# Sprint Planning

Tasks and Test scenarios

## Sprint 1 Goal: Architecture design sprint

* Come up with technologies, data model and architecture solution for the project

**Task1:** Discuss technologies, data model and architecture solution to be used for the project.

## Sprint 2 Goal: Ability for Admin to login to admin system and update password

* As an admin user, I should be able to login to admin page

**Task1:** Create a user login form for Admin to enter username and password and login

**Task2:** Add a validation when login button is clicked to verify username and password goes to database

**Task3:** Display error when login validation fails.

## Sprint 3 Goal: Ability for Admin to add class ,teachers ,Subjects , Students into the DataBase

* As an admin user, I should be able to add Class
* As an admin user, I should be able to add teachers
* As an admin user, I should be able to view list of teachers
* As an admin user, I should be able to add name of subject
* As anAdmin user, I should be able to add student information into dataBase

# User Flow Diagram

## 

# Technologies Used:

|  |  |
| --- | --- |
| **IDE** | Eclipse |
| **Programming languages** | Java, HTML |
| **Database server** | MySQL |
| **Project** | Maven |
| **Code repository** | GitHub |
| **Web application** | JSP |
| **ORM** | Hibernate |
| **Programming language class** | Java servlets |

# Core concepts used:

* Classes
* Interfaces
* Objects
* Methods
* Annotation
* Dynamic variables
* Data types - int, string, date, float
* Access modifiers
* Getters/setters
* If else statements
* Exception handling - Try catch exceptions
* List
* Comparison and Logical operators
* Java keywords - public, private, super, try, catch, if, else, import, int, void, new
* Hibernate
  + Configuration
    - Config file: cfg.xml file
  + SessionFactory
    - Opensession
    - Closesession
  + Session
    - Session.close
    - Session.save
    - Session.beginTransaction
    - getTransaction
    - commit
    - Annotations
  + Dynamic webpages using servlet
  + doGet
  + doPost
  + request
  + response
  + dispatcher
  + getParameter
* JSP
  + Dynamic webpages using JSP