# SARVAJANIK UNIVERSITY
## Sarvajanik College of Engineering and Technology
### Master of Computer Applications

**MCA Semester IV**

**Subject Name:** Software Project **Subject Code:** MTCA16401

**Type of course:** Application

**Learning Objectives:**

- To solve industrial (or society or research) problems.
- To plan, schedule, and monitor the software project
- Development, coding, and testing of a large project cohesively.
- Documentation of project

**Prerequisite (if any):**

- Software Engineering
- Programming / Coding language
- Relational Database Management System

**Rationale: (should also include Description of the relevance of this course in the Program)**

**Teaching and Examination Scheme:**

| TEACHING SCHEME | | | | Theory Marks | | | Practical Marks | | Total |
|---|---|---|---|---|---|---|---|---|---|
| L | T | P | C | TEE | CA1 | CA2 | TEP | CA3 | |
| 0 | 0 | 30 | 15 | 0 | 0 | 0 | 600 | 300 | 900 |

**CA1:** Continuous Assessment (assignments/projects/open book tests/closed book tests **CA2:** Sincerity in attending classes/class tests/ timely submissions of assignments/self-learning attitude/solving advanced problems **TEE:** Term End Examination **TEP:** Term End Practical Exam (Performance and viva on practical skills learned in course) **CA3:** Regular submission of Lab work/Quality of work submitted/Active participation in lab sessions/viva on practical skills learned in course

**Guidelines for Project**

- It is recommended that the team should be about 2-3 students.

- The project should be free from plagiarism of any kind.
- It is mandatory to include self-declaration in support of non-plagiarism project and project report.
- Internal guides (i.e. The regular faculty members) must be allocated to projects.
- Project plan along with the division of work amongst teammates would have been prepared and got approved within a maximum of 5 days of the start of the project.
- Coding standards should be followed meticulously. At the minimum, the code should be self documented, modular, and should use the meaningful naming convention.
- It is advisable that object-oriented methodology is used with the reusability of classes and code, etc..
- The output reports must include MIS reports, if applicable.
- The documentation should include a chapter on "Learning during Project Work", i.e. "Experience of Journey during Project Duration".
- It is strongly recommended that Data structure/Database design is included in the report. At least portions of code (preferably full code) are mandatory. The student may be asked to write the code related to the project during the examination.
- If a student is compelled to follow certain instructions (by the external, i.e. organization's Guide) which he/she does not agree to, such a student must prepare a supplementary report to document his/her version and present it to the examiners if such a need arises.
- Internal guides (i.e. The regular faculty members) must devote the time allocated as per the timetable to guide the students for the project. The time allocation will be in accordance with the scheme for the 6th semester project as given.
- Internal guides should preferably visit external guide to track the project
- Project document should be printed on both sides of paper.

**Accomplishments of the student after completing the course:**

- Doing the project will enable the student to go through rich experience in developing large projects. Such an experience will include encountering various technical issues, finding sources to resolve the issues and finally finding the solution of all these issues satisfactorily.
- Thinking analytically, analyzing, and synthesizing requirements and complicated information for getting a good comprehension of the solution methodology to be adopted.
- Ability to document and write well.
- Organizing the time effectively.

- Working with teammates and generating substantial output of the efforts.
- It will prepare the students for analyzing and programming for industrial problem and large projects work in future

**Documentation:**

- The project has to be well-documented in the form of a Project Report (at least 50 pages consisting of the design, data dictionary, source code, screenshots, etc.).
- Format: Print out on both sides of the page with single line spacing. Use Times New Roman of size 10 for normal text.
- Suggested table of contents is given below.

## TABLE OF CONTENTS

1. Introduction
    1.1. Existing System
    1.2. Need for the New System
    1.3. Objective of the New System
    1.4. Problem Definition
    1.5. Core Components
    1.6. Project Profile
    1.7. Assumptions and Constraints
    1.8. Advantages and Limitations of the Proposed System
2. Requirement Determination & Analysis
    2.1. Requirement Determination
    2.2. Targeted Users
3. System Design
    3.1. Use Case Diagram
    3.2. Class Diagram
    3.3. Interaction Diagram
    3.4. Activity Diagram
    3.5. Data Dictionary
4. Development
    4.1. Coding Standards
    4.2. Screen Shots
5. Agile Documentation
    5.1 Agile Project Charter
    5.2 Agile Roadmap / Schedule
    5.3 Agile Project Plan

5.4 Agile User Story (Minimum 3 Tasks)
5.5 Agile Release Plan
5.6 Agile Sprint Backlog
5.7 Agile Test Plan
5.8 Earned-value and burn charts
6. Proposed Enhancements
7. Conclusion
8. Bibliography

**Evaluation Parameters :**

- Evaluation of the projects would be done considering the framework available at the Institute. The main parameter of assessment would be the ability of the students to code.
- Though the project and domain specific knowledge would be assessed for, the evaluation would predominantly depend on the students' ability to explain, modify or revise code.
- Coding standards should have been implemented.
- Though the project would be evaluated for the entire team, the examiner should emphasize on the contribution of each team member in the project development
- Total Marks ( 900 = 600 External + 300 Internal )

| | |
|---|---|
| Documentation (more specifically, Correctness and completeness of UML diagrams, and relationship between Class diagram & Database structure) | 100 |
| Explanation of Analysis & Design | 200 |
| Explanation of Code (To test the ability to explain how to code few functionalities used in the project) | 200 |
| Presentation | 100 |

**Course Outcomes:**

| Sr. No. | CO statement | Marks % Weightage |
|---|---|---|
| CO-1 | Understand software project characteristics and different stages of a development process. | 10% |
| CO-2 | Analyze the techniques for project planning, scheduling and feasibility | 10% |
| CO-3 | Ability to conceptualize the project using system design | 10% |

| | | |
|---|---|---|
| CO-4 | Develop a model using programming language | 40% |
| CO-5 | Understand the effectiveness of agile methodology for software project development | 10% |
| CO-6 | Develop an ability to present the software project. | 20% |

**Mapping with POs:**

| | PO 1 | PO 2 | PO 3 | PO 4 | PO 5 | PO 6 | PO 7 | PO 8 | PO 9 | PO 10 | PO 11 | PO 12 | PS O1 | PS O2 | PS O3 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| **CO-1** | 3 | 3 | 1 | 3 | 1 | 1 | 1 | 2 | 2 | 2 | 2 | 2 | | | |
| **CO-2** | 3 | 3 | 1 | 3 | 1 | 0 | 0 | 2 | 2 | 2 | 2 | 2 | | | |
| **CO-3** | 3 | 3 | 1 | 3 | 1 | 0 | 0 | 2 | 2 | 2 | 2 | 2 | | | |
| **CO-4** | 3 | 3 | 3 | 2 | 1 | 0 | 0 | 2 | 2 | 2 | 2 | 2 | | | |
| **CO-5** | 3 | 2 | 1 | 2 | 1 | 0 | 0 | 2 | 2 | 2 | 2 | 2 | | | |
| **CO-6** | 3 | 2 | 1 | 1 | 1 | 0 | 0 | 2 | 2 | 3 | 2 | 2 | | | |
| **Rationale*** | | | | | | | | | | | | | | | |

**Rationale*: Explaining why it is matching this particular program outcome**