CptS 322 Software Engineering Principles I
Spring 2021
Project Milestone 3
Project Umbrella

(Due March 18 th , 2021 on Blackboard)
Milestone Description:
Further clarify customer requirements by identifying the class-based elements of the requirements model. This milestone focuses on data modeling after understanding and describing high-level use scenarios in the previous milestone.
Your team will deliver a PDF including all class diagrams, package diagrams, and CRC index cards with each diagram being complete (having the three elements included) and the entire diagrams being complete (having relationships specified).
For clarifying requirements, you could develop a prototype of the software at this point.
NOTE: There is a big, but perhaps subtle, difference between customer requirements and product design. The customer requirements phase is about understanding the customer's problem, their pain. The product design phase is about the solution to the customer's problem.
For when you do the class modeling, think about what again (e.g., which operations and attributes for each class, what kind of relationship between classes, etc.), not how yet (e.g., how each operations will be realized).
What to submit: A single PDF, where you include:

1. class diagrams, package diagrams, and CRC index cards, indicating all relationships between classes,

Classes:

EngineProfile:

- +name
- -IsMember
- +getName()
- +setName()
- +IsAMember()
- -saveprofile()
- -viewRecords

TestData:

- +LoadProfile()
- +TestHydroplane()
- +Displaytime()
- +SaveRace()

Menu:

- +showMenu()
- +Displaytime()
- +SaveRace()

Retrieve components:

- +RetrieveUser()
- +LoadRace()
- +CompareUsers()
- +Display()

Mychron5:

- +mychronId
- +name
- Startandstop()
- -oneClockRead()
- -continuousRead()
- -arduinoDataRead()
- -serialRead()
- -myChronDLLSetUp() “is there a dll for MyChron”

Clock:

- +clock
- +name
- -setFrequency()
- -resetClock()

Arduino:

- +clock
- +enableSerialCommunication()
- +enableRS232Communication()
- +enableFTDICommunication()
- +enableSPICommunication()

univeralSerialBus:

- +clock
- +enableSerialCommunication()
- -checkDrivers()
- -communicateUSB()

engineComponents:

- +readSensor()
- +enableSerialCommunication()
- -checkDrivers()
- -communicateUSB()
- logWearLimits()

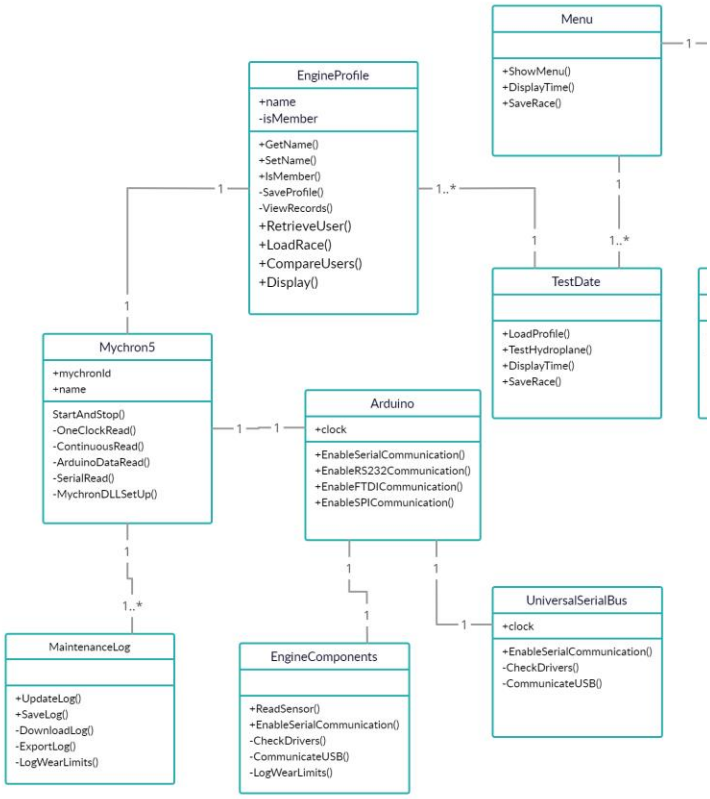
MaintenanceLog:

- +updateLog()
- +saveLog()
- -downloadLog()
- -exportLog()
- logWearLimits()

Admin:

- +createAdminAccout()
- +createGuestAccout()

Class Diagram:



Package Diagrams:

MyChron
+ MyChron5 + EngineProfile - MaintenanceLog - Arduino

Menu
+ Menu + TestData # Admin

CRC Cards

CRC Index Card 1

Menu	
DisplayTime()	Mychron5
saveRace()	EngineProfile

CRC Index Card2

EngineProfile	
name	
viewRecords()	MaintenanceLog
loadRace()	Data
createProfile()	Admin
saveProfile()	
compareProfile()	

CRC Index Card3

TestDate	
testHydroplane()	Mychron5
displayTime()	UniversalSerialBus

CRC Index Card4

UniversalSerialBus	
clock	Arduino
checkDrivers()	EngineComponents

CRC Index Card5

Mychron5	
ArduinoDataRead()	Arduino
MychronDLSetup()	EngineComponents
serialRead()	UniversalSerialBus

CRC Index Card6

EngineComponents()	
readSensor()	Arduino
checkDrivers()	Arduino
logWearLimits()	exportLog()