

Wright State University
Department of Computer Science and
Engineering

CEG 7370 – Distributed Computing
Design Challenge Project

Spring 2026

Objective:

The purpose of our open design challenge team project is to utilize foundational principles in Distributed system design, software development, testing and validation to implement your creative ideas with the goal of solving real-world problems (Rapid Prototyping) in a multi/single-member collaborative development environment.

Teams consisting of 2-3 students will propose your project ideas in line with a focused theme then participate in a sequential system development tasks of project conceptualization, development, integration, testing and refinement.

Focused Theme: *Explore innovative ideas for supporting real-time distributed interaction and collaborations* for remote-work settings OR remote-learning OR e-commerce based applications.

To accomplish this you are encouraged to look into existing platforms such as web and mobile-cloud based platforms (example android/iOS). In the applications developed on these platforms students **must** make use of any **ONE** of the following up-and-coming technologies 1) Artificial Intelligence 2) Web-based Augmented Reality.

- 1) **If you choose AI**, you **must** use machine/deep learning-based APIs(Application Programming Interface) for image recognition **and/or** speech recognition/translation **and/or** natural speech processing from service providers like [Google Cloud](#), [IBM Watson](#), [Microsoft Azure](#), or [AWS](#) or you may develop/deploy your own deep-learning model.
- 2) **If you choose Web-based AR**, you may choose web-based AR development platforms such as [playcanvas](#), [AR-kit](#) or [AR-core](#). The nature of these platforms are such that you donot need expensive headsets to develop and demonstrate AR. You may do so easily on your own mobile device (tablet or phone). A typical example for a web-AR based application is [online AR shopping tools from companies such as Amazon](#)

Irrespective of the technology/platform you use in your application there must be elements of ingenuity and creativity in your project. Most importantly your project must **exhibit typical characteristics of a Distributed Computing System.**

A Good place to Start:

1. [Github tutorial](#)
2. [Google Cloud Speech to Text](#)
3. [IBM Watson speech to text](#)
4. [Playcanvas AR](#)
5. [AR-core](#)
6. [ARkit](#)

Tip:

I cannot stress this enough. Distributed Systems/applications are characterized by reliability and connectivity issues. They are incredibly temperamental and will require constant testing. Start your project “AS EARLY AS POSSIBLE”.

Teams:

An important aspect of this project is collaborative-learning. Students are expected to work in teams of 2 or 3. Teams are considered as startups who are working on an innovative idea to solve the problem at hand.

Collaboration:

All teams must use github for version control and collaboration. After completing stage 1 of the project presentation and receiving feedback from the instructor individual team members must have at least **one commit per** week which is code related. The nature of these commits will be evaluated by the instructor/TA to determine the effectiveness of a team-member’s contribution to the project. During project presentations each team-member’s performance will be assessed based on their commits individually.

Acceptable Git commits: Git commits must of a substantial nature. What is substantial nature? Commits of a nature that contribute significantly to an app’s functionality. For example: a subroutine, class or even a collection of smaller commits that has brought a significant difference to the overall functioning of an application.

Unacceptable Git commits: Commits whose effect on the project is insubstantial. For example: a commit that adds/subtracts text in readme file/comments in a code that are absolutely irrelevant to the functionality of the app-under-development.

Note: All teams must add the TA/instructor as a collaborator to their respective GitHub repositories.

Team Management:

All teams must choose a startup/company name for your team. *Afterall it is a startup!* All teams must elect a team-leader who acts as a manager/CEO of your startup.

At earlier stages, we will work together to identify major features your app must demonstrate to succeed in a competitive market. We need to pay particular attention to such aspects as user experience, device constraints, e.g., battery usage, screen size, resolution, as well as product distribution.

Presentation Venue:

All presentations will be in-class.

Stage 1: Idea Phase

Tasks:

1. Understand **Distributed collaboration**,
2. Application ideas ...
3. Teaming: Get your co-founders (keep in mind: *ideas* help get team together.)
4. Get familiar with the development tools. For example, resources available for delivery and development? Identify the SDK. Is the APIs made available? How fast will it work? In short, does it meet your need?

Deliverables: (20 points out of the total 100 points) Due on TBD on Dropbox

1. A 15-minute online-presentation for each team project which also serves as your team project/ proposal)
2. A PowerPoint template for your presentation will be provided to you through Pilot. Sample list:
 - Slides 1: Title (Product Name with a Logo), Team members, Target market, Value propositions

- Slides 2: The pain you are addressing.
 - Slides 3: How do you fix it using AI and distributed computing technology, (may also include mobile/cloud computing)? (It has to fit into the focused theme!)
 - Slides 4: Your conceptualization and justification for your solution.
 - Slides 5: Market research on current offerings available commercially and their limitations.
 - Slides 6: Your target market and value propositions. (Why use yours?)
 - Slides 7: Your business model. How you are going to make money?
 - Slides 8: Team members expertise and contributions.
 - Slides 9: Collaborative tools to accommodate your development and communication efforts: **GitHub (required)**, facebook, google doc, etc.
3. Submit your PowerPoint Slides in PDF through Pilot Dropbox by the deadline.

Stage 2: Design Phase

Tasks:

1. Understand what features and capabilities are required to ace the target use cases.
2. Understand the **existing competitive offers already available in the commercial market**.
What unique features that you offer distinct and make yours a winner.
3. Need to consider, e.g.:
 - UI (e.g., voice command and control, touch, gesture, gaze, ...)
 - Contents (e.g., use case specific)
 - Features: training through virtual role-play, ...
 - Workflows
 - System Architecture

Deliverables: (20 points out of the total 100 points) - Due TBD on Dropbox

- Presentations on **why your design will make a winner for the target use cases**.
- 15-minutes presentation for each team. (up to 6 slides)
 - Slides 1: Title, Team members, Target market, Value propositions (from Phase 1)
 - Slides 2-4: Key Winning features, capabilities, and your justifications
 - Slides 5: System Architecture
 - Slides 6: Conclusion, and Plan for Implementation Phase
- Submit your PowerPoint Slides in PDF through Dropbox by the deadline.

Stage 3: Final Phase:

Tasks:

1. Execute your design
 - implement the prototype based on your ranking of winning features
 - may consider scale back from your design, but have enough to:
 - show feasibility
 - show your team's ability to execute the design (software design, programming)
2. Documentation for final implementation:
 - Submit a single page consisting of project title, team-members' names, link to a recorded video of the project workflow, link to the github repository of the project

Deliverables: (40 points out of the total 100 points)

1. Deliver a 15-minute final pitch deck presentation
2. 15-minutes presentation for each team. (up to 6 slides)
 - This presentation will involve a demo of your prototype.
 - Questions to individual team members related to the project.
 - **Venue: In-person**
3. Submit a link to your GitHub containing:
 - pitch deck PPT in pdf
 - recorded demo video
 - project source code with a Readme file clearly explaining sample inputs and sample outputs
 - **Due date: TBA on Pilot Dropbox**