# Software Test Rapport – Unit & Integration Testing

## 1. Generel Information

**Projekt / Systemnavn:** PDF-downloader  
**Testansvarlig: Viktor From**  
**Versionsnummer / Build ID: 1**

## 2. Testformål

**Formål med testen:**

Formålet med denne test er at verificere at PDF Downloaderen korrekt kan håndtere download af PDF’er.

## 3. Testresumé

**Overordnet opsummering af testforløbet:**

| **Testtype** | **Antal tests** | **Bestået** | **Fejlet** | **Kommentar** |
| --- | --- | --- | --- | --- |
| Unit tests | 8 | 5 | 3 |  |
| Integration tests | 4 | 3 | 1 | Kan ikke håndtere navnefejl |
| **Samlet resultat** |  |  |  |  |

## 4. Testet Funktionalitet

**Beskrivelse af hvilke dele af koden der blev testet:**

| **Klasse / Modul** | **Metoder / Funktioner** | **Formål med testen** | **Resultat** |
| --- | --- | --- | --- |
| **Prepare.py** | Prepare\_folders\_and\_find\_pdf\_duplicates()  load\_and\_filter\_excel\_data | Formålet er at undersøge hvordan de to funktioner håndterer fejl | Begge funktioner kører uden fejl |
| Download\_files.py | Delete\_downloaded\_files  Download\_pdf  Process\_downloads\_threaded  Summarize\_downloads | Formålet var at undersøge hvilke fejl de forskellige funktioner kan give, når de køres hver for sig. | Alle funktioner kører som forventet. |
| Report\_writer.py | Write\_to\_report  Clean\_report\_file | Formålet var at se om de to funktioner fungerer hver for sig | De kører begge to fint. |

## 5. Testtilgang og Miljø

**Tilgang til test:**  
Testene blev udført manuelt ved brug af unittest. Komponenterne blev isoleret fra hinanden ved brug af mocking.

**Testmiljø:**

| **Komponent** | **Version / Setup** |
| --- | --- |
| Operativsystem | Windows 11 |
| IDE / Framework | Visual Studio Code |
| Database | Ingen |
| API’er / eksterne services |  |
| Build-værktøj |  |

## 6. Testresultater

**Detaljeret resultatoversigt:**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Test ID** | **Testnavn / Enhed** | **Input** | **Forventet Output** | **Faktisk Output** | **Resultat** | **Kommentar** |
| UT-001 | prepare\_folders\_and\_find\_pdf\_duplicates | Ingenting | Liste over eksisterende filer | Liste over eksisterende filer | ✅ Bestået |  |
| UT-002 | AddUser() | tomt navn | Fejlmeddelelse | Fejlmeddelelse | ✅ Bestået |  |
| IT-003 | OrderService + PaymentGateway | ordredata | Ordre betalt | Timeout | ❌ Fejlet | Fejl ved API kald |

## 7. Fejl og Observationer

**Beskrivelse af fundne fejl eller uoverensstemmelser:**  
(Angiv kort beskrivelse, sværhedsgrad og evt. forslag til løsning.)

| **Fejl-ID** | **Beskrivelse** | **Sværhedsgrad** | **Forslag til løsning** | **Status** |
| --- | --- | --- | --- | --- |
| BUG-01 | Download\_task fejler ved skrivning til rapport | Lav | Inkludér report\_writer objekt i klassen | Åben |
| BUG-02 | Download\_task kan ikke håndtere tomme datatyper | Middel | Tjek for om datatypen er tom, før den bruges | Åben |
| BUG-03 | Summarize\_downloads() kan ikke håndtere at summary bliver returneret uden data | Lav | Tjek om summary er tom inden der returneres | Åben |

## 8. Opsummering og Konklusion

**Overordnet vurdering af testforløbet:**  
Overordnet er koden godt struktureret. Alting er veldokumenteret og kodens funktion er let gennemskuelig.

Det er også værd at bemærke at integrationstestene køres fint igennem.

## 9. Refleksion og Videre Arbejde

**Refleksion:**  
Generelt fungerede testene meget godt. Den primære arbejdsbyrde lå i at finde ud af hvordan unit tests fungerede og hvilke tests der skulle køres

**Plan for videre arbejde:**

* Udvide testdækning for edge cases
* Udvide test af fejlscenarier.
* Automatisere testkørsel i CI/CD pipeline