

## ASSIGNMENT 1 - PART II

### Smart vehicle booking system

---

For this task, each team will be tasked with crafting a high-level design for a software system. The design will be visualised through a single UML diagram, showcasing the comprehensive class structure of the system. This diagram should encompass class names, primary responsibilities, and the connections between classes, including aspects like inheritance, association, aggregation, and potentially details like cardinality and role names.

It is essential for students to come up with a consensus on a single diagram, as this would be helpful in this part of the assignment. The diligence and clarity exercised in creating this high-level design will significantly influence the success of the task.

---

#### Problem Description:

We anticipate the development of a software system for reserving smart bikes intended for on-campus transportation at IIIT-H. The design should support users in booking smart bikes via a mobile app and conducting payment transactions. Users should be able to complete the onboarding process, scan QR codes on bikes located in the parking lot to initiate and conclude trips, and facilitate payments manually or through an auto-deduct feature. While creating this prototype, it is essential to consider various user profiles, ensuring their unique characteristics are clearly reflected in the design, associated documentation, and during the presentation.

#### Part - II:

**This is a team based assignment** that builds upon the individual work you have recently completed. **Each team** has to create a single UML class diagram for the previous task with a mutual consensus.

Your task is to submit the structural code and a corresponding UML diagram of the foundational structure of your assignment, ensuring that the code is logically organised, well-commented, and reflective of a comprehensive understanding of the assignment (smart vehicle system) requirements.

For this stage, we do not require the code to be “executed”. Instead, focus on providing a clear and concise representation of the project's structure, outlining attributes, functions, classes, and their relationships. This approach allows us to assess the architectural integrity and planning of your code.

#### CODE:

- You should be able to capture all the classes and their relationships with relevant attributes and methods. Basically transforming your UML diagram to code. **You are only required to create the skeleton code of the system, NOT THE ENTIRE SYSTEM.**

- **We don't require your code to execute and give us an output.** But we require the code to have structural integrity and should “execute” in the sense that all the classes and included relationships are working smoothly.

**NOTE:**

Make a zip file per team with the following:

1. A single UML class diagram for the team
2. Folder containing the code for the entire system
  - a. Comments must be used, be it assumption(s) or explanations
3. **No deadline extensions will be entertained for the same**

**Submission Guidelines:**

- Submission would be done by **a single member of the team**. No need for all members to submit.
- The deadline for the submission of Part-II is **Monday, January 22th - 11:59 pm**.
- The naming format should be **Assignment1\_Part2\_<Team No>.zip**
- Please follow the format strictly otherwise a penalty may be incurred.