NANYANG
TECHNOLOGICAL
UNIVERSITY
SINGAPORE

Test Cases and Requirements Test Coverage Report For TimeWise

Lab group : SSP7
Group : Team 0

Group members	**Roles**
Mantri Raghav	Project Manager
Dwivedee Lakshyajeet	DEV Lead
Harding James	DEV Back-End
Alex Bernini	DEV Front-End
Mittal Madhav	Release Manager
Xue Xueting	QA Engineer
Koh Hui Ling	QA Engineer
Lek Jie Ling	QA Manager

Version History

Version #	Implemented By	Revision Date	Approved By	Approval Date	Reason
1.0	Raghav Mantri	18/3/2020	Laksh	18/3/2020	Initial test report draft
1.1	Xue Xueting	21/03/2020	Raghav	23/3/2020	Updates on document
1.2	Hui Ling	22/3/2020	Madhav	23/3/2020	Updates on document
1.3	Dwivedee Lakshyajee	23/3/2020	Raghav	23/3/2020	Finalising and approval

Table of Contents

Unit Testing

Test Case 1

Test Case #: 1	Test Name: AddButton.js
System: TimeWise	Sub-System: Task Page
Designed By: Quality Assurance Team	Design Date: 15 March 2020
Executed By: Quality Engineer	Execution Date: 16 March 2020
Short Description: Testing the opening of modal for adding a task	

Pre-Conditions	
1	User must have an internet connection
2	User must be logged in

Step	**Action**	**Expected System Response**
1	Click on the "+" icon	A modal opens up requesting the user to add task details

Post-Conditions	
1	A modal to add tasks is displayed on the page

Test Case 2

Test Case #: 2	Test Name: MenuButton.js
System: TimeWise	Sub-System: Task Page
Designed By: Quality Assurance Team	Design Date: 15 March 2020
Executed By: Quality Engineer	Execution Date: 16 March 2020
Short Description: Testing the opening of menu sidebar	

Pre-Conditions	
1	User must have an internet connection
2	User must be logged in

Step	**Action**	**Expected System Response**
1	Click on the hamburger menu icon in the navbar	A sidebar opens up from the left which shows the options of logging out and refreshing schedule

Post-Conditions	
1	The menu sidebar is opened up on the left side

Test Case 3

Test Case #: 3	Test Name: CompleteTask.js
System: TimeWise	Sub-System: Task Page
Designed By: Quality Assurance Team	Design Date: 15 March 2020
Executed By: Quality Engineer	Execution Date: 16 March 2020
Short Description: Testing the completion of tasks	

Pre-Conditions	
1	User must have an internet connection
2	User must be logged in

Step	**Action**	**Expected System Response**
1	Click on the radio button next to the task name	The task must get removed from the list of tasks

Post-Conditions	
1	The task must get removed from the list of tasks

Test Case 4

Test Case #: 4	Test Name: DeleteTask.js
System: TimeWise	Sub-System: Task Information Modal
Designed By: Quality Assurance Team	Design Date: 15 March 2020
Executed By: Quality Engineer	Execution Date: 16 March 2020
Short Description: Testing the deletion of task	

Pre-Conditions	
1	User must have an internet connection
2	User must be logged in

Step	**Action**	**Expected System Response**
1	Click on Trash icon	- The task must be deleted from the list of tasks - The task information modal must be closed

Post-Conditions	
1	The respective task must get deleted from the list of tasks
2	The project information modal must be closed

Integration Testing

Test Case 1

Test Case #: 1	Test Name: Login.js
System: TimeWise	Sub-System: Login Page
Designed By: Quality Assurance Team	Design Date: 15 March 2020
Executed By: Quality Engineer	Execution Date: 16 March 2020
Short Description: Testing logging in	

Pre-Conditions	
1	User must have an internet connection
2	User must be on the login page

Step	**Action**	**Expected System Response**
1	Enter NTU Email and Password	N/A
2	Click on Login button	If credentials are correct: ● Retrieves time table If user credentials are incorrect: ● The timetable is not retrieved

Post-Conditions	
1	User sees the actual timetable retrieved as tasks

Test Case 2

Test Case #:2	Test Name: TaskDetail.js
System: TimeWise	Sub-System: Task Page
Designed By: Quality Assurance Team	Design Date: 15 March 2020
Executed By: Quality Engineer	Execution Date: 16 March 2020
Short Description: Testing the opening of task description	

Pre-Conditions	
1	User must have an internet connection
2	User must be logged in

Step	**Action**	**Expected System Response**
1	Tap on a task	Shows details about the task

Post-Conditions	
●	Details about the task (type, location, date time) are shown

Test Case 3

Test Case #: 3	Test Name: AddTask.js
System: TimeWise	Sub-System: Add Task Modal
Designed By: Quality Assurance Team	Design Date: 15 March 2020
Executed By: Quality Engineer	Execution Date: 16 March 2020
Short Description: Testing the addition of a new task	

Pre-Conditions	
1	User must have an internet connection
2	User must be logged in

Step	**Action**	**Expected System Response**
1	Tap on the ‘+’ icon next to the name of the day	Shows dialog for creating a new task
2	Enter name of task	Successfully create a name for a task
3	Either choose a recommended time slot or select time manually	See time entered for the task
4	Click on submit button	Creates a task

Post-Conditions	
1	Creates task for that day with the given name, time and category ‘Task’

Test Case 4

Test Case #:4	Test Name: EditTask.js
System: TimeWise	Sub-System: Edit Task Modal
Designed By: Quality Assurance Team	Design Date: 15 March 2020
Executed By: Quality Engineer	Execution Date: 16 March 2020
Short Description: Testing the modal to edit tasks	

Pre-Conditions	
1	User must have an internet connection
2	User must be logged in

Step	Action	Expected System Response
1	Tap on a task	Shows details about the task
2	Tap on the ‘pencil’ icon (edit)	Opens options to edit the task

Post-Conditions	
1	Opens a pop-up to edit details - location, time

Test Case 5

Test Case #: 5	Test Name: DeleteTask.js
System: TimeWise	Sub-System: Task Description Modal
Designed By: Quality Assurance Team	Design Date: 15 March 2020
Executed By: Quality Engineer	Execution Date: 16 March 2020
Short Description: Testing the deletion of a task	

Pre-Conditions	
1	User must have an internet connection
2	User must be logged in

Step	**Action**	**Expected System Response**
1	Tap on a task	Shows details about the task
2	Tap on the delete icon	Task should be deleted

Post-Conditions	
1	Task is removed from the list

Test Case 6

Test Case #: 6	Test Name: RefreshSchedule.js
System: TimeWise	Sub-System: Menu Sidebar
Designed By: Quality Assurance Team	Design Date: 15 March 2020
Executed By: Quality Engineer	Execution Date: 16 March 2020
Short Description: Testing the refreshing of the schedule	

Pre-Conditions	
1	User must have an internet connection
2	User must be logged in

Step	**Action**	**Expected System Response**
1	Tap on menu icon	Displays the hamburger menu
2	Click on “Refresh Schedule”	Reopens tasks with updated classes

Post-Conditions	
1	Tasks page is opened with updated tasks

Test Case 7

Test Case #: 7	Test Name: Logout.js
System: TimeWise	Sub-System: Menu Sidebar
Designed By: Quality Assurance Team	Design Date: 15 March 2020
Executed By: Quality Engineer	Execution Date: 16 March 2020
Short Description: Testing the logout button	

Pre-Conditions	
1	User must have an internet connection
2	User must be logged in

Step	**Action**	**Expected System Response**
1	Click on “Logout” button	Logout successfully

Post-Conditions	
1	Logout is successful, user gets back to the opening app screen

Test Case 8

Test Case #: 8	Test Name: Notification.js
System: TimeWise	Sub-System: Notification
Designed By: Quality Assurance Team	Design Date: 15 March 2020
Executed By: Quality Engineer	Execution Date: 16 March 2020
Short Description: Testing system to notify user of upcoming tasks	

Pre-Conditions	
1	Have a Student account
2	Logged in at least once
3	Added a upcoming task
4	Notification permissions are granted for TimeWise

Step	**Action**	**Expected System Response**
1	Wait for task to be within 20 minutes from now	A notification about the upcoming task must pop up

Post-Conditions	
1	Get a notification of an upcoming task

System Testing

Test Case 1

Test Case #: 1	Test Name: Main.js
System: TimeWise	Sub-System: N/A
Designed By: Quality Assurance Team	Design Date: 15 March 2020
Executed By: Quality Engineer	Execution Date: 16 March 2020
Short Description: Testing entire system	

Pre-Conditions	
1	Have a Student account
2	User opens TimeWise application

Step	**Action**	**Expected System Response**
1	Clicking on respective buttons	The functionalities work as expected based on the integration and unit tests done previously

Post-Conditions	
1	Desired action occurs based on the user’s actions

Summary

Test case	Pre-conditions	Test Steps	Test data	Expected result	Actual Result	Status
To login Successfully	1.Have a Student account 2.In Login Page	1.Enter NTU Email 2.Enter password	Username, password	Retrieves time table	Shows actual timetable	Pass
Invalid login credentials	1.Have a student account 2.In login page	1.Enter valid(or invalid) NTU email 2.Enter invalid(or valid) password	Random/invalid username and/or random/invalid password	Displays error message	Displays error message	Pass
Click on task in timetable view	1.Login successfully 2.Scrape timetable successfully	1.Tap on a task	Timetable task	Shows details about the task	Details about the task (type, location, date time) are shown	Pass
Creating a task	1.Login Successfully	1. Tap on the '+' icon next to the name of the day 2. Enter name of task 3. Either choose a recommended time slot or select time manually 4. Click on submit button	Task name Task time	Creates task	Creates task for that day with the given name, time and category 'Task'	Pass
Update a task	1.Login Successfully	Tap on a task Tap on the pencil (edit) icon	Timetable task	Opens options to edit	Opens a pop-up to edit details - location, time	Pass
Delete a task	1.Login Successfully	1.Tap on a task 2.Tap on the delete icon	Timetable task	Task deleted successfully	Task is removed from the list	Pass
Refresh Schedule	1.Login Successfully	1. Tap on menu icon 2. Click on "Refresh Schedule"	Username, password	Reopen tasks page with updated classes	Tasks page is opened with updated tasks	Pass
Logout	1. Already Log in	1.Click on "Logout" button	-	Click on "Logout" button	Logout successfully	Pass
Notification	Logged in at least once	-	List of tasks/classes	Get a notification of upcoming tasks	Notification not sent	Fail