# Jupyter Notebook Week 6 Prototype - Software Requirement Specification (SRS)

## 1. Introduction

### Purpose

The purpose of this prototype is to ensure the code written thus far by team members sufficiently completes the two key functionalities of converting both Jupyter Notebook files and Word document files into PDF format. The successful implementation will ensure the team is on the right track in terms of both their ideas and their coding work.

### Intended Audience

* Students
* Teachers/Tutors/Lectures

### Intended Use

The intended use of this prototype is to be a more-basic first draft of the logic the team wishes to implement in the final product of the project scope. It will test the conversion to PDF for both Jupyter Notebook and Word document files. The successful implementation of this prototype will give the team the knowledge that the code they’ve written works on a smaller scale than OnTrack.

### Scope

The scope of the prototype is to create a simple front-end interface similar to that of OnTrack and, through the use of command-line commands, allow for the following functionalities:

* Converting a Jupyter Notebook file into PDF format
* Converting a Word file into PDF format

### Definitions and Acronyms

Jupyter Notebook – a web application that allows users write and run live code, equations, visualizations, and plain text in various languages.

.ipynb – Jupyter Notebook file type

.doc – Microsoft Word file

.docx – Microsoft Word file

PDF – Portable Document Format

Library – A library is a collection of pre-written code that provide further access to system functionality such as file I/O that would otherwise be inaccessible. This is done importing the library at the beginning of the program.

HTML – HyperText Markup Language

RubyOnRails – a server-side web application framework written in Ruby.

Nbconvert – is a library of pre-written code used to convert Jupyter Notebook file to PDF.

Backend – Is development that happens behind the scenes, it is all the parts of a computer system or application that is not directly accessed by the user, it is responsible for storing and manipulating data through code.

Frontend – Is development on what the user can see and/or directly interact with (i.e., what can be seen on the computer screen, such as a window, or buttons and input fields/boxes)

## 2. Overall Description

### 2.1 User Needs

As a student, I want to be able to upload Jupyter Notebook (.ipynb) files without having to go through the extra step of converting them to a PDF first.

As a tutor, I want students to be able to upload any file they are working on so they can focus on the quality of the work.

### 2.2 Assumptions and Dependencies

Assumptions include:

* The user has a working and valid Jupyter Notebook/Word Document ready to be converted to PDF.
* The user wants the input file to be converted as uploaded.
* The user has access to OnTrack.
* Key project member’s availability
* Key project member’s performance
* Key project member’s skills

Dependencies include:

* Input of a valid Jupyter Notebook/Word Document file.
* A valid internet connection, to interact with OnTrack environment
* A Docker container must be created first before testing of Jupyter Notebook/Word Document conversion can be tested.
* Approval of project expansion must be given before work on expansion begins

## 3. System Features and Requirements

### 3.1 Functional Requirements

* Convert Jupyter Notebook files to PDF format
* Convert a Word Document to PDF format
* Compatibility with existing code

### 3.2 External Interface Requirements

* nbconvert library
* Apache library
* Forked repositories
* User interface (interaction logic between software and user)

### 3.3 System Features

* Basic front-end user interface for user to trigger logic
* Ability to convert Jupyter Notebook file to PDF file
* Ability to convert Word document file to PDF file

### 3.4 Non-functional Requirements

* Usability – User standards
* Scalability – ability of software to be scaled to encompass project scope in its entirety
* Maintainability – ability of software to be maintained, ensuring consistent and upmost performance
* Reliability – The ability of the system to consistently perform its required functions under stated conditions.
* Documentation – User documentation, testing results, meeting minutes and notes, contribution notes, discussions.