Data Foundation Systems

Vikram Pudi Gundimeda Venugopal

Data Foundation Systems Course Flow (1/2)

- Course Overview (Vikram) - Week 1-1
- Data Foundation Architecture overview (Venu) Week 1-1
- Data Foundation System Demo (Dr. Raghunath) + Developer Guidelines and Coding practices (Amey / Ritvik / Shaantanu) Week 1-2
- Data Foundation Projects on offer (Venu) + Workshop on Data Modeling and UML Week 2-1 (Ritvik Aryan)
- **Project Selection by Week 2-2 (11 August)**
- Project use case: Submit Requirements (on GitHub Project Page)

[Metrics: Level of Understanding of the problem. Requirements Coverage, Functional / Non-functional Requirement, Representation (e.g., Usecase/Activity diagrams]

- Project Execution Plan (Approach, Components & work load distribution) / Schedule + Project use case: Test Strategy / Test cases after Requirements are validated and approved (on GitHub project page)

[Metrics: Plan / Approach – How ?, Break of work + coordination, Workload distribution, Test Approach, Test Coverage]

Data Foundation Systems Course Flow (2 / 2)

- Project use case: Architecture / Design Approach and Detailed design (Review ideas and make any course corrections)

[Metrics: Architecture / Design Approach – How ?, Design alternatives considered, Design constraints, Component details, Internal / External Component Integration, Design Representation (diagrams), Advantages of the design]

- Project Intermediate Demo (on laptop) + Project Code Organization (TAs / Amey / Peer Code Review / Venu)

[Metrics: Feature completion , Code Structure / Quality / Standards conformance, Requirements / Design Conformance]

- Project Integration: (Integration with Data Foundation Core System) + Testing + Make it work in Production + (Recorded Demo) Ready to Go Live test (TAs / Peer Code Review / Venu / Vikram)

[Metrics: Feature completion, Deployment scripts, Making it work in Data Foundation Environment]

- Project Demo: (A Working demo on IHUB desktop/server + Deployed in production) (Venu / Vikram)

[Metrics: Feature Completion, Requirements/ Architecture/ Design / Test Highlights, Traceability (Requirements → Design → Code → Test → Demo)]

Project Evaluation: Score Card

#	Item	Score	Deadline
1	Requirements Functional Requirements Usecase/ Flow chart/Activity Diagrams Non-functional Requirements Stakeholder buy-in before deadline	25	27 Aug 2023
2	Design (Validation by (TA + Amit + D Raghunath) and Venu) Architecture compliance Meet Functional Requirements + DB Model/ Workflow Sequence Diagrams Meet Non-functional Requirements	25	02 Oct 2023
3	Code (validation by (TA + Amit + D Raghunath) and Venu) Architecture compliance Structure, Code quality, Readability and GITHUB Check-in (working)	40	30 Nov 2023
4	Testing: Unit Testing + Integration Testing	25	30 Nov 2023
5	Deliverables + Documentation GITHUB (on DFADAMIN) Traceability: Requirements to Design to Code to Test to Solution feature Mapping. All deliverables and Handover (to TA, Amit and Dr Raghunath)	20	30 Nov 2023
6	Demo Meeting Functional Requirements Presentation (on DFS Demo machine)	40	1/2/4 Dec 2023
7	Demo Meeting Non-functional Requirements (on DFS Demo machine)	10	1/2/4 Dec 2023
8	Production Deployment on DFS Server	15	2-6 Dec 2023
	Total	200	

Skills Needed (1/2)

Intensive, Practical and Hands-on course → Get ready (real competitive world) !

Fixed Timeline (Fixed Bid – real world)

- Technology Knowhow
  - *nix Systems and Networking (*nix basics: File systems, Networking, Sockets programming, TCP / UDP) [Efficient Data Transfer Projects]
  - Good understanding of Docker / Jupyter Notebooks, GPs, Nvidia / Tensor Flow / PyTorch Setup [Custom Environments and customized Jupyter notebooks]
  - React / NodeJS Web Application and Database (MySQL/ Mongo) Setup / configuration [Web development and data related projects]
  - Python for Batch projects + building AI Models (where needed)
  - Other Technologies : OpenCV, AI/ML, MinIO ….
  - Avoid: Django and other web frameworks

- Software Engineering (Good understanding of all stages pf SDLC)
  - Ability to create Usecases
  - Ability to create Design and Architecture documents (Diagrams: E-R, Sequence diagrams and Architecture)
  - Ability to create Test Strategy, Test Cases and Test Environment
  - Ability to create good User Interfaces, Abstract Services, Reusable Libraries / Components
  - Ability to work with GitHub

Skills Needed (2/2)

- Make it work
  - [Assignment mindset] to [Product / Solution Realization mindset]
  - Hard work pays
  - Make it work (Your application/system works even when you are not there. You Don't have to baby sit, little or no support needed.)
  - Time is Money

- Collaborative Culture (Learn through Collaboration) :
  - Take Ownership + Ability to work with multiple groups to achieve a common Goal
  - Express your ideas clearly and concisely (Oral, Diagrams and Written).
  - Validate ideas / requirements / Show small POC outcomes / Identify challenges early – Work closely with TAs/Supervisors / Domain Experts
  - Better Design and Code quality through Peer Reviews

System Building Approach

- A Hands-on Semester Course ***Data Foundation Systems*** course
  - Spring 2022 (144 students, 40+ projects)
  - Spring 2023 (109 students, 35+ projects)
- A Learner – Contributor Model to engage IIITH students and External interns (through Srishti Summer Research Internship Program) to support Data Foundation Projects has been created.
- Shrishti Summer Internship Program: looking for longer term
- A small core team of Data Foundation Core Developers and IIITH Interns
- GitHub Project Examples: Generic Data Ingestion ML Challenges Platform