**App Capstone Instructions**

**Role Goals**

1. Help students decide which role(s) you want to pursue during the Capstone (you can change your mind)
2. Simulate real-world scenarios
   1. External updates to management
   2. Enhance presentation skills
   3. Develop consultant skills
   4. Get experience in rapid prototyping
   5. Anticipating business needs
   6. Translating business needs into technical requirements
   7. Making tough decisions on tight deadlines
   8. Managing division of labor on shared responsibilities

**Procedure**

1. You will be divided into teams of 2 students
   1. Efforts will be made to balance junior and senior experiences and a diversity of role goals, but everybody is expected to do work on all
   2. Odd numbered cohorts may assign an individual team who has demonstrate enough self-sufficiency to accomplish goals alone
2. Personal capstones are allowed for second and final presentation in addition to team capstone work and subject to instructor approval for fitness in portfolio inclusion for placement
3. After the initial four hours and every four hours of instruction, alternating members of each team will provide an update to the instructor or where every team stands. The Team Scrum Master role internal to the team and document the following:
   1. The status of what was accomplished since the last scrum
   2. Goals for the next scrum
   3. Any issues encountered in the past round of work that need attention
4. Each team will manage their own Issues and ***kanban board in Azure DevOps, Microsoft Planner or Github*** with an expectation that the information will match the report given.
5. Checkpoint presentations of progress towards the final capstone presentation are available upon request at 50% of completion of the allotted project time with 24 hr notice
6. Each team member will be expected to do an individual presentation on their work done that is their individual contribution
   1. All work must be checked into Github from their accounts
   2. The comments must make a reference to the GitHub issue associated with the task
   3. While collaboration is encouraged on each deck , you must be able to answer technical questions on your own in defense of your own understanding of their creation

**Planning for a Minimally Viable Product (MVP)**

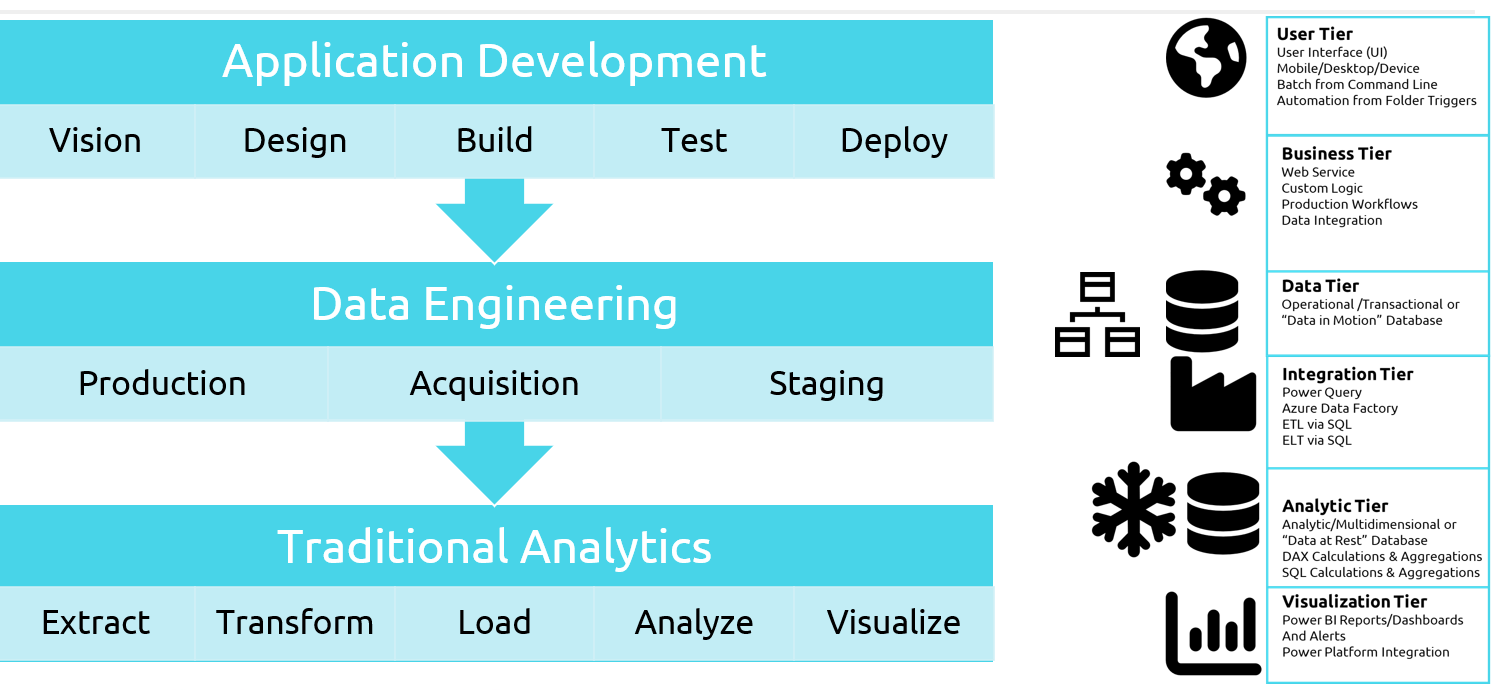
Your instructor will work with each team to determine the scope of the work that fits a combination of the teams’ expected capacity and role skills mastery.

**Business Documentation Goals**

Teams will combine the requirements gathering of being product managers/owners with being the individual contributors. Business problems and high level pain points from several stakeholders will be presented to each team. Individuals will need to own each stakeholders’ outcomes and work together to deliver first a clear cut plan in these documents:

1. Vision
   1. What problem will this app solve?
   2. Usually delivered as a vision statement – written doc
2. Overview
   1. Basic structure of app showing simplest depiction
   2. Level of Detail
      1. Inputs
      2. Actions
      3. Integrations
      4. Phases
      5. Outputs
   3. Delivered usually as a diagram or series of diagram
      1. www.websequencediagrams.com etc.
3. Persona
   1. What are the general groups of tasks in different phases
   2. Usually delivered in PowerPoint
   3. One slide per persona
4. Business Requirements Document (BRD)
   1. Detailed descriptions of all functionality
   2. Sample screenshots of expected look and feel
   3. Design constraints
      1. Platform requirements
      2. Service levels
      3. Legal compliance
      4. API requirements for integration
   4. Written description of all behaviors
      1. Types of interfaces
         1. Command line
         2. Desktop native
         3. Mobile native
         4. Web native
   5. User Stories
      1. Persona doing one task
      2. End to end flow or one component at a time
         1. See other presentation guidance for components

**Technical Documentation Goals**

The depth and breadth of the App Capstone project has significantly more components and deliverables than the Dashboard Capstone. Below is a graphic showing the ideal level of complexity for simulating a real world traditional analytics and application development project Teams will be expected to work together and divide tasks into individual and joint ownership after presenting proposals. Details for expected technical documentation:

1. Technical Requirements Document (TRD)
   1. Software Architecture
      1. Systems Design
      2. Usually delivered in diagrams or bullet point outline
      3. Data Design
      4. See other Presentation Guidance
      5. Detailed Component Design
      6. Combination of diagrams and text as needed
   2. Analytic Architecture
      1. Criteria for Success
      2. Traditional Analytics - How many, $ etc.
   3. Business Requirements
      1. Constraints for Analysis
   4. Technical Requirements
      1. Data gathering from the app architecture
      2. Production deployment

**Core Technical Skills Inventory**

Visualization Tier (SQL & Power BI)

1. Students will be expected to build at least one of each the following analytic assets:
   1. Tables in a data warehouse schema
   2. All types of data modeling layers
      1. SQL based views for verifying DAX calculations
      2. Power query queries loaded into Power BI
   3. At least one type of DAX constructs
      1. DAX Table
      2. DAX Column
      3. DAX Measure
   4. A minimum of 3 visualizations
   5. At least one analytic experience
      1. Drillthrough
      2. Bookmarks
      3. Groups
      4. Hierarchies
      5. Toolitip pages

Integration & Analytic Tier (SQL , Azure Data Factory & Power Platform)

1. Students will be expected to build at least one of each the following analytic assets:
   1. Data populated from an integration solution in two modes
      1. Initial load from an empty set of tables
      2. Incremental load for an arbitrary insertion of new data
   2. All data coming from two sources
      1. Constrained queries on an operational database
      2. Transactional level Flat files directly
   3. Data integration through all methods covered in class
      1. Azure Data Factory pipelines and Data Flows
      2. Power Query and Power Automate steps
      3. Azure SQL stored procedures

Presentation & Business Tier (SQL , Azure Data Factory & Power Platform)

1. Students will be expected to build at least two of each the following analytic assets:
   1. Power App (Portal or Canvas) for providing data entry to user
   2. Power App (Portal or Canvas) for delivering analytic summaries to business users
   3. Power Platform Model-driven App to manage workflow state
   4. Power Automate flows for integrating and automating workflows
   5. Power Virtual Agent for surfacing state of other Power Platform activity

**Presentation Expectations**

Each student will complete a presentation communicating the findings of their capstone using PowerPoint (or something similar) along with live demos of their Power BI dashboard and SQL work. **Cameras will be turned on for the presentation**. Presentations will cover the following outline:

* 15 minutes (1 min pitch, 2 mins for Q&A, the 12 mins in between are up to you)
* Begin with 60-sec elevator pitch (who are you, where are you going and something memorable)
* Problem Description
* Proposed Solution
* Demo
* Pain points, lessons learned

**CAPSTONE Outputs:**

Create a folder in 2022-07-DSI-WDX/Files/06-App Capstone Project/Capstone 1.0 Outputs and label that folder with your first and last name. At the end of your capstone project the following items will be uploaded into your folder:

* Presentation Slide Deck
* PBIX file
* Data Dictionary
* Power Platform assets
* Azure SQL deployment
* Azure Data Factory Pipelines 7 Data Flows

Inside of your GitHub Repo, create a project folder and track your SQL, DAX, ADF, Power Platform ( Apps/Automate/Virtual Agents) and Power Query code