Department course scheduling assistant

Software Requirements Specification

Version 3.0

November 15 2016

Team-D Sigma

**Nitheesha Kotagiri**

**Harika Malempati**

**Nikhil Kumar Vemula**

**Sai Kiran Gandham**

**Sravya Kandepu**

**Sanket Devrao Selokar**

**Prepared for**

**Graduated Directed Project - I**

**Instructor: Dr. Michael Oudshoorn**

**Fall 2016**

**Revision history:**

|  |  |  |  |
| --- | --- | --- | --- |
| **Date** | **Version** | **Description** | **People** |
| October 07, 2016 | 1.0 | First draft | **Project Owner and Client:**  Dr. Michael Oudshoorn  **Faculty Advisor:**  Dr. Michael Oudshoorn  **Project Group:**  Nitheesha Kotagiri  Harika Malempati  Nikhil Kumar Vemula  Sai Kiran Gandham  Sravya Kandepu  Sanket Devrao Selokar |
| November 15, 2016 | 2.0 | Second draft  Updated prototypes and usecases. | **Project Owner and Client:**  Dr. Michael Oudshoorn  **Faculty Advisor:**  Dr. Michael Oudshoorn  **Project Group:**  Nitheesha Kotagiri  Harika Malempati  Nikhil Kumar Vemula  Sai Kiran Gandham  Sravya Kandepu  Sanket Devrao Selokar |
| November 15, 2016 | 3.0 | Third draft  Updated use case and porotypes. Added new functionality like adding and deleting the faculty. | **Project Owner and Client:**  Dr. Michael Oudshoorn  **Faculty Advisor:**  Dr. Michael Oudshoorn  **Project Group:**  Nitheesha Kotagiri  Harika Malempati  Nikhil Kumar Vemula  Sai Kiran Gandham  Sravya Kandepu  Sanket Devrao Selokar |

**Document approval:**

The following Software Requirements Specification has been accepted and approved by the following:

|  |  |  |  |
| --- | --- | --- | --- |
| **Signature** | **Printed Name** | **Title** | **Date** |
| Saikiran Gandham | Saikiran Gandham | Client Manager | 11/15/2016 |
|  | Dr. Michael Oudshoorn | Faculty Advisor |  |
|  | Dr. Michael Oudshoorn | Project Sponsor |  |

#### **Table of Contents**

[**1. Introduction** 1](#_Toc467009902)

[**1.1 Purpose** 1](#_Toc467009903)

[**1.2 Scope** 1](#_Toc467009904)

[**1.3 Definitions** 1](#_Toc467009905)

[**1.4 Overview** 2](#_Toc467009906)

[**2. General Description** 3](#_Toc467009907)

[**2.1 Product Perspective** 3](#_Toc467009908)

[**2.2 Product Functions** 3](#_Toc467009909)

[**2.3 User Characteristics** 3](#_Toc467009910)

[**2.4 General Constraints** 3](#_Toc467009911)

[**2.5 Assumptions and Dependencies** 4](#_Toc467009912)

[**3. Specific Requirements** 5](#_Toc467009913)

[**3.1 External Interface Requirements** 5](#_Toc467009914)

[3.1.1 User Interfaces 5](#_Toc467009915)

[3.1.2 Hardware Interfaces 10](#_Toc467009916)

[3.1.3 Software Interfaces 10](#_Toc467009917)

[3.1.4 Communications Interfaces 10](#_Toc467009918)

[**3.2 Functional Requirements** 10](#_Toc467009919)

[3.2.1 Feature #1: User login 10](#_Toc467009920)

[3.2.2 Feature #2: Faculty Home Page 11](#_Toc467009921)

[3.2.3 Feature #3: Faculty Course Page 11](#_Toc467009922)

[3.2.4 Feature #4: Administrator Home Page 12](#_Toc467009923)

[3.2.5 Feature #5: Administrator scheduler screen 12](#_Toc467009924)

[3.2.6 Feature #6: Scheduler Conformation page 13](#_Toc467009925)

[3.2.7 Feature #7: Add professor to professor list 13](#_Toc467009926)

[3.2.8 Feature #8: Delete professor from professor’s list 14](#_Toc467009927)

[**3.3 Use cases** 15](#_Toc467009928)

[3.3.1 Common use cases for each user 15](#_Toc467009929)

[**** 15](#_Toc467009930)

[**3.4 Non Functional Requirements** 16](#_Toc467009931)

[3.4.1 Performance 16](#_Toc467009932)

[3.4.3 Availability 16](#_Toc467009933)

[3.4.4 Security 16](#_Toc467009934)

[3.4.5 Maintainability 16](#_Toc467009935)

[3.4.6 Portability 16](#_Toc467009936)

[**4. Analysis Models** 17](#_Toc467009937)

[**4.1 Data Flow Diagram (DFD)** 17](#_Toc467009938)

[**5. Change Management Process** 18](#_Toc467009939)

[**6. Client Acceptance** 19](#_Toc467009940)

# **1. Introduction**

This document is to present the purpose, scope and functional and non-functional requirements which we are going to be implemented during the development of the project. This document will give an overview of the application.

## **1.1 Purpose**

The purpose of this document is to give an outline of the application and it’s working to all the team members and the client.

## **1.2 Scope**

The project is about developing a web based application titled “Department course scheduling assistant”. Our application helps University administrator to schedule and view classes in two ways like course wise scheduling and faculty wise scheduling. Faculty can view his/her class schedules and represents in graphical user interface and also the schedules of other faculty who are teaching the university. The idea of the application evolved mainly to easy the process of scheduling classes for thousands of students by considering all the limitations and priorities of all the faculty members.

## **1.3 Definitions**

|  |  |
| --- | --- |
| **Term** | **Definition** |
| System | It is web based application. |
| Database | A collection of all the information monitored by this system. |
| Software Requirements Specification | A document that completely describes all of the functions of a proposed system and the constraints under which it must operate. For example, this document. |
| Stakeholder | Any person with an interest in the project. |
| User | The person who uses the application seeking help. |
| Scheduler | A Person who schedule classes for faculty. |
| Use cases | Graphical representation of flow in project. |
| Mockups | Sample User Interfaces of the project. |

## **1.4 Overview**

The SRS document provides an outline of the system functionality and system interfaces and interaction. Further, non-functional requirements, use cases, software and hardware interfaces and other requirements of the project.

# **2. General Description**

This section will give an overview of the whole application. This section describes the perspective of the product, the basic functions, and the types of users that will be using the application, the constraints and the assumptions.

## **2.1 Product Perspective**

The main goal of this web project is to provide an effective user interface to schedule the courses in Northwest Missouri State University.

The “Department course scheduling assistant” application is a web project. This application is mainly useful for university authorities to schedule the classes easily and faculty can view their schedule in an interactive way. University Administrator and Faculty can access this application.

## **2.2 Product Functions**

With the web based application administrator can login with provided credentials. Once the user logs in, user will be navigated to home page. User can access multiple tabs in this page like scheduler, calendar view. Administrator can schedule classes by using scheduler tab by clicking on it and can view classes details using calendar view based on our priority like course wise view or faculty wise view. Faculty can also login to this web based application using his provided credentials. Faculty will be navigated to home page, where he can view calendar based on his priority like course wise view or faculty wise view. Administrator can have multiple selection criteria while scheduling classes like Class room selection, faculty selection and timing etc. And at any point of time the user can logout by clicking on the logout button.

## **2.3 User Characteristics**

There are two types of users that interact with this application – administrator user and faculty user.

The Administrator user is the one who’s having permissions to schedule the classes and he acts like responsible person for all scheduling tasks. The Administrator user assign faculty to different courses based on their requirement criteria and limitations of that faculty like subjects he/she teach , hours he/she works.

The Faculty are the one who can view his classed schedule like class’s timings, lab session timings. Faculty can also view other faculty schedules.

## **2.4 General Constraints**

This “Department course scheduling assistant” application is specifically for desktops. It is a web project.

The internet connection is must for this application. To login to this application and scheduling class’s internet connection we need.

## **2.5 Assumptions and Dependencies**

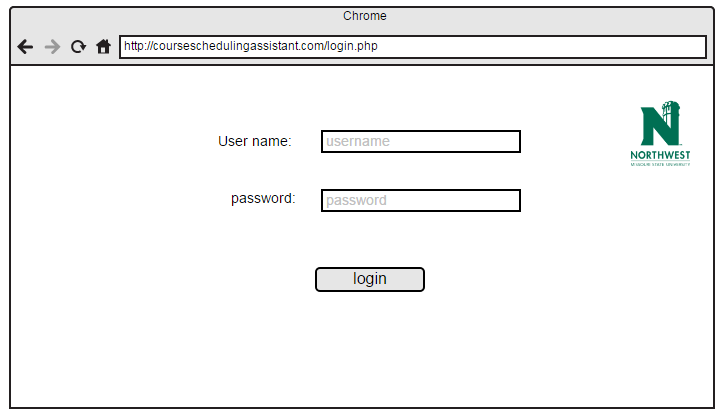
The assumptions about the product is that it can also be used from mobile phones with high speed. But it is difficult schedule class from mobile interface, so it is specially designed for desktops. If server gets down or in poor internet connection situation, it is not possible to use this application. Only authorized persons can use this application. Normal user can’t use this application by creating an account.

# **3. Specific Requirements**

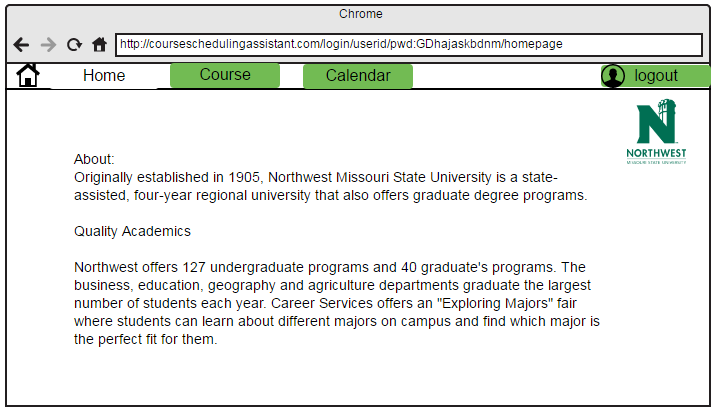
## **3.1 External Interface Requirements**

### 3.1.1 User Interfaces

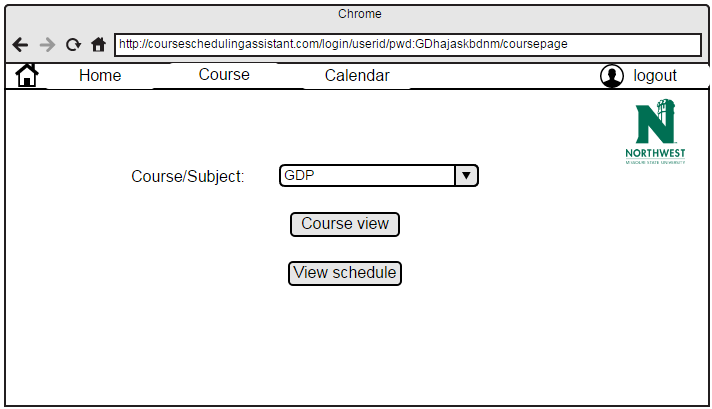
1. Login screen: For both user/ admin.



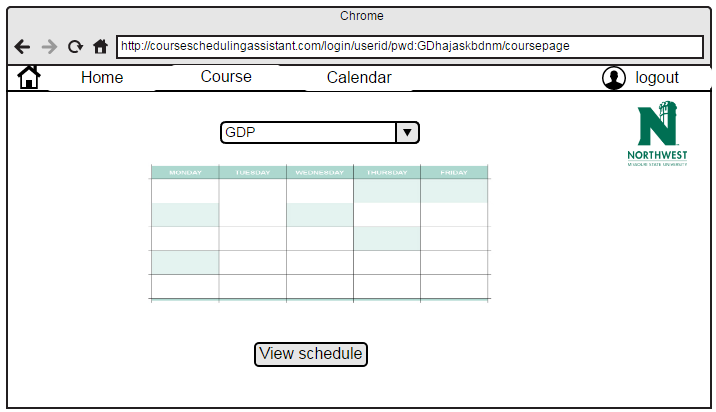
2. Home screen is displayed after the login.



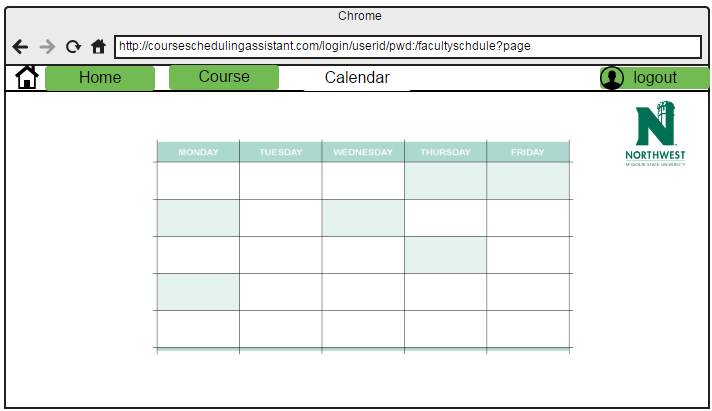
3. Course view page is displayed after navigating from home page by clicking on course tab.

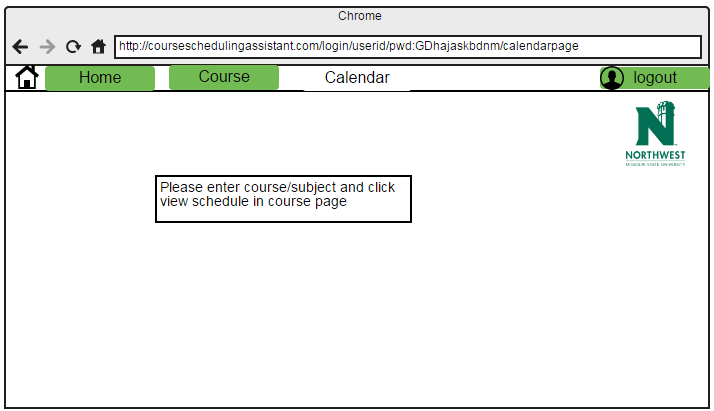


4. Course view page is displayed after clicking on course view tab in course page.

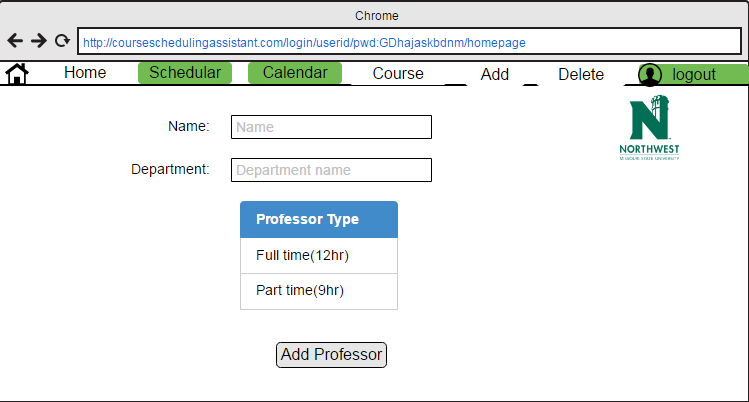


5. Calendar view is displayed after clicking on Calendar tab in course page.

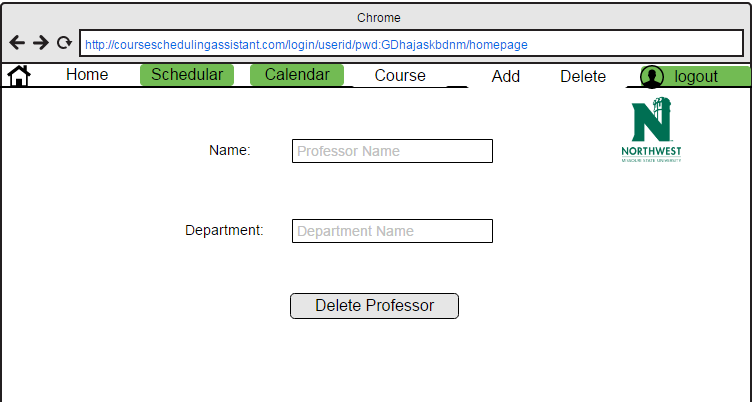


6. Error message is displayed when user try to navigate to calendar page and he/she is not scheduled with classes.

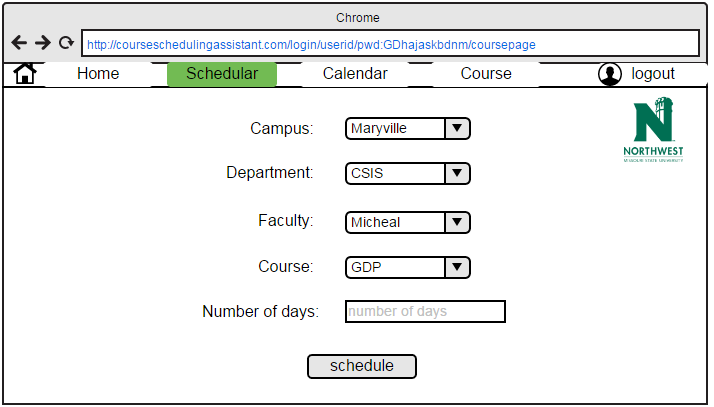
7. If user is administrator and administrator want to add professor to list.



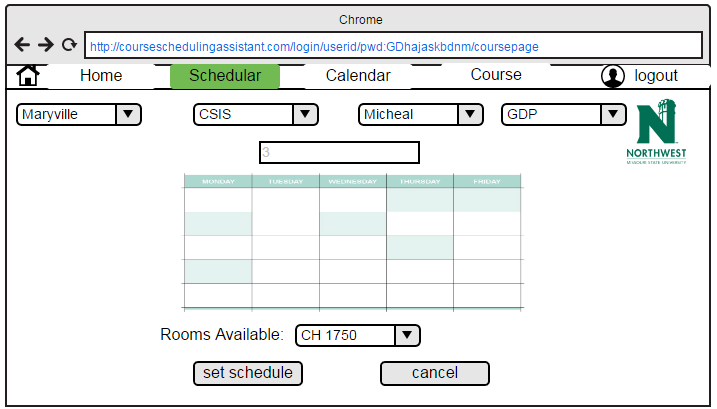
8. If administrator want to delete professor from list.



9. If administrator want schedule class for professor.



10. Faculty schedule page with calendar which displays class room and class timings.



### 3.1.2 Hardware Interfaces

The hardware interfaces required for this application are – Desktop or laptop with windows Operating system, 2 GB RAM and 250 GB memory. The device should also have internet.

### 3.1.3 Software Interfaces

Before we start developing a web based application in, we need to setup software development environment for that we need to have following tools.

* NetBeans IDE
* Windows system
* Advanced java script
* Enrollment into Java developer program
* Testing tools

### 3.1.4 Communications Interfaces

The Department course scheduling assistant application shall use the HTTP protocol for communication over the internet and for the intranet communication will be through TCP/IP protocol suite.

## **3.2 Functional Requirements**

### 3.2.1 Feature #1: User login

**Requirements:**

Once the webpage is launched the first display in the screen faculty will be directed to login page. In user login page user need to enter his/her credentials and these credentials will be validated in the database and the authentication for login will be provided.

**Input:**

User need to enter his/her credentials i.e. Username and password and need to click/select login button to login.

**Description:**

1. User need to provide username and password to login.

**Outputs:**

If user is faculty he/she will be redirected to faculty home page. If user is administrator he/she will be redirected to administrator home page. If the administrator is a faculty then he is provided with the administrator rights.

**Error Handling:**

If the provided login credentials are wrong or forget to enter credentials and click login an error is thrown.

### 3.2.2 Feature #2: Faculty Home Page

**Introduction:**

In faculty home page there will be calendar and course buttons. Those 2 buttons will be redirected to faculty course page and faculty calendar page.

**Inputs:**

1. In this page faculty/user will not enter any data. He/she need to click button depending on his requirement.
2. Faculty has to click on course button or calendar button to view his schedule.

**Description:**

If user is faculty he/she will be redirected to faculty home page. If user is administrator he/she will be redirected to administrator home page.

**Outputs:**

If faculty clicks on the course button he/she will be redirected to course page. If he/she clicks on calendar it will be redirected to calendar page.

### 3.2.3 Feature #3: Faculty Course Page

**Introduction:**

In faculty course page, faculty will select subject for which he/she going to see the schedule.

**Inputs:**

1. User/faculty has to select a subject from dropdown.
2. In this page user/faculty has to click course view or view schedule after selecting subject.

**Description:**

User need to select subject from dropdown menu and he/she has to click on the course view or on the view schedule.

**Outputs:**

1. If user clicks on course view it displays information about particular course in calendar.
2. If user clicks on the view schedule it displays timetable of the professor.
3. The faculty can view all the scheduled class of the department.

**Error Handling:**

If user did not select subject from dropdown menu, it displays alert message.

### 3.2.4 Feature #4: Administrator Home Page

**Introduction:**

If user is administrator who has the responsibility to schedule classes for professors.

**Inputs:**

In this page administrator will be seeing description about university and he/she need to select scheduler button to schedule a class or he/she can select course button to see the information about the course.

**Description:**

This page is the administrator home page.

**Outputs:**

User can view description and images about the university.

### 3.2.5 Feature #5: Administrator scheduler screen

**Introduction:**

When user click on the scheduler button this page will be displayed.

**Inputs:**

In this page administrator has to select by clicking the items such as campus location, department, faculty name, course and number of days (professor is going to teach in week).

**Description:**

In this administrator is going to schedule class for professor.

**Outputs:**

After giving inputs in the boxes and clicks on the set schedule button, then this page will redirected to another conformation page.

**Error Handling:**

If administrator did not enter any value, it displays alert message to enter mandatory fields.

### 3.2.6 Feature #6: Scheduler Conformation page

**Introduction:**

If administrator clicks on the set schedule button on the scheduler page this page will be displayed.

**Inputs:**

If administrator can select set schedule button or he/she can cancel the process.

**Description:**

This page is to conform about the details given in previous page and to select class room.

**Outputs:**

After checking all fields and clicking on the set schedule class will be allotted to professor and displays calendar view of particular course.

**Error Handling:**

If administrator did not enter the room number, an alert message is displayed.

### 3.2.7 Feature #7: Add professor to professor list

**Introduction:**

If a new professor came to college, administrator have to add professor to list of professors in university.

**Inputs:**

Administrator have to enter faculty name and Department name and have to click on Add professor button.

**Description:**

This page is to add professor to professor’s list.

**Outputs:**

After entering data, new professor will be added to list.

**Error Handling:**

If administrator did not entered the data in given fields, pop up message will be displayed.

### 3.2.8 Feature #8: Delete professor from professor’s list

**Introduction:**

If a professor is resigned his/her job, administrator have to delete professor from list.

**Inputs:**

Administrator will enter name and department and click on Delete professor button.

**Description:**

After administrator entering data system searches for professor in department entered by administrator.

**Outputs:**

After giving data and clicking Delete professor button, professor will be deleted from list.

**Error Handling:**

If administrator did not entered the data in given fields, pop up message will be displayed.

If given data is incorrect (i.e., given name does not matches from given list) then pop up message will be displayed.

## **3.3 Use cases**

This section describes the use cases of different users in this project. Different users of this application are university administrator and faculty.

### 3.3.1 Common use cases for each user

## 

## **3.4 Non Functional Requirements**

### 3.4.1 Performance

The system should be able to show the classes and its timings for faculty in course wise view and in faculty wise view. The system should show the scheduled classes to Admin in two ways, in addition to that Admin can able to schedule class’s course wise and faculty wise based on the requirements and limitations.

**3.4.2 Reliability**

System is reliable in terms of scheduling classes by admin, viewing the classes by both admin and faculty.

### 3.4.3 Availability

The app is available to both the users.

### 3.4.4 Security

All the user information data can able to fetch from database so there is no chance for fault misrepresentation, every user is provided a unique credentials, so that only authorized persons can access this web application.

### 3.4.5 Maintainability

The system can be flexible to add more functionality in future. System can be tested

### 3.4.6 Portability

This app is portable for all the Windows devices from version windows XP and above.

# **4. Analysis Models**

## **4.1 Data Flow Diagram (DFD)**



# **5. Change Management Process**

1. All the changes in the requirements will be initially discussed with our instructor and will be documented as a minor version.
2. The modified document will be sent to the client for approval.
3. This document will be published as a major version only after the approval from the client.

# **6. Client Acceptance**



Date: November 15, 2016