CSI-4999 Senior Capstone Project

Course Syllabus

Winter Session Jan – April, 2018
MW 1:00PM - 2:47AM
HHB 1005
Dr. Nilesh Patel
Associate Professor
Computer Science and Engineering Department
Oakland University
Email: *npatel@oakland.com*
Office: 248.370.2247

Course Catalog Description:

Team oriented senior project to synthesize the knowledge and skills gained in the CS/IT curricula. Written and oral reports are required in addition to working demonstration. The course is cross listed.

Prerequisites

– CSE 364, CIT/CSE 337, CIT/CSE 345. Requires a major standing in IT/CS

Learning Objectives

These classes satisfy the General Education requirements for a Capstone Experience. The student will demonstrate General Education Learning Outcome:

Capstone Experience

The student will demonstrate:

- Appropriate uses of a variety of methods of inquiry and recognition of ethical considerations that arise
- Ability to integrate the knowledge learned in general education and its relevance to the student's life and career

In addition, student will demonstrate the General Education Cross-Cutting Capacities:

- **Effective communication** - Exhibited through written and oral communication, designs, implementations and presentations
- **Critical thinking** - Developing novel and innovative solutions to technological problems
- **Social awareness** - Considering and allowing for multicultural aspects of the global marketplace as technical solutions are sought that are socially responsible and culturally appropriate

- **Information literacy** - Finding, evaluating and utilizing information to assess and develop innovative and creative solutions to technical solutions that are socially responsible and culturally appropriate

Course Objectives

At the end of this class, students shall be able to **demonstrate**

a) An ability to apply knowledge of computing and mathematics appropriate to the discipline. (CIT/CSE)
b) An ability to analyze a problem, and identify and define the computing requirements appropriate to its solution. (CSE/CIT)
c) An ability to design, implement and evaluate a computer-based system, process, component, or program to meet desired needs. (CIT/CSE)
d) An ability to function effectively on teams to accomplish a common goal. (CSE/CIT)
e) An understanding of professional, ethical, legal, security, and social issues and responsibilities. (CSE/CIT)
f) An ability to communicate effectively with a range of audiences. (CIT/CSE)
g) An ability to analyze the local and global impact of computing on individuals, organizations and society. (CSE/CIT)
h) Recognition of the need for, and ability to engage in, continuing professional development. (CSE/CIT)
i) An ability to use current techniques, skills, and tools necessary for computing practice. (CSE/CIT)
j) An ability to use and apply current technical concepts and practices in the core information technologies. (CSE/CIT)
k) An ability to identify and analyze user needs and take them into account in the selection, creation, evaluation and administration of computer-based systems. (CSE/CIT)
l) An ability to effectively integrate IT-based solutions into the user environment. (CSE/CIT)
m) An understanding of best practices and standards and their application. (CSE/CIT)
n) An ability to assist in the creation of an effective project plan. (CSE/CIT)

Office Hours:

Time: Wednesday 3:00PM to 3:30PM or By Appointment

Location: EC 524

Team Formation Guideline:

Each project team will not have no less then 5 or more then 7 members. You must ensure that team is comprised of complimentary skills. Computer Science and Information Technology projects require requirements elicitation, analysis and design, implementation and quality assurance skills. Over the period of project development, each member must demonstrate skills in number of areas as per various efforts areas

identified in **Efforts Areas** and show accomplishment of the course outcomes as detailed above.

Project Development Guideline:

We will follow an agile software development process. Each team will divide the feature set they plan to develop into 1 weeks long sprint. Team members must continuously capture various project artifacts. All artifacts must be quantitatively measured as defined in following **Effort Areas** which maps to demonstrable objectives.

- Tools, technology and Algorithm Development **[h, i]**
- Project vision, Glossary of terms, System requirements **[b]**
- User stories, Scenario definition and Use case Requirements **[b, g, k]**
- System security, data protection and ethical issues **[e, g]**
- Class or Entity Relationship Diagram**[c, i, j]**
- Interaction / State Diagram**[c, i, j]**
- System Architecture**[c]**
- User Interface Design**[b, i]**
- Unit and System Testing and test data selection **[a, j]**
- Implementation, Build programming, Server Setup and Configurations **[a, i, j]**
- Algorithms Design and Development **[a, h]**
- Project Plan Update**[i, n]**
- Power point Preparation and Presentation **[f, i]**
- Test Execution and Bug Reporting**[i]**
- System deployment **[l]**
- Use of best practices and standards**[m]**

For each sprint, team members will divide responsibilities with non-overlapping efforts such that produced artifacts measure individual contribution and effort. Software activities are highly coupled and dependent on each other. Members who are responsible for requirements must complete their activity ahead of time to ensure that dependent team members get enough time for completion. To ensure planning of such dependency, all team must generate quality artifact to help team succeed.

Your work will be assessed in by-weekly deliverables. Each student must keep a working journal and update Sprint Logbook of their contributions and show the proof that they have demonstrated desired skills as outlined in objectives [A to N]. You can download the **Sprint Logbook template** from **Moodle**. While working in harmony, real world projects do implement freedom and control to ensure that each member delivers the best artifact without influence of the other. Each team member is expected to give minimum between 16 to 24 hours per two week of their time working on the project. **Weak deliverable by any team member will result in 0 grade for that week.**

Sprint Management Guideline

All team members must collaborate using GitHUB and must share repo with class TA. Each team member must work in his/her own branch for documentation, implementation, testing etc. At the end of each Sprint, branches must be merged to the main branch. You must download the branch and submit.

- Every member must show evidence of around 16-24 hours of work per sprint.
- Your sprint grade will be according to your efforts and evidence of your work submitted individually which shall include –
    - Sprint Logbook.xls file updated for that sprint
    - All supporting evidences including (documents, research report, source code etc.)
    - **Do not submit entire project tree. You can only submit your portion of work. Failure to do so will result in penalty.**

Course Assignments & Grading

Individual Efforts Grade:	**Grade**
• *Sprint Logbook.xls, all evidences (documents, zipped version of code)*	*0.0-2.0*
• *Percent Course Objectives Accomplishment*	*0.0-0.5*

Group Efforts Grade (Weighted By Peer Evaluation/Contribution)

• *Project poster and documentation (Final version)*	*0.0-0.5*
• *Competition Judges Grade*	*0.0-1.0*
○ *Top three winning projects – 1.0*	
○ *Forth to Last project – Between 0.1 – 0.9*	

*Your individual grade for **Group Effort** category will be calculated based on peer evaluation. If each member of the team has given equal efforts as per peer evaluation sheet, all team members will receive same points. If not, the member receiving poor evaluation will receive lower points based on team evaluation.*

Class schedule

Sprint	Date	Completed Activities
0	1/3/18	Team formation and Project Allocation
1	1/10/18	Sprint presentation, demonstration, risk and challenges discussion, mitigation plan, next sprint objectives. Must complete – • Project requirements • Project plan • Tool chain, language, Config. Management, server setup, development environment,…etc.
2	1/22/18 1/24/18	Sprint presentation, demonstration, risk and challenges discussion, mitigation plan, next sprint objectives. Must complete – • User account creation, registration, role management, profile update, locking/unlocking of user accounts, email verification • Must complete all user stories for next spring • Must complete all UI/UX for next sprint • Must create all QA tests for user authentication system
3	2/5/18 2/7/18	Sprint presentation, demonstration, risk and challenges discussion, mitigation plan, next sprint objectives. Must complete – • Must complete all user stories for next spring • Must complete all UI/UX for next sprint • Must complete QA for all prior sprint and create new tests cases for current
4	2/26/18 2/28/18	Same as above
5	3/12/18 3/14/18	Same as above
6	3/26/18 3/28/18	Same as above
7	4/10/18 4/12/18	Dry Run
8	4/19/18	Final Competition in front of Judges

Project Documentations:

1. Project vision
    1.1. Backgrounds
    1.2. Socio-economical Impact, Business Objectives, and Gap Analysis
    1.3. Security and ethical concerns

1.4. Glossary of Key Terms
2. Project Execution and Planning
    2.1. Team Information
    2.2. Tools and Technology
    2.3. Project Plan
    2.4. Best standards and Practices
3. System Requirement Analysis
    3.1. Function Requirements
    3.2. Non-functional Requirements
    3.3. On-Screen Appearance of landing and other pages requirements.
    3.4. Wireframe designs
4. Functional Requirements Specification
    4.1. Stakeholders
    4.2. Actors and Goals
    4.3. User stories, scenarios and Use Cases
    4.4. System Sequence / Activity Diagrams
5. User Interface Specifications
    5.1. Preliminary Design
    5.2. User Effort Estimation
6. Static Design
    6.1. Class Model
    6.2. System Operation Contracts
    6.3. Mathematical Model
    6.4. Entity Relation
7. Dynamic Design
    7.1. Sequence Diagrams.
    7.2. Interface Specification
    7.3. State Diagrams
8. System Architecture and System Design
    8.1. Subsystems / Component / Design Pattern Identification
    8.2. Mapping Subsystems to Hardware (Deployment Diagram)
    8.3. Persistent Data Storage
    8.4. Network Protocol
    8.5. Global Control Flow
    8.6. Hardware Requirement
9. Algorithms and Data Structures
    9.1. Algorithms
    9.2. Data Structures
10. User Interface Design and Implementation
    10.1. User Interface Design
    10.2. User Interface Implementation
11. Testing
    11.1. Unit Test Architecture and Strategy/Framework
    11.2. Unit test definition, test data selection

11.3. System Test Specification
11.4. Test Reports per Spring
12. Project Management
12.1. 11.1 Project Plan
12.2. 11.2 Risk management
13. References

Preparing for class presentations

- Presentation must be prepared using MS Power Point
- All students must attend and present their project contribution. **Student not attending and presenting will receive 0 grade point for that week, unless another team member presents his/her work.**
- Each team will have between 12 to 15 minutes for presentation

Academic Conduct

Students are expected to read, understand and comply with the Academic Conduct Policy of Oakland University, found in the schedule of Classes and in the Undergraduate Catalog. Suspected violations will be taken before the Academic Conduct Committee. Students found guilty of academic misconduct in the course will receive a grade of 0.0 in addition to any penalties imposed by the Academic Conduct Committee.

Add/Drops

The University add/drop policy will be strictly followed. It is the student's responsibility to be aware of the University deadline dates for dropping the course.

Special Considerations

Students with disabilities who may require special considerations should make an appointment with campus Disability Support Services. Students should also bring their needs to the attention of the instructor as soon as possible.