



**TRIBHUVAN UNIVERSITY**  
**INSTITUTE OF ENGINEERING**  
**PULCHOWK CAMPUS**

**A PROJECT PROPOSAL**

**ON**

**UNIVERSITY CLASS PORTAL**

**IN PARTIAL FULFILLMENT OF THE REQUIREMENTS FOR THE COURSE  
OF SOFTWARE ENGINEERING**

**SUBMITTED BY**

AASHISH MANANDHAR	072/BCT/501
ASHISH PAUDEL	072/BCT/506
AVISHEKH SHRESTHA	072/BCT/507
NRIPENDRA TIMALSINA	072/BCT/523
RAMESH PATHAK	072/BCT/527
SARAMSHA DOTEL	072/BCT/534

**DEPARTMENT OF ELECTRONICS AND COMPUTER ENGINEERING**

**JANUARY, 2018**

## **ABSTRACT**

The ‘University Portal’ is an education management platform. It aims to fulfil the requirements of three kinds of users in an university: the administrator, the tutor and the student. The requirements of an administrator include things like admitting a new batch into the university, creating courses, assigning tutors to classes, etc. The teacher will have facilities to publish his/her weekly plan, make changes to his/her class schedules, fix assessment and assignment dates, share study materials to students, etc. Finally, the student will have facilities to enroll into courses, view the daily schedule, view assignment deadlines, post assignments to teachers, etc. The major challenges we expect to face are managing the wide variety of requirements of three different kinds of users and managing the complicated internal relationships between the users and other components of our software.

# **1. INTRODUCTION**

‘University Portal’ is a proposed solution to improve the efficiency of teaching and learning in universities. With the wide range of streams involved, the courses offered and the students enrolled per stream, managing a university can become overwhelming. So, the first kind of user of the software is the university administration. Activities such as managing streams, updating syllabi, assigning teachers to classes, generating optimal class schedules etc. will be seamless through the portal. The second kind of user is the teacher. For the teacher, the software will provide facilities to manage class schedule, assign work to students after each lecture, provide the weekly plan to the students, provide date of assessments, accept the assignments from students and keep track of the syllabus covered in each class he/she teaches. The final kind of user the portal will aid is the student. The student can keep track of their subjects, get assignments from their teachers, find details of any project that they have been assigned, etc. The aim of the portal is to do the heavy lifting of education management so that everyone at a university can focus on education instead.

## **1.1 Problems and Objective**

### **1.1.1 Engineering Problems**

Some problems that we expect to come across during this project are as follows:

- i) Since it involves multiple users with varying levels of access and entirely different use cases, specifying the requirements correctly will be tough.
- ii) The functional relationships between different entities of the system are so complex and intertwined that it will be difficult to break the software into modules.

iii) Since the development team has never done this kind of project before and since the team is not well versed in the dogmas of software engineering practice, managing resources, analyzing bugs, collaborating on the project, etc. will be a headache too.

### **1.1.2 User Problems**

#### **I. Problems of Students**

- Dependency on the class representative for their education management, who is almost certainly overworked.
- Difficulty in keeping track of real-time change in class schedule.
- Non-existent notification system regarding important events such as exams, administrative notices, programs conducted in the university, etc.
- Ineffective distribution system of assignments and lab works.

#### **II. Problems of Teachers**

- Ineffective management of class plans and schedules.
- Difficulty in keeping track of all the classes and their course progress.
- Non-existent assignment evaluation system.

#### **III. Problems of Administration**

- Difficulty in keeping track of a large number of streams and the corresponding students.
- Ineffective notice distribution system.
- Non-existent data collection system for evaluation of syllabi essential for improving the quality of education.

**Objectives:**

1. To provide an online platform to manage education at universities.
2. To make the education system transparent.
3. To build a bridge between administrators, teachers and students.

**1.2 Scope of the work**

As its name suggests, ‘University Portal’ is designed considering the hierarchy of managing structure in Universities. So, obviously, the scope of the software are educational institutes such as Universities, Affiliated Colleges, etc. The software will be general so that it can be adapted according to the structure of the institute.

**1.3 Related Work**

The following are a few of many online university portals out there which are, more or less, similar to this project:

**i) MIT OpenCourseware(<https://ocw.mit.edu>)**

MIT OpenCourseWare is a web-based publication of virtually all MIT course content. OCW is open and available to the world and is a permanent MIT activity.

**ii) Harvard Education Portal(<https://my.harvard.edu/>)****iii) University of London International Programmes(<https://my.londoninternational.ac.uk/>)****iv) University of the People Online Portal(<https://your.uopeople.edu/>)**

## 2. LITERATURE REVIEW

The primary purpose of an **information system** is to manage and provide access to a database of information. Issues in information systems include security, usability, privacy, and maintaining data integrity.

Many teachers argue that using interactive software systems to support education can lead to both improved learner motivation and a deeper level of knowledge and understanding in students. However, there is no general agreement on the ‘best’ strategy for computer-supported learning, and teachers in practice use a range of different interactive, web-based tools to support learning. University Class Portal is an interactive web app also aimed to improve learning.

Account of students, teachers and admin (Department and Administration)

Admin will add/remove courses and manage Courseware.

Teacher will give course plan inside courses, give assignments or projects; weekly, mark attendance of students, make alterations on course plans

Teacher will also give out result of class tests, performance, assessments through their account

Students will get to select the courses they study (to be approved by admin), view assignments and submit, check their progress.

Students receive notification on their account when new assignment is added, or new result is published by teacher.

Students can use their account as an academic profile to record the classes they have been to, grades they have received and details of projects of the respective courses.

Courseware helps to organize courses taught in a university (its features, descriptions and highlights).

Professors can monitor student’s class and lab activities, keep track of reports, assignments and assessments

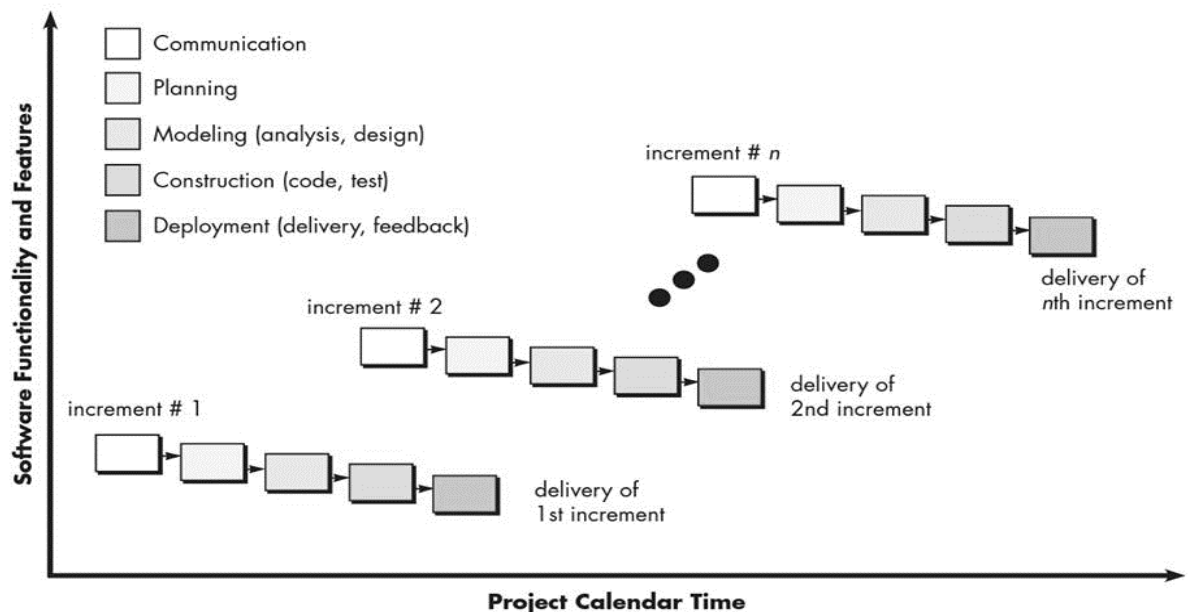
Discussion Forum helps to clarify doubts, extends classroom time, increases interaction time

Daily Schedule, Calendar Events, Notice published by department, administration and student unions are stacked in a section so that students do not miss out on any important details they ought to know.

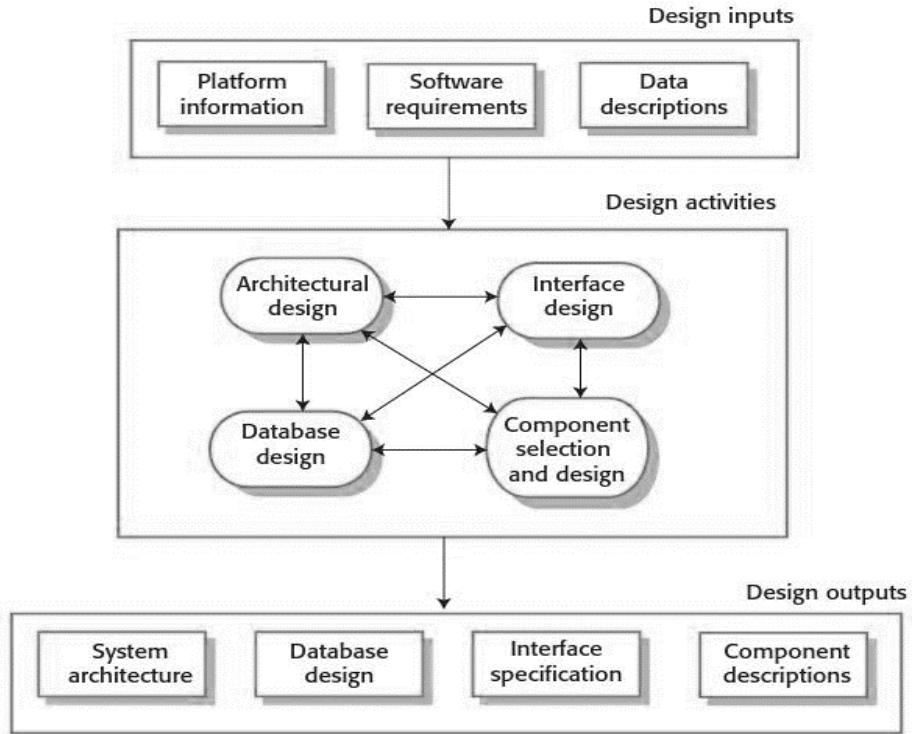
### 3. PROPOSED METHODOLOGY

#### 3.1 Process Model: Incremental Model

The incremental model is a hybrid of waterfall and agile models of software development. This model is powered by the design efficiency of waterfall model and the testing and prototyping facilities of agile development. We plan to design and implement the core features of our project in the first installment, then after testing and analysis, move on to the remaining features to complete the project. The most important feature of the incremental model is the flexibility it offers. Users are often unsure about the more abstract features of the software and thus may change requirements with time. The core requirements however, are less likely to be changed by the user. So, the software can be developed with extensive interaction with the users.



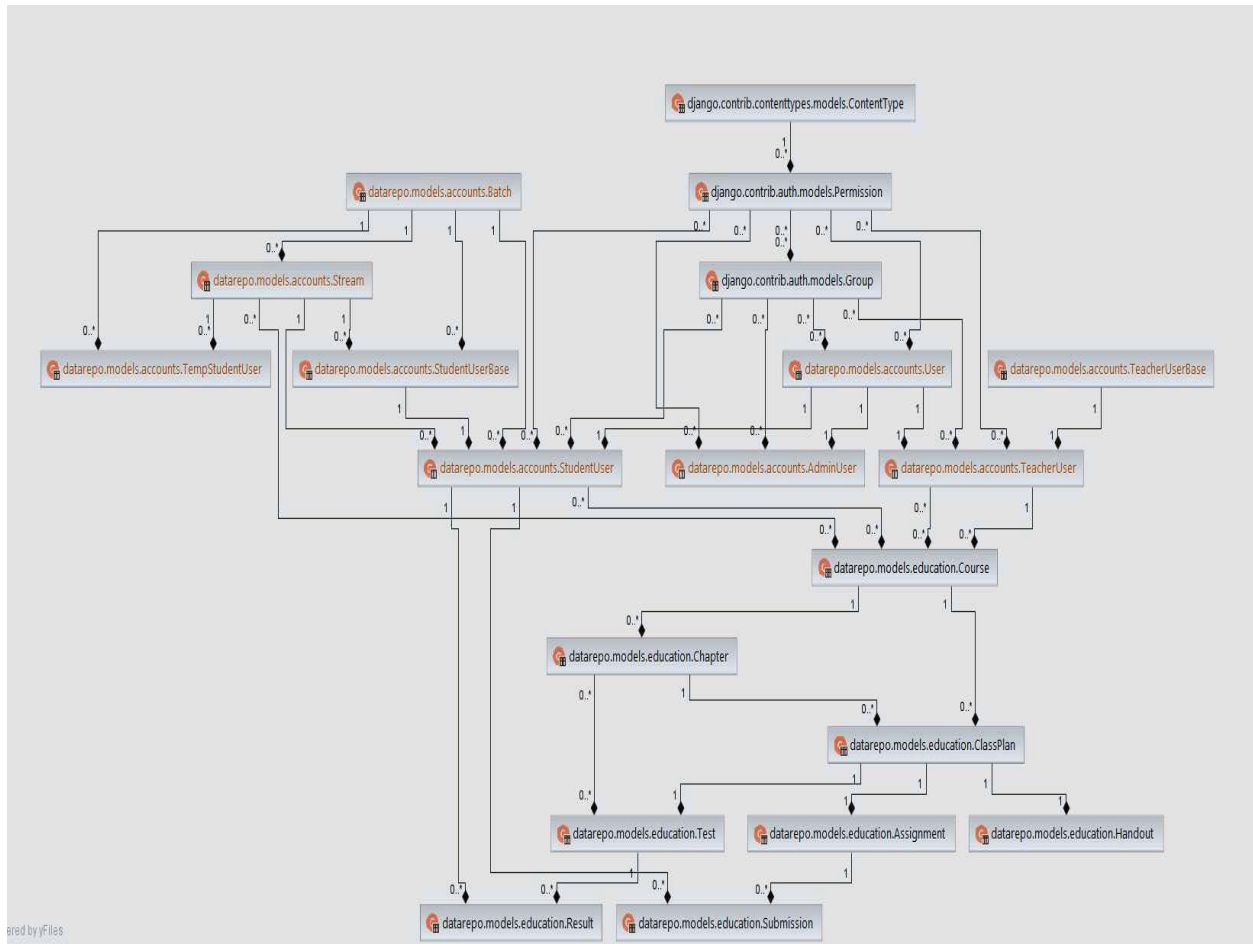
**The incremental development model**



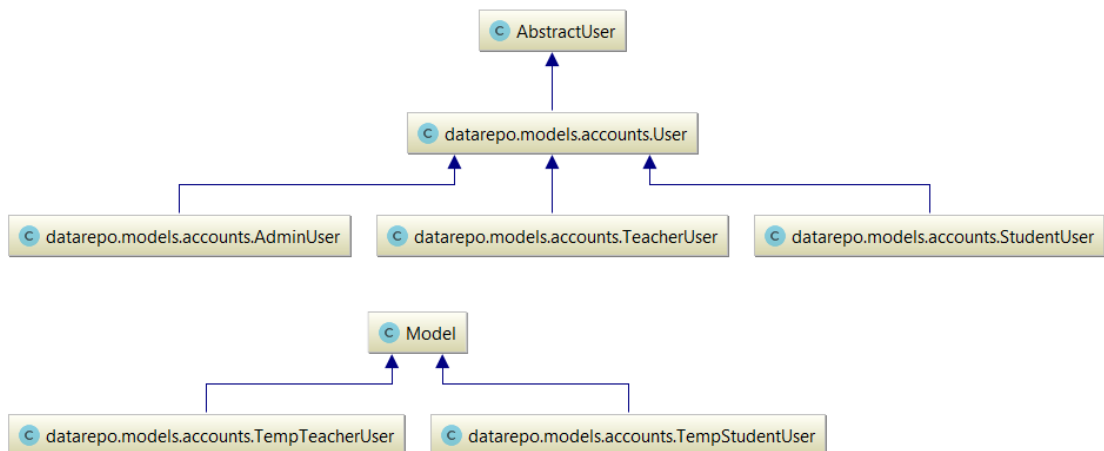
**General Design Process**

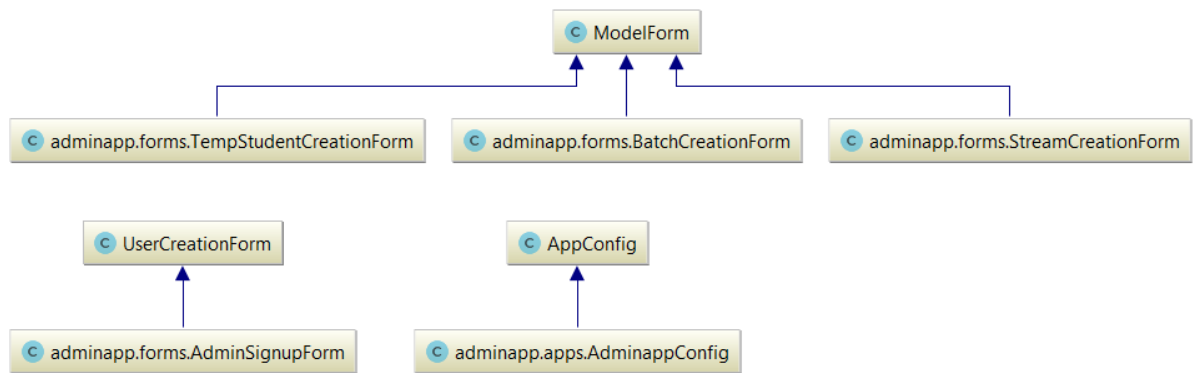


## Entity Relationship Diagrams:



## Class Hierarchy diagrams:





## 3.2 Tools and Techniques that will be used

### Platform:

Web

### Technology used:

#### I. Implementation technology

HTML5, CSS3, Twitter Bootstrap, Javascript (ES5), JQuery, MorrisJs, Python Django

#### II. Technology for software planning, design and collaboration

Axure RP, StarUML, Draw.io, Github

## 3.3 Performance Parameters and System Validity

### Performance requirements:

- i) Expected users: 1000-10000
- ii) CPU usage: Nominal
- iii) Memory requirement: Nominal
- iv) Connectivity: Essential
- v) Response Time: no more than 1 second
- vi) System Dependability: Crucial

- vii) Security: Crucial
- viii) Data validity: Crucial
- ix) CSRF protection: Crucial
- x) Cross platform support: Essential
- xi) Application extendibility: Nominal

#### **Data and system validity:**

Since we plan to use the industry standard django APIs, data validity and consistency are criteria that we need not worry about. These are core features of Django API. Furthermore, we plan to use Django's test suite to perform specific unit tests to ensure that each part of the software system works properly without loopholes that may compromise user experience or security. A common example of data validation is the login/logout system. This feature is a part of core django and is implemented internally. The API exposes the user object only if the data entered by the user is valid, otherwise no user instance will be available. Similarly, the management of http session protocol is also handled by the API so that safe login-logout routines are implemented. Another issue that web applications face is cross-site referencing by hackers. Django prevents this by blocking form submission without a CSRF token. This token is auto-generated by django and forms without the token cannot post information onto the server.

## **4. COST ESTIMATION AND SCHEDULE**

The following is the planned schedule for the project. Note that it is meant to be flexible and will be subject to change.

01/12/2017—05/12/2017 Communication

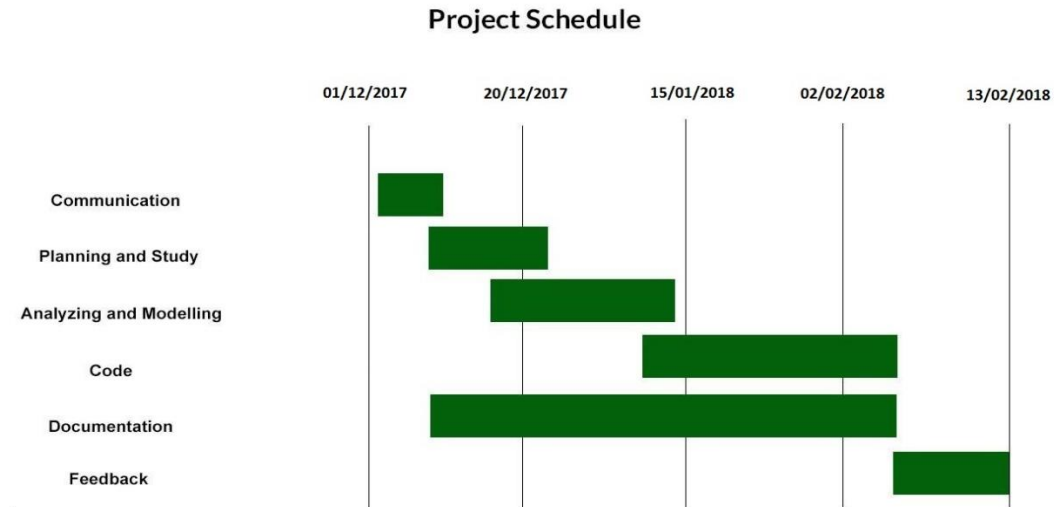
06/12/2017—20/12/2017 Planning

21/12/2017—15/01/2018 Design

16/01/2018—02/02/2018 Implementation

03/02/2018—05/02/2018 Initial Testing

06/02/2018—13/02/2018 Revision



## 5. CONCLUSION

World's topmost universities have their own consolidated online platform to access information and record transactions related to academic advising, teaching and grading, student enrollment and financials, training sessions and workflow enabled administrative processes. We are very much familiar with MIT's Open Courseware where course contents and possibly everything related to that course is very well organized and available to view online. At a moment where everything is switching towards an online platform, it's high time we make amendments to our way of organizing and planning courses taught in university. Education is the essence of Smart City. A city can be smart if people living in it are well educated. This project is a small step towards developing and organizing things required in education sector of a Smart City.

## 6. REFERENCES

- Sommerville, I., & Christensen, M. J. (2003). Software engineering. New York: IEEE computer Society.
- (2018, January 18). Simple is Better Than Complex. Retrieved January 23, 2018, from <http://www.simpleisbetterthancomplex.com/>