Student Faculty Synchronized Calendar

Introduction

Student Faculty Synchronized Calendar is a project that emphasizes on ensuring efficient communication between faculty and students. The lack of communication between them often confuses students about their schedules and important dates that cause them to miss important deadlines. Whereas the faculties face problems while assigning exam dates that may clash with student's schedules. To solve these problems is the main goal of our project along with many other similar problems. The calendar will be a synchronized calendar that will get updated in real time whenever a faculty make changes and students will be notified immediately. Along with showing class timings, the calendar will display quiz dates, assignment deadlines and exam dates to students. Therefore, all necessary details being in one place, students can easily plan for their studies.

Motivation

This project's main purpose is to ensure stress free and maximum productive communication among students and faculties of BRAC University. The motivation has come observing students' daily schedules and challenges in BRAC University. As BRAC University is an open credit university, students attending a class have different class routine and examination schedules. It is a challenging issue for not only faculties but also students to make a proper schedule considering each and every students situation. To overcome this problem, project "Student Faculty Synchronized Calendar" has been initiated.

Methodology

To implement our project, we have followed Waterfall development method. The reason behind this is that, we have had clear specification for our project and we have worked with familiar technology. Waterfall model is appropriate in these kind of scenario [1]. As all of us are students of BRAC University and we consulted with our faculties regularly, we figured out the specific requirements and also we worked with php, mysql, html and css which are familiar to us.

System Description

Our system is mainly a website which will require login with proper credentials. Login credentials will be given by university authority. Our System Request will emphasize business need,

business requirements, business value and special issues of our system. Our system request is shown in figure 1.

System Request	
Project Sponsor	BRAC University
Project name	Student Faculty Synchronized Calendar
Business Need	This project has been initiated to ensure efficient communication among students and faculties and to keep faculties and students updated regularly.
Business requirements	2 types of users. 1. Faculty • Assigning quiz dates, assignment deadlines, mid and final dates. • Events / Competition information • Cancelling classes and uploading make up classes schedule • Consultation hour 2. Students • Can switch between weekly and monthly calendar • All necessary details in the calendar • Only view only access
Business value	Selling value: $100

Special Issue	1. Improve students' experience and mental health. 2. We will launch it before spring 2019 semester

Figure 1: System Request

We have done proper Feasibility Analyses of our system to calculate cash flow, Return On Investment and Break Even Point. Our overall feasibility analysis shows that our system's break-even point is 2.8 years and our return on investment is 5.6%. In addition, we have done a specific requirement analysis in which we have distinguish functional and non-functional requirements. Functional requirements are divided based on 3 main categories and non-functional requirements are divided into 4 main categories.

Feasibility Analysis

	Year 0	Year 1	Year 2	Year 3	Total
Total Benefits		200	250	300	750
Total Costs	400	100	110	100	710
Net Benefits	-400	100	140	200	40
Cumulative Net Cash Flow	-400	-300	-160	40	

ROI = (Total Benefits – Total Costs) / Total Costs
= (750 – 710) / 710
= 5.6%

BEP = Numbers of years of negative cash flow + (That year's Net Cash Flow – That year's Cumulative Cash Flow) / That year's Net Cash Flow
= 2 + (200 – 40) / 200
= 2.8 years.

Requirement Analysis

Functional requirements:

1. Calendar
   1.1 Two types of user will be able to use their own synchronized calendar.

1.2 All calendars will have two views: Weekly and monthly.
1.3 All the changes made by faculties will be synchronized with the student's calendar in real time as well as notify the students.

2. Editing and updating schedule (Faculty only)
   2.1 Faculties will be able to edit or update class schedules.
   2.2 They will be able to assign quiz and assignment deadlines.
   2.3 They will be able to change mid or final exam schedules without creating a clash with other timings.
   2.4 They will be able to display their consultation hours.

3. Viewing the calendar(For Students)
   3.1 Students will be able to see upcoming events and deadlines below their calendar.
   3.2 Students will not have access to change anything in the calendar.
   3.3 Clashes with their schedules will be automatically notifies to faculties.

Nonfunctional requirements:

1. Operational requirements
   1.1 The system will operate in Windows, Macintosh, Linux environments.
   1.2 The system will be able to read HTML, CSS, PHP file. It will use MySQL for database.
   1.3 The system will be able to import jpg file.

2. Performance requirements
   2.1 The system must be up to date after any information is uploaded
   2.2 The system's server must be functioning 24/7.
   2.3 There will be 2 type users and must be 2 different user interfaces where faculties will have the ability to alter information and students can only view the data.

3. Security requirements
   3.1 Each and every individual user must have distinct login credentials

4. Cultural and political requirements
   4.1 No specific cultural and political requirements.

Our system's user is generalized into 2 categories. Between these types of users, one is student type and another is faculty type. Here only users who are categorized as faculty can only update information. USE CASE diagram of our system is shown in figure 2.

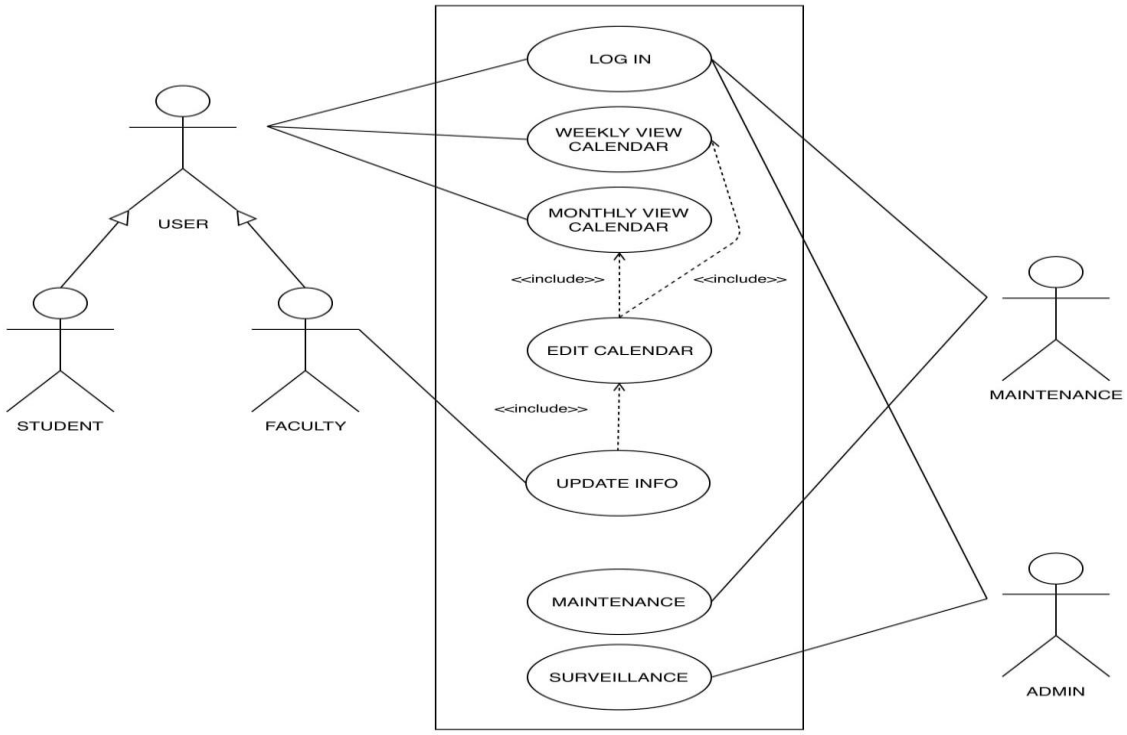

Figure 2: Use Case Diagram

In our Use Case diagram there are 4 actors; Student, Faculty, Maintenance and Admin. Student and Faculty can be generalized into the actor User. The User, Maintenance and Admin can all login Log In but they will have different access levels. The User can view a weekly and monthly view of the Calendar. The Faculty can update the information on the calendar and any changes made can be seen by the User. Maintenance can keep a record of their assets, schedule and track maintenance tasks to make sure the system is running smoothly. Admin will do the initial setup every semester by assigning routines to all the Students and Faculties. They will make sure the information is correct in the calendar.

Moreover, our system's main two users are student and faculty. Our sequence diagram in figure 3 and figure 4 will describe the users' system using sequence.

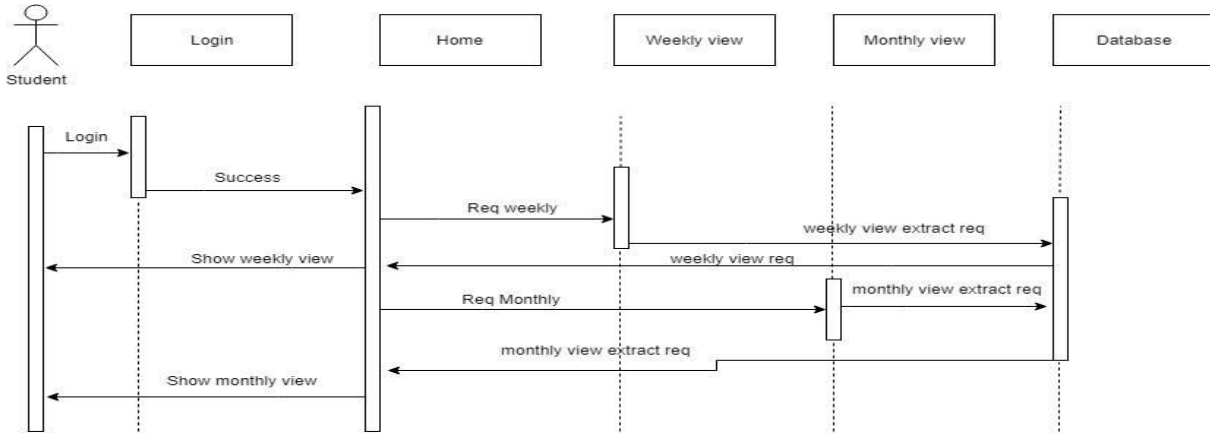

Figure 3: Sequence Diagram of Student

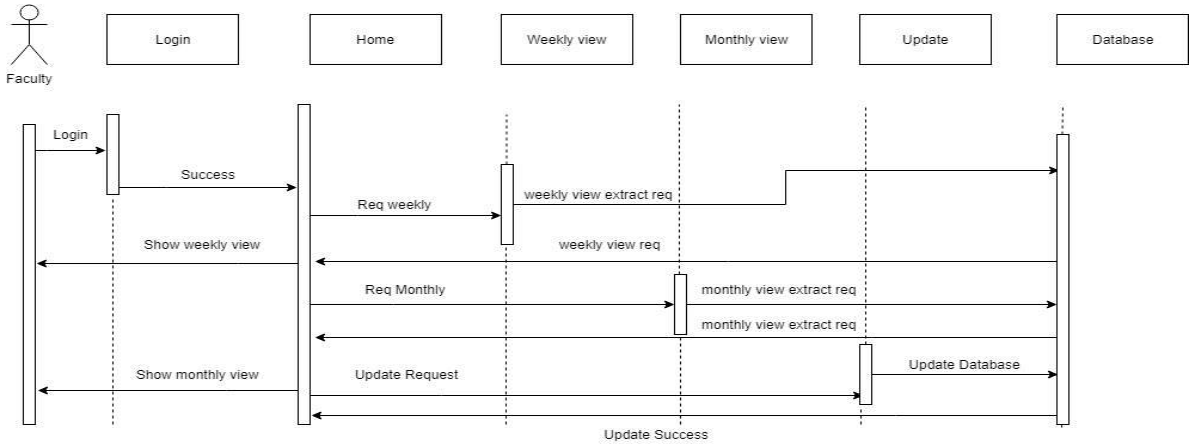

Figure 4: Sequence Diagram of Faculty

Apart from that, our activity diagram will show the overall activity process of our system. We have a dedicated database which is shown via ER diagram in figure 8 and Schema diagram in figure 6. Here, our diagrams illustrate our system from different aspects. For example, sequence diagram illustrates system use sequence, data flow diagram represents how data flows from one action to another action.

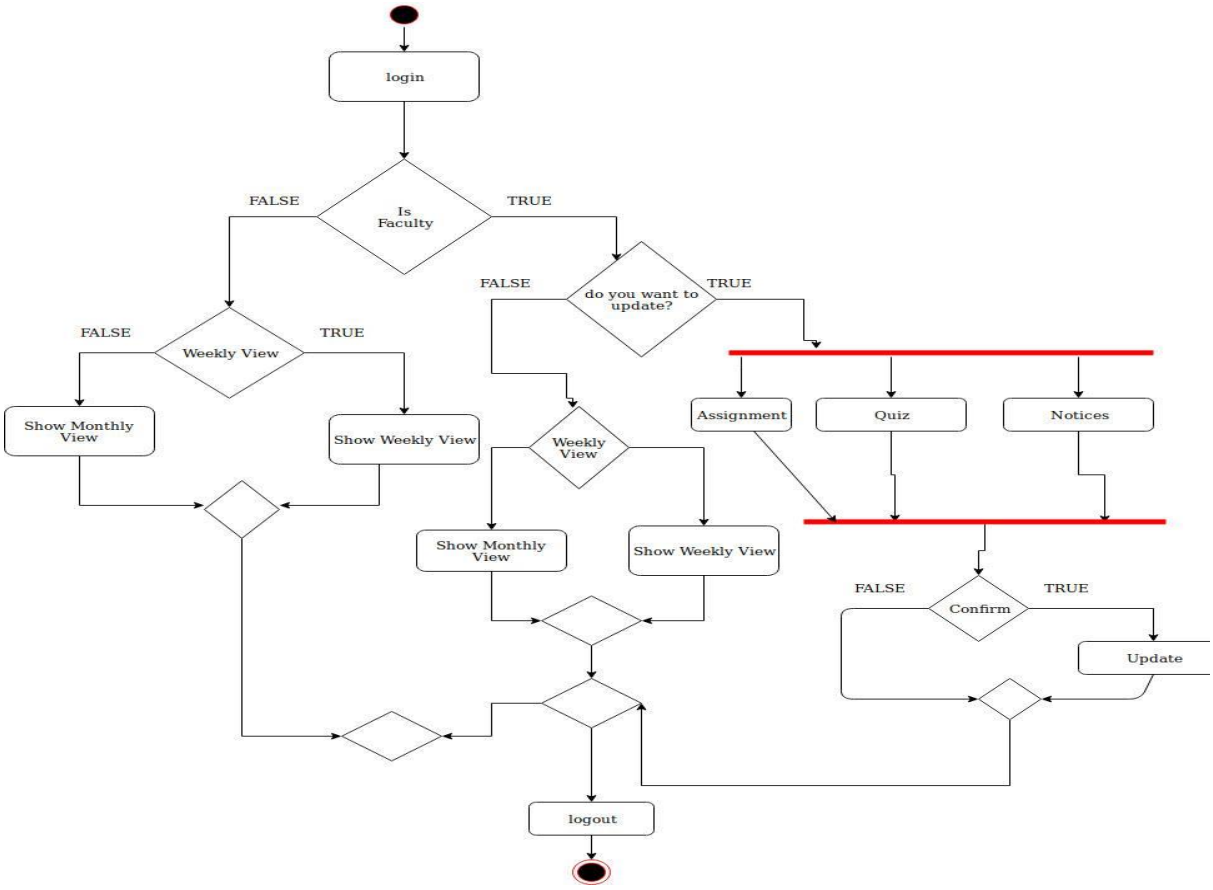

Figure 5: Activity Diagram

Our activity diagram in figure 5 shows both user categories activity from log in to log out. Besides, our system use private database. As a result, while executing any instruction, data will flow from one state to another. Our data flow diagram in figure 7 will represent flow of data for executing any instruction.

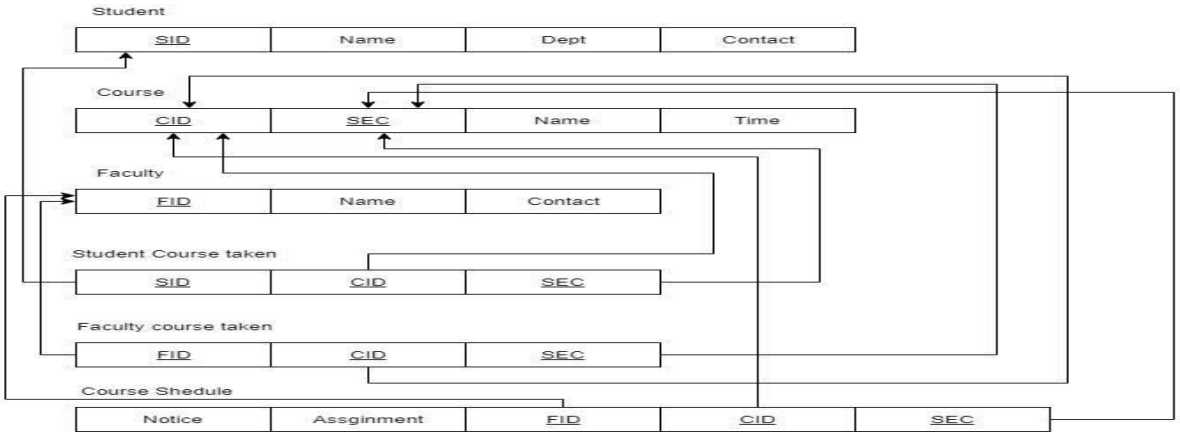

Figure 6: Schema

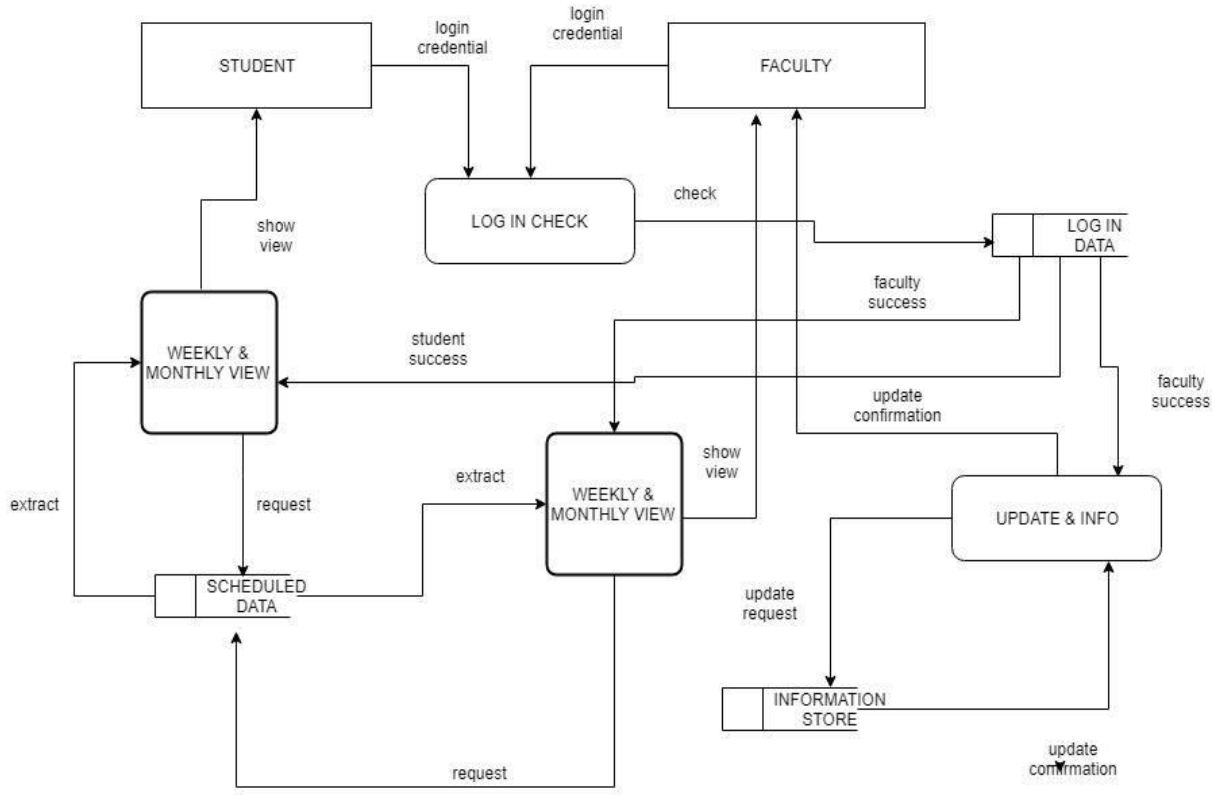

Figure 7: DFD Diagram

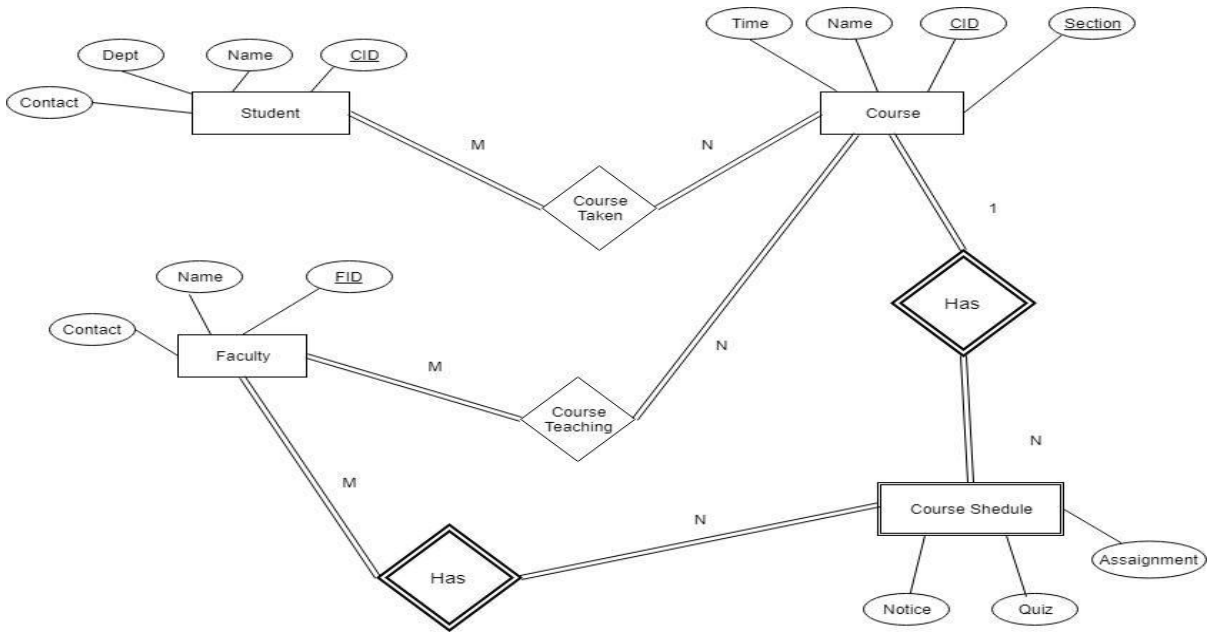

Figure 8: ER Diagram

Software Cost Estimation

For our software cost estimation we have used COCOMO model. Using this model we have calculated our functional points and based on functional points we have calculated total line required for our codes, person-month and estimated time of our project.

		Function Point (COCOMO Model)			
Description		Complexity			
	Total Number	Low	Medium	High	Total
Inputs	4	2 x 3	1 x 4	1 x 6	16
Outputs	7	2 x 4	4 x 5	1 x 7	35
Queries	9	4 x 3	2 x 4	3 x 6	38
Files	1	0 x 7	1 x 10	0 x 15	10
Program Interface	1	0 x 5	0 x 7	1 x 10	10

Total Unadjusted Function Points (TUFP): 109

Step 2:

AFP = UFP x 1 = 109 x 1 = 109

Step 3:

In PHP: 90 FP

Line required = 90 x 67 = 6030

In HTML: 19 FP

Line required = 19 x 15 = 285

Total Line required = 6030 + 285 = 6315

Step 4:

Effort = 1.4 * KLOC = 1.4 * (6.315) = 8.841 person-month

Step 5:

Schedule Time = 3.0 * person-month1/3

= 3.0 * 8.8411/3

= 6.20 months

Estimated time is little over 6 months.

It shows that to implement our system we need 8.841 person-month and we estimated that we need little over 6 months of time.

Timeline

To complete the system development within the timeline, we have created a Gantt Chart to divide our task in proper time line. Figure 9 shows the Gantt chart of our project to illustrate the timeline.

Conclusion

In short, our system named "Student Faculty Synchronized Calendar" will ensure best communication among students and faculty. It will update on real time which will ensure every students get their faculty's notice and update for quizzes and assignments. This system will make both students and faculties life in BRAC University easy and productive.

References

1. Tutorialspoint.com. (n.d.). SDLC Waterfall Model. Retrieved from https://www.tutorialspoint.com/sdlc/sdlc_waterfall_model.htm