PROJECT REPORT
Assignment No. 4

Members.	**Roll No.**
Abhashri Deshmukh	202IS002
Sayali Deo	202IS009
Tridiv Kumar Rabha	202IS027
Karan Singh Bisht	202CS015

Feasibility Study

Feasibility study is carried out based on many purposes to analyze whether software products will be right in terms of development, implantation, contribution of project to the organization etc. Types of Feasibility Study includes Economic Feasibility, Technical Feasibility, Operational Feasibility, Legal Feasibility, Schedule Feasibility. Our focus of this report will be Technical Feasibility and Schedule Feasibility as other types are redundant for the purpose of this assignment.

1. Technical Feasibility :
    a. Software Requirements :
        i. Browser : Google Chrome, Firefox, etc.
        ii. Code Editor : Visual Studio Code, Sublime Text, etc.
        iii. Version Control : Github for Desktop, Gitkraken, etc.
        iv. Technology Stack :
            1. Frontend : HTML, CSS, Jinja 2 Templates
            2. Backend : Python Flask Framework.
            3. Database : SQLAlchemy - A Python SQL toolkit
2. Schedule Feasibility :

Week	Date	Project Plan
1	14/12/20	Assignment 4 issued
2	21/12/20	Setup Github Repository, added members and prepared project plan
3	28/12/20	Completed Backend and Admin Interface
4	4/12/20	Frontend Work Ongoing
5	10/12/20	Completed Project Report

Software Development Life Cycle SDLC

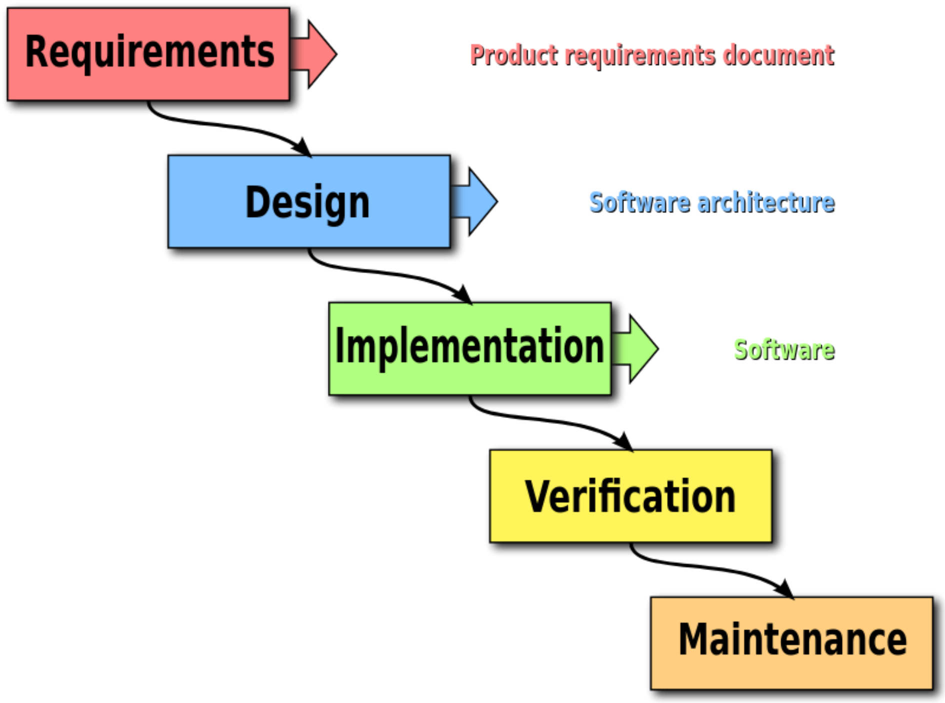
SDLC is a process followed for a software project, within a software organization. It consists of a detailed plan describing how to develop, maintain, replace and alter or enhance specific software. The life cycle defines a methodology for improving the quality of software and the overall development process.

In this project, we have used the **Waterfall Model.**

Reasons for the choice of the model :

This is because it is a great model for small-sized applications which do not need multiple rounds of software development cycles. Also, the product requirements are stable and well-understood at the very beginning of the project. The Waterfall model is very easy to manage due to the rigidity of the model, which suits our purpose.

Stages of the Waterfall Model :



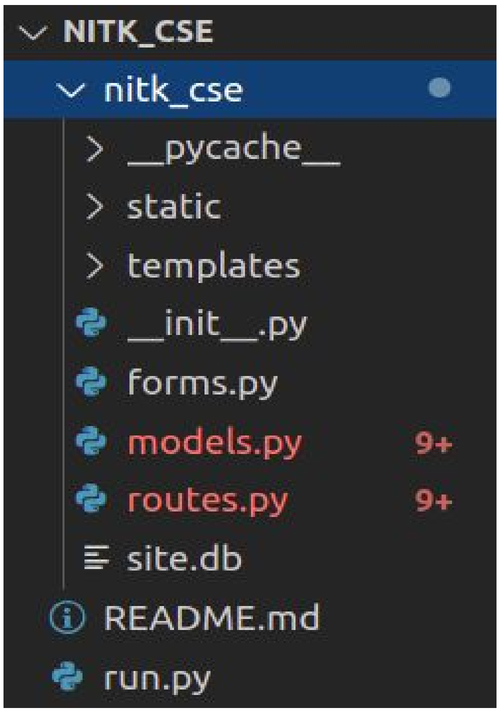
-Img src : Wikipedia

1. Requirement Gathering -
    - All possible requirements of the system to be developed are captured in this phase and documented in a document.
    - Carried out in weeks 1 and 2 with information from https://cse.nitk.ac.in/

2. Design and Prototyping -

The requirement specifications from the first phase are studied in this phase and the system design is prepared.
This system design helps in specifying hardware and system requirements and helps in defining the overall system architecture.
Carried out in week 2.
Directory Structure -

- The Main Folder named NITK_CSE
- Contains 1 subfolder : nitk_cse and 2 files : readme.md and run.py
- Within nitk_cse we have 3 folders :
    - static - containing images, scripts and stylesheets
    - templates - containing HTML pages
    - __pycache__ : containing python logs.
- Nitk_cse also contains 5 files :
    - __init.py__ : Python file contains various flask library imports.
    - forms.py : Python file contains all forms displayed in HTML.
    - models.py : Python file contains all database models(i.e. tables)
    - routes.py : Python file contains all routing information from forms to database.
    - site.db : The SQL Database binary file.

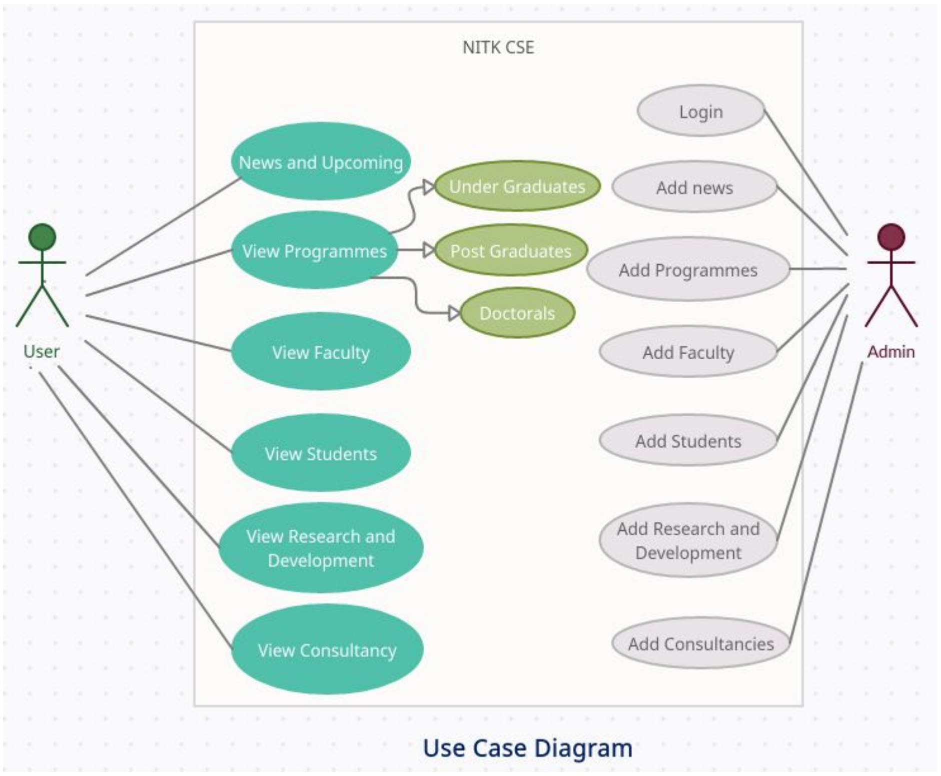


```
NITK_CSE
  nitk_cse
    __pycache__
    static
    templates
    __init__.py
    forms.py
    models.py      9+
    routes.py      9+
    site.db
  README.md
  run.py
```

UML Design

The Unified Modeling Language (**UML**) is a general-purpose, developmental, modeling language in the field of software engineering that is intended to provide a standard way to visualize the design of a system.

Below is the **Use Case Diagram** of our system



3. Implementation :
    - With inputs from the system design, the system is first developed in small programs called units, which are integrated in the next phase.
    - Each unit is developed and tested for its functionality, which is referred to as Unit Testing.
    - Was implemented in weeks 3 and 4.
    - The system backend is developed using ***Flask*** which is a micro web framework written in Python. It is classified as a microframework because it does not require particular tools or libraries. It has no database abstraction layer, form validation, or any other components where pre-existing third-party libraries provide common functions.
    - The lightweight, easy-to-scale nature of Flask makes it suitable for our application needs.
    - Flask Documentation : https://flask.palletsprojects.com/en/1.1.x/
    - The system database is built using ***SQLAlchemy*** is the Python SQL toolkit and Object Relational Mapper that gives application developers the full power and flexibility of SQL.

- SQL was chosen as the database as it suits well to the relational model of our application.
- SQLAlchemy Documentation : https://docs.sqlalchemy.org/en/13/orm/tutorial.html

4. Verification/Testing :
    - All the units developed in the implementation phase are integrated into a system after testing of each unit.
    - Post integration the entire system is tested for any faults and failures.
    - Unit testing was done throughout weeks 3 and 4. The entire system was tested by the end of week 4.
    - Testing was simply carried out manually by passing valid and invalid data.

5. Maintenance :
    - There are some issues which come up in the client environment.
    - To fix those issues, patches are released.
    - Also to enhance the product some better versions are released.
    - Maintenance is done to deliver these changes in the customer environment.

View Project Source Code on Github : https://github.com/sayalideo/NITK_CSE