Kravspecifikation

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Navn | Nikolaj Roager Christinensen | | | |
| Opgave | PDF Downloader | | | |
| Krav/prioriteter: *Her kan du oplyse de vigtigste elementer i dit software* | Funktionelle krav:  Programmet skal effektivt kunne downloade alle de rapporter, der har et virkende link fra en excel fil.  Programmet skal downloade disse PDF-rapporter og dele dem med os via NAS (eller anden måde).  De downloadede rapporter skal navngives efter kolonnen "BRNummer".  Programmet skal udskrive en liste over, hvilke rapporter fra excelfilen GRI\_2017\_2020, der er blevet downloadet, og hvilke der ikke er.  readme fil på github skal forklare brug af programmet, og inkludere UML diagrammer  Programmet skal køre hurtigt og effektivt (Skal kører asynkront).  Programmet skal falde tilbage på alternativt link, hvis primær link giver fejl, hvis ikke sekundært virker skal programmet gå videre til næste dokument.  Ikke funktionelle krav  Programmet skal kunne forklare hvorfor en rapport ikke kunne downloades  Programmet skal kunne gemme PDF lokalt.  Der skal inkluderes UML diagrammer i Github, til at forklare hvordan alle klasser heri virker | | | |
| Kodesprog | C# | | | |
| Evt. frameworks | .NET 8.0 | | | |
| Estimeret tid | 5 dage | | | |
| H | | | | |
| Featurelist *Her kan du tilføje hvilket features som skal kodes til dit software* | Programmet tagre Excel fil som input, den vælger og bruger pdf\_URL (med Report Html Address som fallback).  Programmet printer til en excel fil log.xlsx hvilke  Programmet kan køres med kommandolinjeparamtre, jeg forventer at programmet tager følgende parametre: -only N, hvor N er det heltal rapporter der max downloades (per default downloades alt). -input F, hvor F er et filnavn (default GRI\_2017\_2020.xlsx) -log F, hvor F er filnavn (default log.xlsx) -noout, programmet gemmer ikke pdf filerne, den rapportere kun hvilke der virkede  -output M, hvor M er en folder (hvis den ikke findes laves den) hvori alle gyldige PDF gemmes (default ”Downloads” i den lokale folder). -force, programmet overskriver pdf-er som allerede eksisterer i output folderen (normalt springer den over at downloade dem) -NAS S, hvor S er NAS server navn (Jeg ved endnu ikke hvad en NAS server er, men jeg antager at det har et navn)  log indeholder følgende søjler: filnavn , adresse brugt, Downloadet?, Download startet tidspunkt, download stoppet tidspunkt   Downloaded vil enten skrive: success, skipped (filen eksisterede allerede), Ikke downloaded (errorcode) | | | |
| Work breakdown *Her kan du bryde dit software ned i arbejdsopgaver og estimere tiden det vil tage at bygge hvert et element* | Arbejdsopgave | Start dato | Forventet slutdato | Tidsforbrug |
| Basic class structure for reading and writing   * ExcelReader * ExcelWriter   PDFDocument.  A class for loading, keeping and saving PDF document from a URL  This stores the main and fallback address, an Enum with the Status (Downloaded, Failed or Skipped), and the bytes of the pdf (or alternatively the reason it wasn’t downloaded).  This has functions for downloading document, metadata, and functions for downloading and saving metadata and pdf  The Download function is an Asyncronous task  Main  An async task, which prepares a list of PDFDocuments based on input (once ExcelReader is working, otherwise just use default).  Awaits all document loading finishing  Tells all documents to write to pdf and log (once ExcelWriter is working)  Input reading  Create a class and/or function for reading commandline argument  At the start of Main, use this class to update what PDFDocuments we want  NAS server  Read up on what on earth this even is, how to work with it in C#   … Then do it, given my lack of knowledge I don’t know what that even entails | Tirsdag  Mandag  Mandag  Tirsdag  Tirsdag | Tirsdag  tirsdag  Onsdag  Onsdag  Fredag | Et par timer, ikke ekstremt vigtigt, for at debugge download kan vi starte med en hardcoded liste af url  2 dage  Udvikles og forbedres sideløbende med Main funktionen, og udvides gradvist.  Til at starte med downloader den altid, skriver status, og kan gemme.  3 dage, rimelig simpel, men tager langt tid fordi det udvikles Sideløbende med PDFDocument, og Excell reader/writer klasse, input reading, og NAS  Få timer  Rimeligt simpel, jeg har allerede kode jeg kan kopiere fra andre projekter  Først når NAS virker kan jeg tilføje arumenter med dette  Det er ikke klart hvor lang tid dette vil tage, da jeg ikke ved hvad det er |