

Indian Institute of Information Technology, Allahabad

Software Engineering

Software Design Specification



Group Members

Enrollment No.

Name

IIT2019228	-	Anirudh Gupta
IIT2019235	-	Vishwam Shriram Mundada
IIT2019238	-	Chandramani Kumar
IIB2019028	-	Harsh Kedia

TABLE OF CONTENT

1. Introduction.....	3
1.1 Purpose of this document.....	3
1.2 Scope of the development project.....	4
1.3 Definitions, acronyms, and abbreviations.....	4
1.4 References.....	4
1.5 Overview of document.....	4
2. Logical Architecture (Class Diagram, Sequence Diagram, State Diagram).....	5
2.1 Logical Architecture Description.....	5
2.2 Class name: Notification.....	9
2.3 Class Name: Activity.....	10
2.4 Class Name: UI.....	11
2.5 Class Name: Block.....	13
2.6 Class Name: Restriction.....	14
2.7 Class Name: Time Interval.....	15
2.8 Class Name: Tab.....	16
2.9 Class Name: Local Storage.....	17
3. Execution Architecture.....	18
3.1 Reuse and relationships to other products.....	18
4. Database Schema.....	19
4.1 Tables, Fields and Relationships.....	19
4.1.1 Databases.....	19
4.1.2 New Tables.....	19
4.1.2 New Field(s).....	19

1. Introduction

1.1 Purpose

This design will detail the implementation of the requirements as defined in the Software Requirements Specification .

1.2 Scope of the development project

We describe what features are in the scope of the software and what are not in the scope of the software to be developed.

In Scope:

- a. View total active time.
- b. View AFK time.
- c. Chart view showing time spent on a particular website.
- d. List of websites visited.

Out of Scope:

- a. Sites visited in incognito mode.
- b. Sites visited without being signed in on the browser.

1.3 Definitions, acronyms, and abbreviations

IEEE: Institute of Electrical and Electronics Engineers

SDS: Software Design Specification

AFK: Away From Keyboard

1.4 References

1.4.1 R. S. Pressman, Software Engineering: A Practitioner's Approach, 5th Ed, McGraw-Hill, 2001.

1.4.2 IEEE SDS template.

1.5 Overview of document

This SDS is divided into seven sections with various sub-sections. The sections of the Software Design Document are:

1. Introduction: describes about the document, purpose, scope of development project definitions and abbreviations used in the document.

2. Conceptual Architecture/Architecture Diagram: describes the overview of components, modules, structure and relationships and user interface issues.

3. Logical Architecture: describes Logical Architecture Description and Components.

4. Execution Architecture: defines the runtime environment, processes, deployment view.

5. Design Decisions and Trade-offs: describes the decisions taken along with the reason as to why they were chosen over other alternatives.

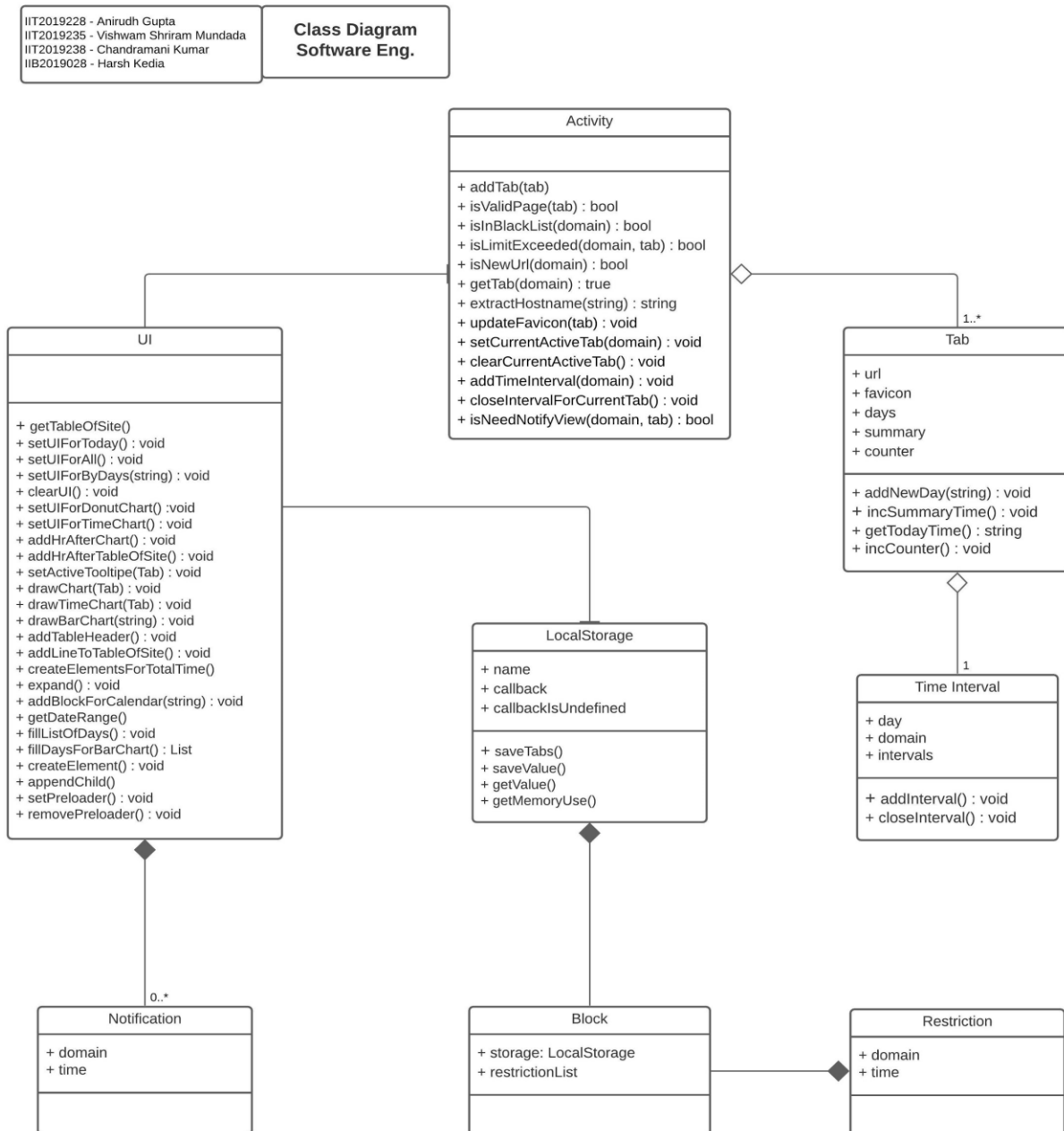
6. Pseudocode for components: describes pseudocode, as the name indicates.

7. Appendices: describes subsidiary matter if any.

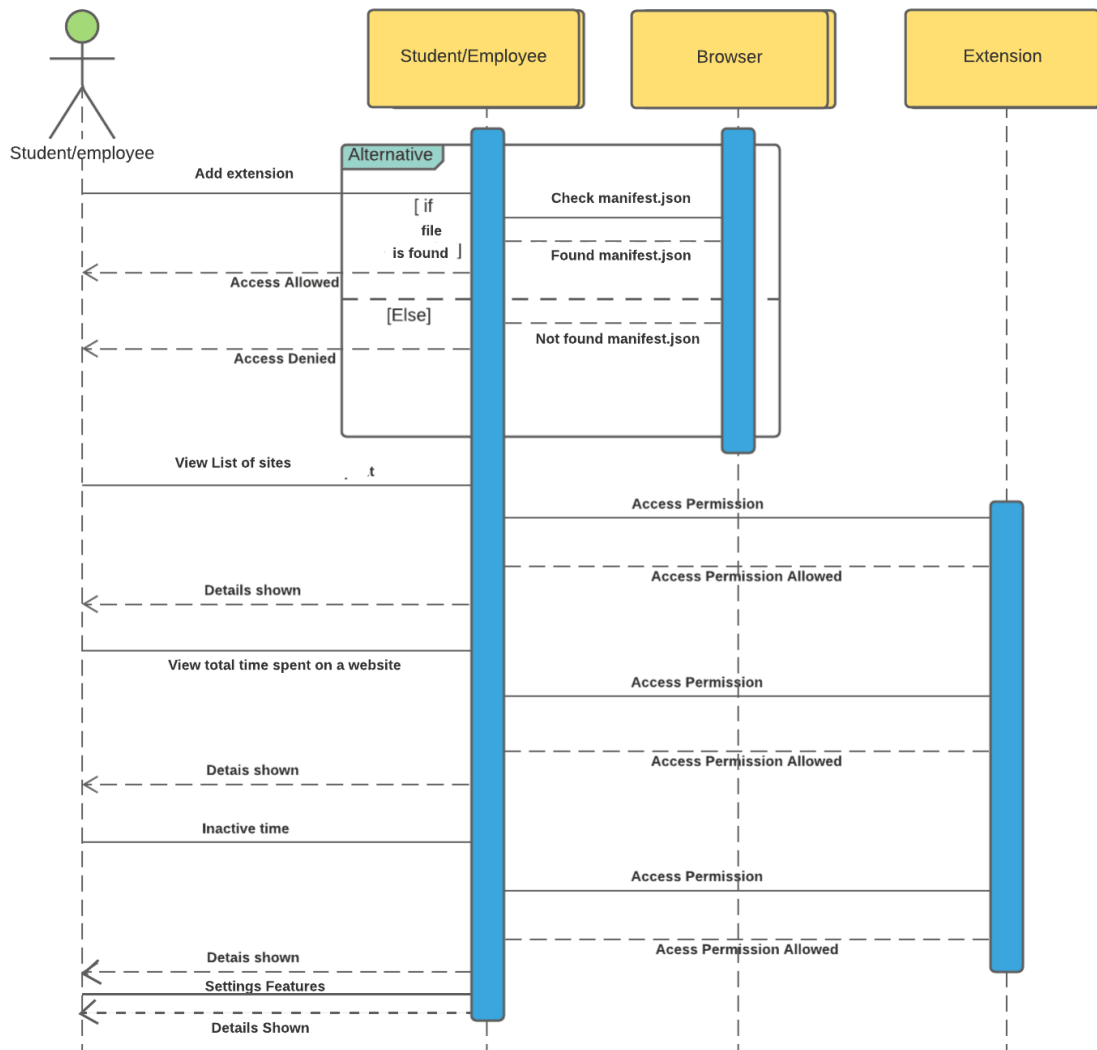
..... continued

2. Logical Architecture (Class Diagram, Sequence Diagram, State Diagram)

Class Diagram:



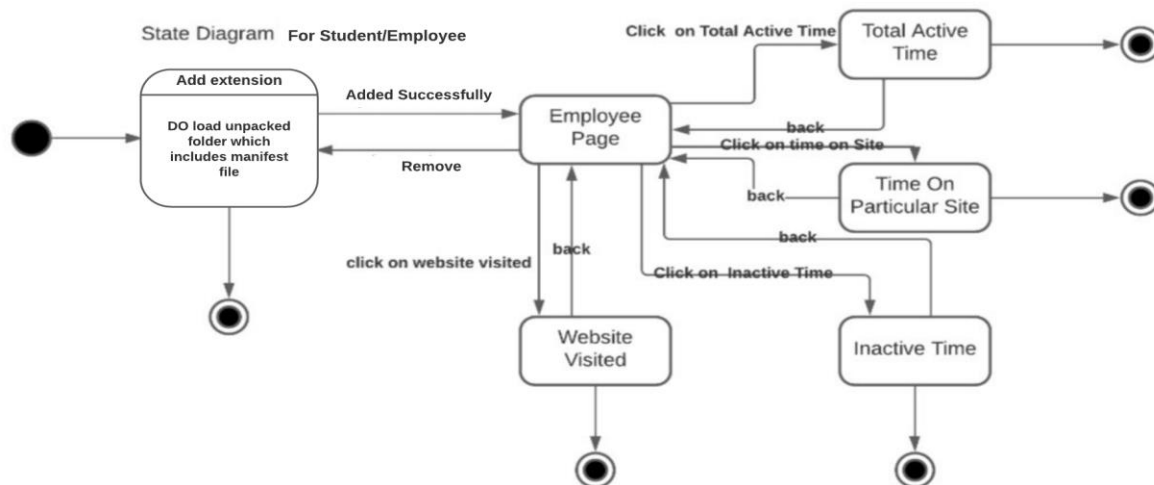
Sequence Diagram:



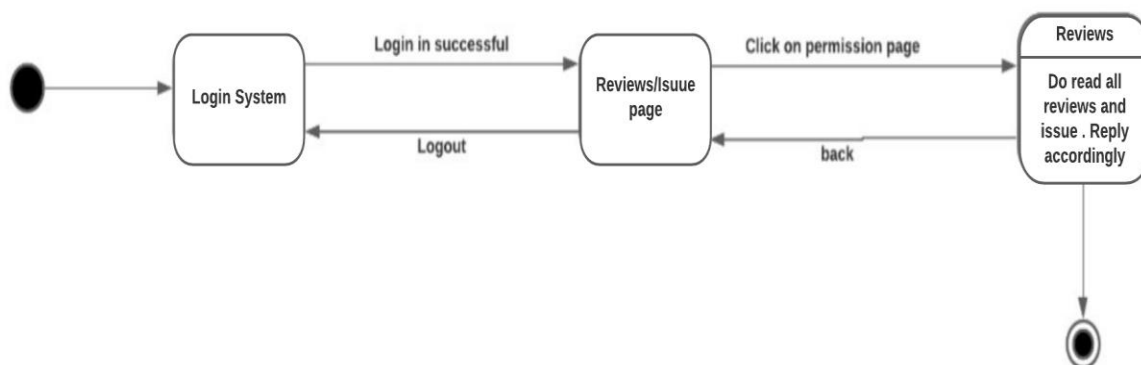
..... continued

State Diagram :

STATE DIAGRAM Of Student/Employee



STATE DIAGRAM OF Developer



2.1 Logical Architecture Description

2.1.1 Class Diagram explanation :

- 1) Association Relationship** - An association indicates that objects of one class have a relationship with objects of another class, in which this connection has a specifically defined meaning. These are: “UI” and “Activity”, and “UI and Local Storage”.
- 2) Aggregation Relationship** - Aggregation refers to the formation of a particular class as a result of one class being aggregated or built as a collection. In aggregation, the contained classes are not strongly dependent on the lifecycle of the container. In the same example, books will remain so even when the library is dissolved. To show aggregation in a diagram, draw a line from the parent class to the child class with a diamond shape near the parent class.
The class “Activity” is made up of one or more “Tab”, “Tab” and “Time Interval”
- 3) Composition Relationship** - The composition relationship is very similar to the aggregation relationship. with the only difference being its key purpose of emphasizing the dependence of the contained class to the life cycle of the container class. That is, the contained class will be obliterated when the container class is destroyed.
These are: “Block” depends on “Local Storage”, and “Block depends on Restriction”

2.1.2 Sequence Diagram :

Arrow line signifies there is a send message taken place. Response is being shown by dotted arrows.

Login : Faculty or employer is able to login into the database server through this page if valid credentials are given. If invalid credentials are given ,then the system asks for entering the credentials again.

Total Active Time : Faculty or employer is able to view the total active hours of the student/employee during working hours on clicking this button.

AFK Time : Faculty or employer is able to view the total inactive hours of the student/employee during working hours on clicking this button.

Sites Visited : Faculty or employer is able to view the sites being visited in the total active hours of the student/employee during working hours on clicking this button.

Pictorial View : Faculty or employer is able to view the total active hours of the student/employee during working hours in chart view on clicking this button. Each section will be of different websites and time spent on them.

2.1.3 State Diagram:

Initial state is being shown by starting with a black dot. Final State is being shown by the black dot surrounded by an empty circle.

1) Faculty/Employer: On clicking the Login button on the Login page, it lands on the Employee page. The user can now click on Total Active Time / Time on Particular Website/ Websites Visited/ Inactive Time, or Logout button, which lands up the user to the Login page again.

2) Student/Employee: On clicking the Login button on the Login page, it lands up in the Employee page. On clicking the Permission button, it would again land up Student/Employee on the Permission Page. On clicking the Logout button, Student/Employee lands up to the login page again.

2.2 Class Name : Notification

Description: This class gives the user to enter the system by authenticating the entered credentials.

Attribute 1: domain

Attribute 2: time

2.3 Class Name : Activity

Description: This class gives the user to enter the system by authenticating the entered credentials.

Method 1: addTab(tab)

Input: A tab

Output:

Method Description: This method adds a new tab in the employer section.

Method 2: isValidPage(tab)

Input: A tab

Output:

Method Description:

This method checks whether it is a valid page or not.

Method 3: isInBlackList(domain)

Input: domain

Output:

Method Description:

This method takes input as email and password from the user and checks whether it is added in the blacklist and if yes, then it will show that it is blacklisted.

Method 4: isLimitExceeded(domain,tab)

Input: domain, tab

Output:

Method Description:

This method takes input as the domain,tab from the current activity list and checks whether it has exceeded the limit and if yes,then it will show the limit exceeded.

Method 5: isNewUrl(domain)

Input: domain

Output: Admin landing page of login successful

Method Description:

This method checks whether it is newurl or not , returning the message as the condition falls.

Method 6: getTab(domain)

Input: domain

Output: Admin landing page of login successful

Method Description:

This method takes input as email and password from the user and checks whether it is authorized login or not and if the result is successful it leads to another activity page.

Method 7: extractHostname(url)

Input: url

Output: Admin landing page of login successful

Method Description:

This method takes input as an url and extracts hostname.

Method 8: updateFavicon(tab)

Input: tab

Output: Admin landing page of login successful

Method Description:

This method takes input as a tab and updates the favicon.

Method 9: setCurrentActiveTab(domain)

Input: domain

Output: Admin landing page of login successful

Method Description:

This method takes input as a domain and sets a current active tab.

Method 10: clearCurrentActiveTab()

Input: empty

Output: Admin landing page of login successful

Method Description:

This method clears the current active tab.

Method 11: addTimeInterval(domain)

Input: domain

Output: Admin landing page of login successful

Method Description:

This method takes input as domain and adds the time interval .

Method 12: closeIntervalForCurrentTab(domain)

Input: domain

Output: Admin landing page of login successful

Method Description:

This method takes input as a domain and closes the interval for the current tab.

Method 13: isNeedNotifyView(domain,tab)

Input : domain,tab

Output: Admin landing page of login successful

Method Description:

This method takes input as domain,tab and it notify when needed.

2.4 Class Name : UI

Method 1: getTableOfSite()

Input : void

Output: Table

Method Description:

This method provides the table which contains the website visited.

Method 2: setUIForToday()

Input : void

Output:

Method Description:

This method sets the UI for the new day.

Method 3: setUIForAll()

Input : void

Output:

Method Description:

This method sets the UI for all required .

Method 4: setUIForByDays()

Input :

Output:

Method Description:

This method sets the UI for days.

Method 5: addHRAfterTableOfSite(domain,tab)

Input : domain,tab

Output:

Method Description:

This method takes input as a domain,tab and adds HR after the table of site.

Method 6: setActiveTooltip(currentTab)

Input : current tab

Output:

Method Description:

This method set the active tooltip for the current tab.

Method 7: isNeedNotifyView(domain,tab)

Input : domain,tab

Output:

Method Description:

This method takes input as domain,tab and notify view if needed.

Method 8: addTableHeader()

Input : void

Output:

Method Description:

This method adds a table header.

Method 9: addLineToTableOfSite(domain,tab)

Input : domain,tab

Output:

Method Description:

This method adds a line to the table of site.

Method 10: createElementsForTotalTime()

Input : void

Output:

Method Description:

This method creates elements for total time.

2.5 Class Name : Block

Description: This class blocks the unauthorized user and keeps a record of it.

Attribute 1: storage

Attribute 2: restriction list

2.6 Class Name : Restriction

Description: This class restricts the unauthorized user.

Attribute 1: domain

Attribute 2: time

2.7 Class Name : Time interval

Attribute 1: day

Attribute 2: domain

Attribute 3: intervals

Method 1: addInterval()

Input : void

Output:

Method Description:

This method adds the time interval .

Method 2: closeInterval()

Input : void

Output:

Method Description:

This method closes the time interval.

2.8 Class Name : Tab

Method 1: addNewDay(string)

Input : string

Output:

Method Description:

This method takes input as a string and adds the provided string as the new day.

Method 2: incSummaryTime()

Input : void

Output:

Method Description:

This method gets the incSummaryTime.

Method 3: getTodayTime()

Input : void

Output: time

Method Description:

This method returns today's time as output.

Method 4: incCounter()

Input : void

Output:

Method Description:

This method gets the incCounter.

2.9 Class Name : LocalStorage

Method 1: saveTabs()

Input : void

Output:

Method Description:

This method saves all the current open tabs.

Method 2: saveValue()

Input : void

Output:

Method Description:

This method saves all the values time interval,website visit data .

Method 3: getValue()

Input : void

Output:

Method Description:

This method gets the values which have been saved in local storage.

Method 4: getMemoryUse()

Input : void

Output:

Method Description:

This method gets memory used to find the values which have been saved in local storage.

3. Execution Architecture

3.1 Execution Architecture

Runtime environment required is any system compatible with browser requirements with the minimum 4gb RAM and processor i3 or above.

3.2 Reuse and relationships to other products

NIL

4. Database Schema

4.1 Tables, Fields and Relationships

table to store data about websites will be created having fields
website_name, opening_time, closing_time

4.1.1 Databases

Data will be taken from chrome browser.

4.1.2 New Tables

Table name : website

Fields : website_url, opening_time, closing_time

4.1.3 New Fields(s)

Table Name	Field Name	Data Type	Allow Nulls	Field Description
website	website_url	varchar(1000)		Get this field from browser history
website	opening_time	varchar(100)		Get this field from browser history
website	opening_time	varchar(100)		Get this field from browser history