure 5 Structural Diagram 12](#_Toc56002923)

[Figure 6 Behavioral Diagram 13](#_Toc56002924)

[Figure 7 Output 17](#_Toc56002925)

[Figure 8 Jasmine framework output 17](#_Toc56002926)

[Figure 9 Git Dashboard 18](#_Toc56002927)

[Figure 10 Behavioral Diagram 20](#_Toc56002928)

[Figure 11 Structural Diagram 20](#_Toc56002929)

[Figure 12 Python console output 22](#_Toc56002930)

[Figure 13 Git Repo Screenshot 23](#_Toc56002931)

**List of tables**:

[Table 1 (Requirements) 6](#_Toc56002932)

[Table 2 Unit Level Test Cases: 8](#_Toc56002933)

[Table 3 Integration level Test cases 9](#_Toc56002934)

[Table 4 Individual Contribution & Highlights 10](#_Toc56002935)

[Table 5 High level requirement 11](#_Toc56002936)

[Table 6 Low Level Requirements 12](#_Toc56002937)

[Table 7 High level Requirements 14](#_Toc56002938)

[Table 8 Low level requirements 16](#_Toc56002939)

[Table 9 Individual Contribution & Highlights 16](#_Toc56002940)

[Table 10 High level and low level requirement 19](#_Toc56002941)

[Table 11 Test Plan 21](#_Toc56002942)

[Table 12 Integration level Test cases 22](#_Toc56002943)

[Table 13 Individual Contribution & Highlights 24](#_Toc56002944)

# 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

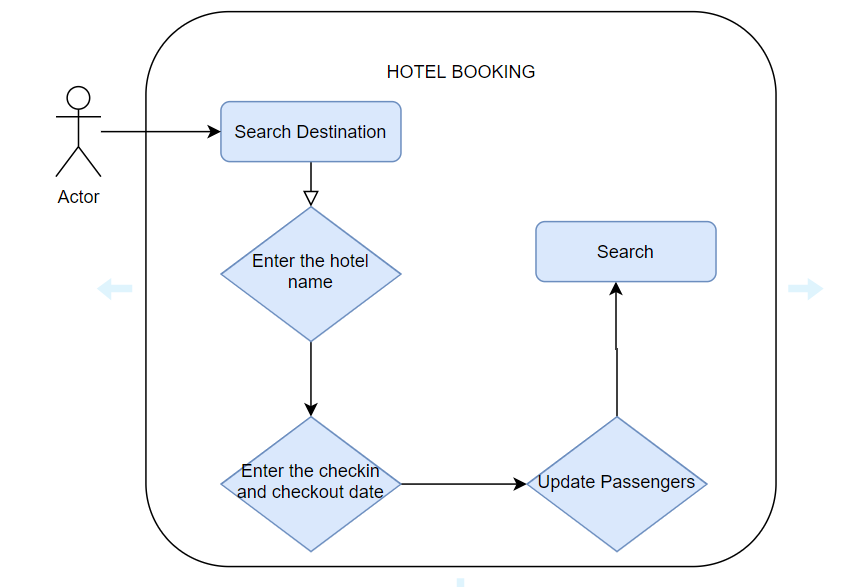


Figure 1 (Behavioral Diagram)

Structural Diagram

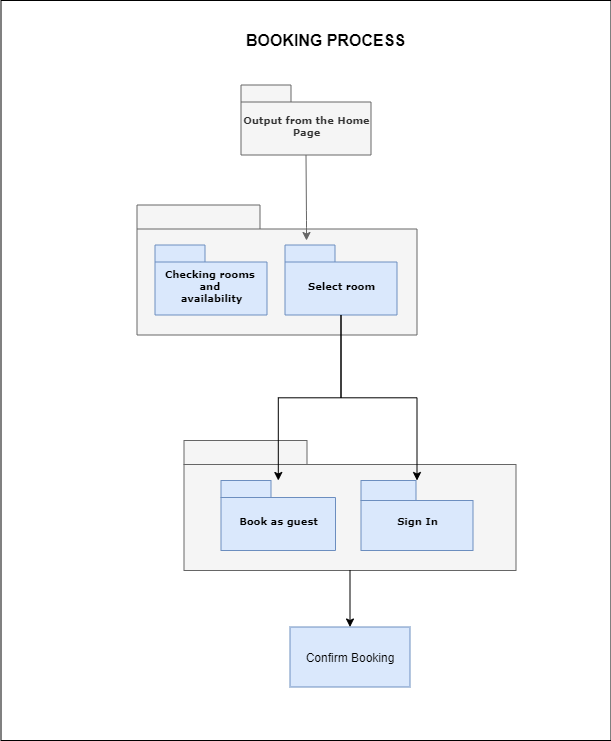


Figure 2 Structural Diagram

## **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:

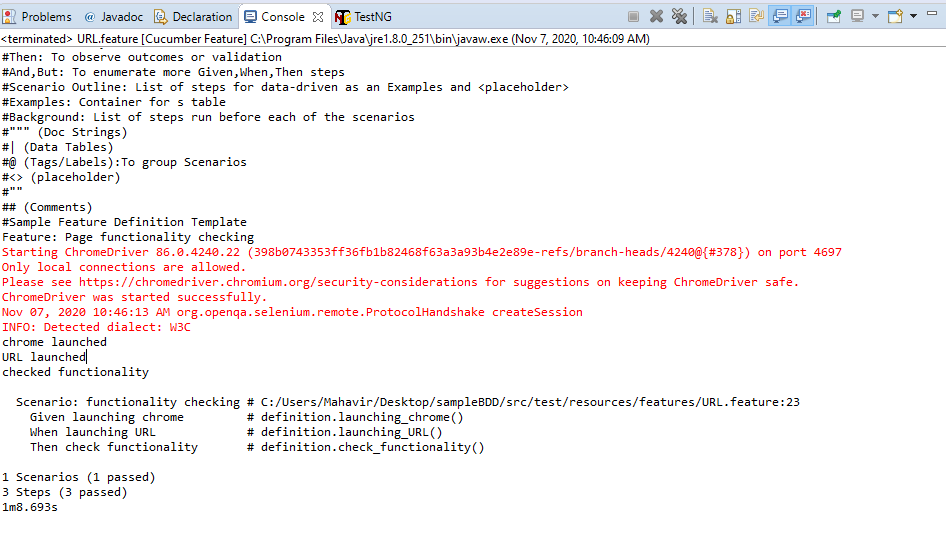


Figure 3 Output Eclipse

### Video Summary

[Video Link](https://github.com/99002616/2009MYSEMB03_Java-Selenium_TEAM_06/tree/main/Video)

### Git Link

[Git Repository](https://github.com/99002616/2009MYSEMB03_Java-Selenium_TEAM_06)

### Git Dashboard

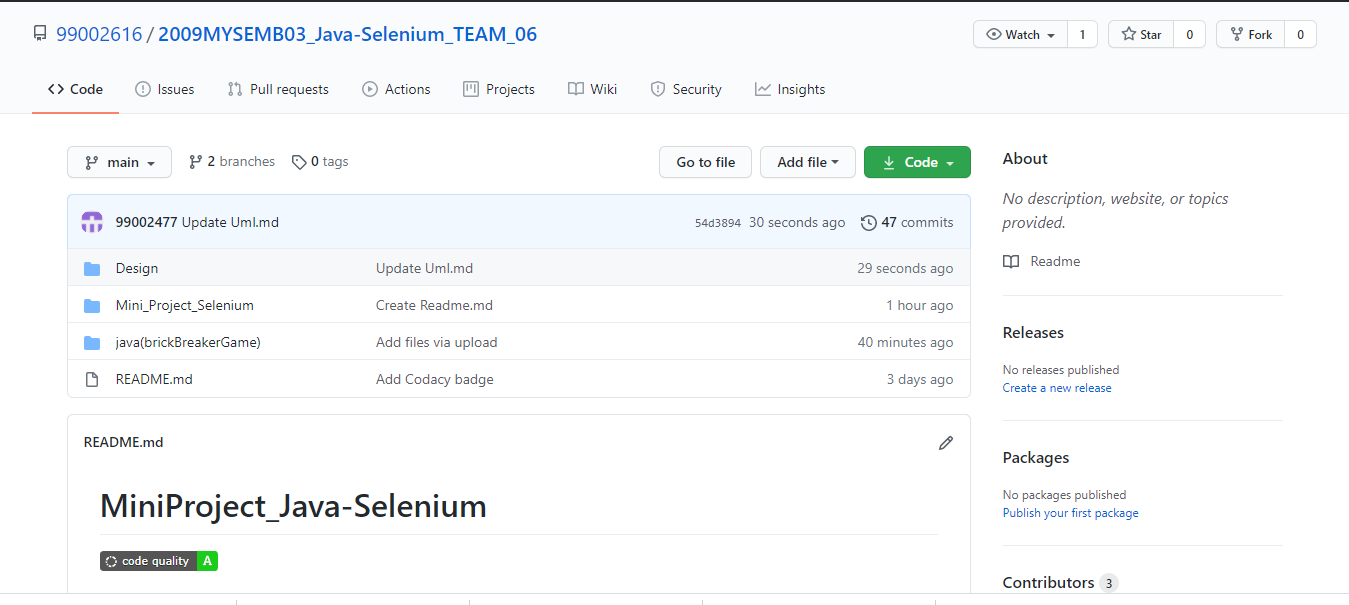


Figure 4 Git Dashboard

## **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

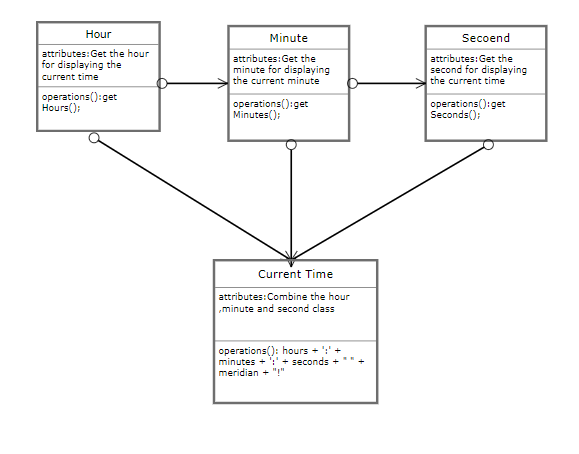


Figure 5 Structural Diagram

### Behavioral Diagram

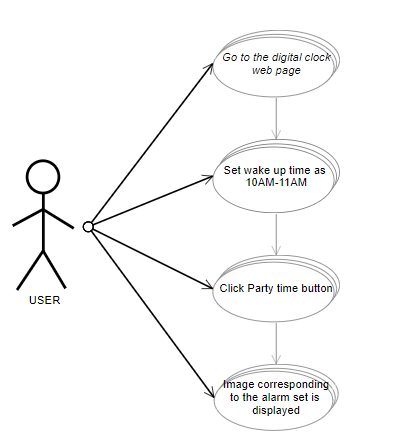


Figure 6 Behavioral Diagram

## **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 alarm time | 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 |
| 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

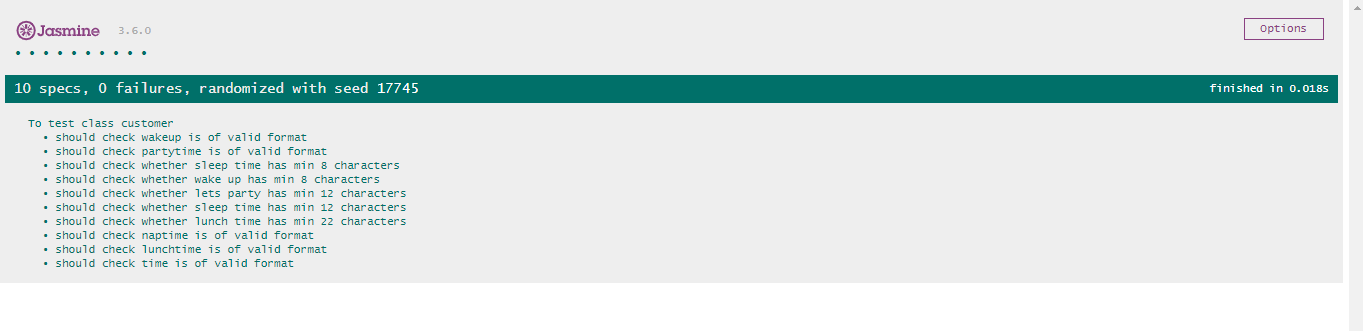


Figure 8 Jasmine framework output

### 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](https://github.com/99002470/2009MYSEMB03_JavaScript_Team_7/tree/main/video)

## **Github link**

[Github JavaScript link](https://github.com/99002470/99002470-2009MYSEMB03_JavaScript_Team_7)

### Git Dashboard

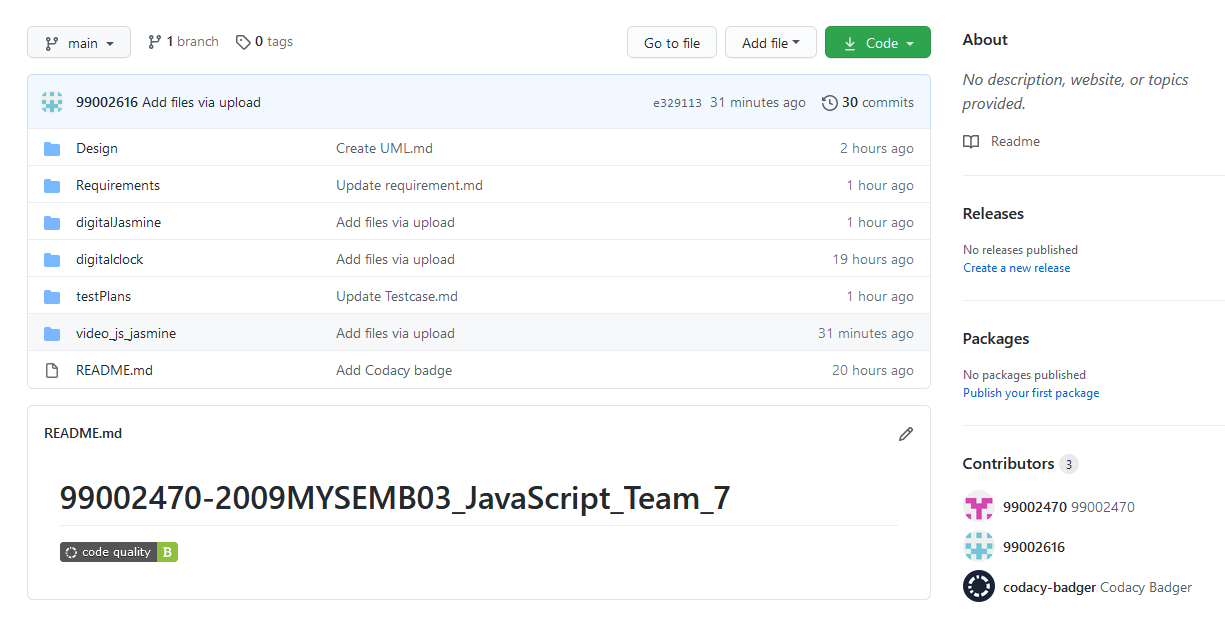


Figure 9 Git Dashboard

## **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 | 1. Low level 05 – command prompt and python shell 2. High level 05 – Able to display existing patient details as requested |
| HL\_06\_L\_06 | 1. Low Level 06 – command prompt and python shell 2. High Level 06 – Able to exit from the log on demand |

Table 10 High level and low level requirement

## **Design**

Behavioral Diagram ( UseCase Diagram):

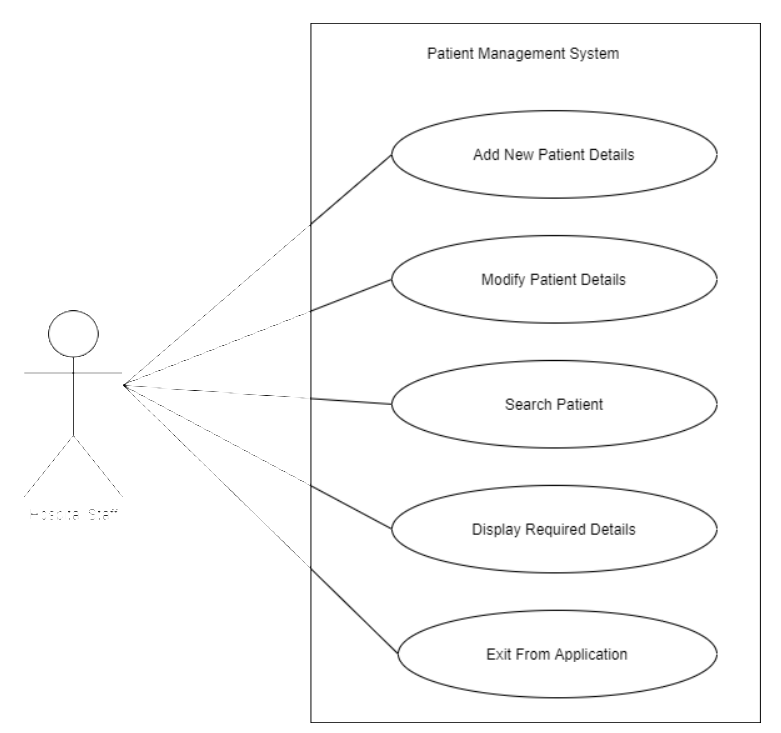


Figure 10 Behavioral Diagram

Structural Diagram (Component Diagram):

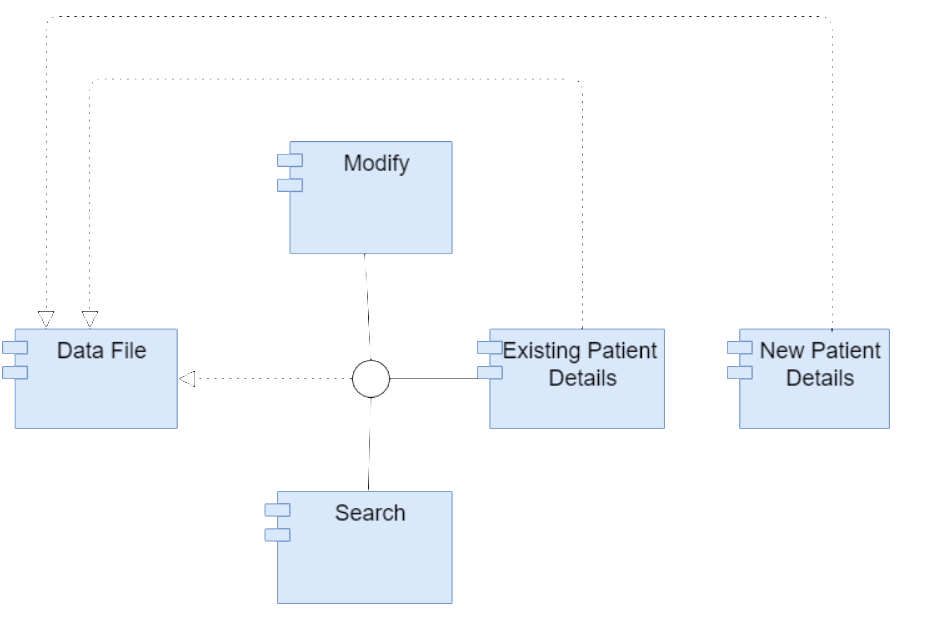


Figure 11 Structural Diagram

## **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:**

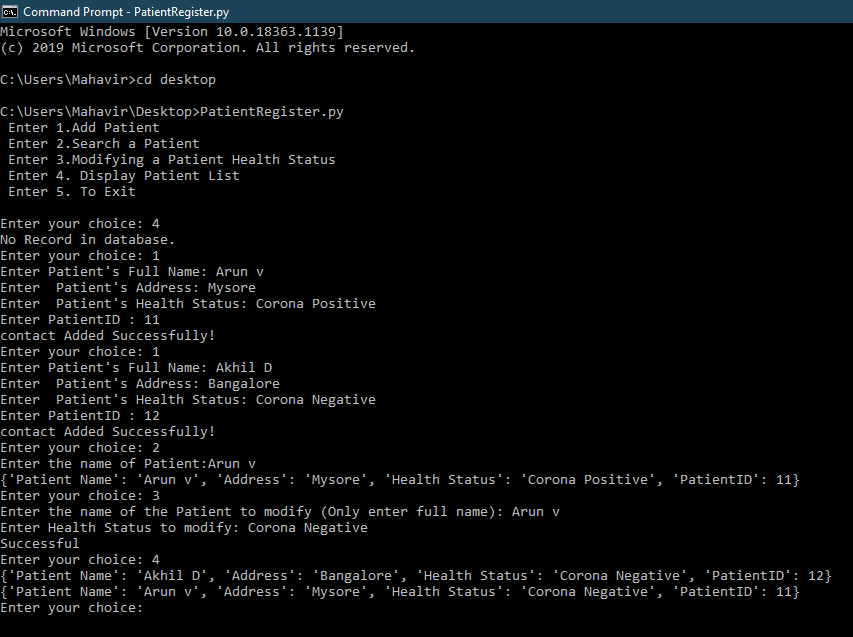


Figure 12 Python console output

### 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](https://github.com/99002616/2009MYSEMB03_Python_Team_5/tree/main/Video)

### Git Link

[Git Repo](https://github.com/99002616/2009MYSEMB03_Python_Team_5.git)

### Git Dashboard

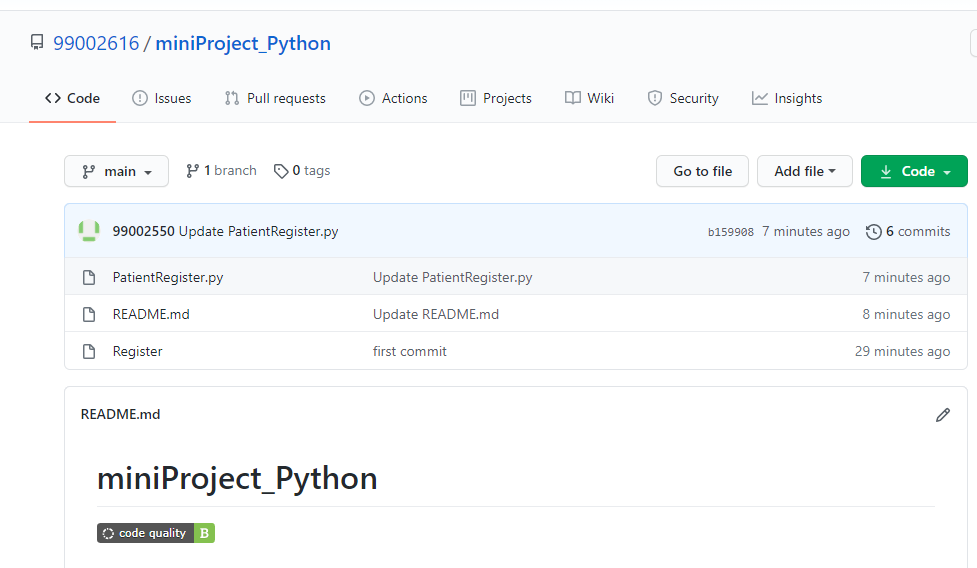


Figure 13 Git Repo Screenshot

### 

### 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.