|  |
| --- |
| **Disclaimer**    This is a template for the User Interface Specification that students may use in SOEN341, SOEN390 and SOEN490.      **Note to authors**    If you add any new sections to the document please make sure that you maintain the header and text styles.    Before submission of the first draft of this document please make sure to update the Table of Contents and to delete this page.      **Author**: Dr. C. Constantinides <cc@cse.concordia.ca> |

**User Interface Specification**

Version 1.0

for

Dr. C. Constantinides

NBDiff

Prepared by

|  |  |  |
| --- | --- | --- |
| <name> | <student-id> | <email> |
|  |  |  |

|  |  |
| --- | --- |
| Instructor: | Dr. C. Constantinides |
| Course: | Software Engineering Capstone Project |
|  | 12/01/2014 |

**Document history**

|  |  |  |  |
| --- | --- | --- | --- |
| **Date** | **Version** | **Description** | **Author** |
| 12/01/2014 | 1.0 | User Interface Specification | NBDiff Team |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

Table of Contents

[1. Introduction 5](#_Toc377319694)

[2. User Centered Design 5](#_Toc377319695)

[2.1 User characteristics 5](#_Toc377319696)

[2.2 Set of tasks performed 6](#_Toc377319697)

[2.3 Context of use 7](#_Toc377319698)

[2.4 Stakeholder objectives 7](#_Toc377319699)

[3. Overall User Interface Architecture 8](#_Toc377319700)

[3.1 Pages/ Screens 8](#_Toc377319701)

[3.2 Options 8](#_Toc377319702)

[3.3 Structure of the UI Rationale 9](#_Toc377319703)

[4. Navigation 9](#_Toc377319704)

[4.1 Visibility 9](#_Toc377319705)

[4.2 Navigation 9](#_Toc377319706)

[4.3 Navigation Rationale 9](#_Toc377319707)

[5. Feedback 9](#_Toc377319708)

[5.1 Application 9](#_Toc377319709)

[5.2 Task Completion 10](#_Toc377319710)

[5.3 Error Recovery 10](#_Toc377319711)

[6. Screen Layout 10](#_Toc377319712)

[6.1 Space Management 10](#_Toc377319713)

[6.2 Items Location 11](#_Toc377319714)

[7. Business Rules 11](#_Toc377319715)

[8. Future Enhancements 11](#_Toc377319716)

List of figures

No table of figures entries found.

# 1. Introduction

This document provide a detail description of the user interface of the NBDiff system. The user interface is designed based on the visibility, feedback, and affordance in order to accomplish a higher accessibility and achieve the required tasks in an easier and faster way. It also describes the general background of the users going to use this system. Which are scientists who use the IPython Notebook in order to produce reproducible research. Also, this document includes user interface architecture, navigation, feedback, screen layout.

# 2. User Centered Design

## 2.1 User characteristics

The target audience of our stakeholder’s education program can be found here:<http://software-carpentry.org/audience.html>

We hope to cater NBDIff to users of the following types (most important in bold): **academic researcher (beginner programmer)**, **academic researcher (expert programmer)**, peer reviewer, study reproducer, undergraduate student, data analyst, blogger.

The most important user types are documented in detail below:

**Representative:** Scientist / Beginner Programmer (User-SBP)

**Description:** This user is typically a graduate-level student in a university who performs research in the sciences, engineering, or medicine. While their research relies on performing computations on data they have collected through experiments, or writing simulations of theoretical phenomena, they have received no formal training in programming. This user is the most important user to our primary stakeholder, as our stakeholder trains these scientists to become better programmers.

**Level:** Beginner

**Responsibilities:** Analyze data by creating models, calculating statistical results, or creating simulations. Producing plots and tables of data for exploratory purposes or for inclusion in published research. Collaborating on IPython Notebooks with other researchers.

**Success criteria:** Accepting changes from other researchers to their Notebooks and successfully producing correct, merged Notebooks.

**Representative:** Data Analyst (User-DA)

**Description:** This user is typically a well-educated statistician or computer scientist employed by a for-profit company interested in analyzing a data set for strategic gain. They are well-versed in data analysis toolkits like the R programming language, Weka, or Pandas. They know how to quickly get a sense of the data they are working with, and how to extract meaningful results from that data. They may share preliminary results of their analysis with other members of their company in the form of an IPython Notebook. They are well-versed in version control tools.

**Level:** Expert

**Responsibilities:** Creating research results and proofs-of-concept of data analysis algorithms for analysing large data sets. Tracking changes to their work and results as they explore data sets.

**Success criteria:** Accepting changes from other analysts to their notebooks. Determining what has changed between different versions of the Notebook.

## Set of tasks performed

1. Merge task

* For the beginner scientist programmer: when they create some model and calculate their statistical result or create simulations, they use the data result by producing plots and tables for exploratory purposes or for inclusion in published research. This system allows them to accept changes from other researchers to their notebooks and successfully producing correct merged Notebooks.
* For the expert scientist programmer: when they want to produce reproducible research in the form of IPython Notebooks that other researchers can run, they will publish the Notebooks alongside a paper, or in some cases submit the Notebooks as the papers itself. In addition to authoring the Notebooks themselves, they also collaborate with other researchers on them. They need to be able to effectively perform code reviews on the changes before they merged into the master Notebook. This system allows accepting changes from other researchers to their Notebooks and successfully producing correct, merged Notebooks. Reviewing changes to Notebooks. Finding bugs or code smells in new/modified code. Understanding the changes between Notebooks to determine when scientific results change for providence.

1. Diff task

* For the expert scientist programmer: when they want to create research results and proofs of concept of data analysis algorithms for analysis large data sets. and when they want to track changes to their work and results as they explore data sets. This system allows accepting changes from other analysts to their Notebooks, and determine what has changed between different versions of the Notebooks.
* For the beginner scientist programmer: when they create some model and calculate their statistical result or create simulations, they use the data result by producing plots and tables for exploratory purposes or for inclusion in published research. This system allows accepting changes from other analysts to their Notebooks, and determine what has changed between different versions of the Notebooks.

## 2.3 Context of use

The target users of NBDiff are scientists who are, or will be, using the IPython Notebook to perform reproducible computational research in science, technology, engineering and medicine (STEM). Our target users are scientists with a wide range of computational ability, including beginner and expert programmers. The NBDiff project seeks to make version control of Notebooks easier; including collaboration with multiple scientists (for instance, in a lab).

Most of our users will be graduate students, but some will be faculty members of universities, and others will be researchers working in publicly- or privately-funded labs. In most cases, the users will have at least a bachelor’s degree in a technical field, though some universities are using Notebooks as part of their undergraduate curriculum.

## 2.4 Stakeholder objectives

Existing solutions lack the ability to display diffs and merge Python notebook properly. Since the format of the notebooks is JSON, existing merge tools are inadequate in dealing with this format as they will show massive amount of incomprehensive text. Not all scientists would be able to read JSON and would prefer a similar user interface present in IPython notebook.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Need | Priority | Concerns | Current Solution | Proposed Solution |
| Diffing | High | Algorithm | Manual Diffing | Automatic Diffing |
| Merging | High | Algorithm | Manual merging | Automatic merging |

# 3. Overall User Interface Architecture

## 3.1 Pages/ Screens

The system contains two main pages which are the diffing page and the merge page. The diffing page displays the two IPython Notebooks that are being diffed on both sides of the page. Each cell of left Notebook is compared with the element of the right Notebook. As a result of this process the two Notebook files will include some colors to notify the user of all differences between these two Notebook’s cells such as having red color when something has been deleted, green color when something has been added, orange, when something has been modified.

The merging page shows three columns where the notebooks are on either sides of the page and the middle column shows which version of the elements in the notebooks will be added to the final notebook. The Notebooks still have the elements in the cells highlighted in different colors just as in the diffing page.

## 3.2 Options

In the diff page, a check mark can be enabled or disabled in between each elements being compared in the notebooks. When a check mark is enabled, the element next to it on the diffing page will be staged.

As for the merge page, in order to pick which version will be part of the middle column, arrows are placed in between the columns, which makes it easier for the user to pick which element he wishes to keep in the merged notebook.

## 3.3 Structure of the UI Rationale

The approach that was chosen to make diffing and merging easier is to compare each cell in the different notebooks and place them side by side in different columns. This approach enables the users to notice the differences between the cells faster and therefore make a choice between which cell he/she wishes to keep easily. The structure of the UI is designed in a way that helps the user be more efficient in completing his or her task.

# 4. Navigation

## 4.1 Visibility

In our system the User interface is simple enough to be visible. The user first will easily open one of the two pages and upload two Notebook files in the appropriate place for them and start either the diffing process or the merge process. The visibility of the result in both processes is clear enough by having the coloring system described before that makes it easier for the user understand what happened in each cell compare to the one horizontally parallel to it.

## 4.2 Navigation

The navigation between the two main pages (diff page and the merge page) is not implemented and designed yet. At the time this system will be done the navigation between different screens will be designed by having a button in each page. In the diff page, the button will navigate to the merge page, and in the merge page, the button will navigate to the diff page.

## 4.3 Navigation Rationale

N/A

# 

# 5. Feedback

## 5.1 Application

No user has yet given feedback for the final application’s user experience. Refer to document usability testing, we did two usability testing for our User Interface and we got some feedback that took into account.

## 5.2 Task Completion

* Diff Tool

After the user select both Notebook files and start the diffing process, he/she will notified that the task completed by seeing the colors added to both files. Red color to the removed cells or subcells, green to the added cells or subcells, and orange to the modifiable cells.

* Merge Tool

After the user select both files and start the merging process, he/she will be notified of the completed task by seeing the middle bar filled in and being able to press on the middle arrow buttons to select left or right cells. In addition to being able to save the merge result file that appear in the middle par. In addition to have colors in the original Notebook files placed in the left and right bars. Red color to the removed cells or subcells, green to the added cells or subcells, and orange to the modifiable cells.

## 5.3 Error Recovery

If an error ever occurs during a diffing or a merging process, the error will be displayed to the user along with how to resolve the error. Error messages will be displayed on top of the current page and the user will have to click on a button to remove the message.

# 6. Screen Layout

## 6.1 Space Management

In our system we don’t have much empty spaces in both the merge and the diff pages. At the top we made use of the empty spacing by adding the NBDiff logo at the top as a header that moves with the page up and down. Also, the question mark button is place in the empty space at the top right of the page which moves in the same way as the logo moves. In addition to that, we added a save button in the merge page to make it possible for the user to save the result of merging the two Notebook files and it moves in the same way the logo and the question mark move.

## 6.2 Items Location

* Bars

In diff tool we have two vertical bars that divide the page into two equivalent columns each will include a Notebook file. In merge tool we have three vertical bars. The left and the right ones will include the two Notebook files. The middle one will include the merged result of the two existing Notebook files.

* Arrow buttons

In order to have an organized, visible merge page, we designed the user interface to have arrow buttons placed between the left bar and the middle bar in addition to place them between the right bar and the middle bar. These arrow buttons located there to allow the user to choose the required cell and placed it in the middle bar (merged result bar).

* NBDiff Logo

The system logo is placed on the top left of the page. Also, it is visible all the time even if the user move up or down.

* Question mark button

This button is used as a help menu for the user. Whenever he/she clicks on it he will have some help notes that allows him/her accomplish the current task. We place it on the top to make it as visible as possible all the times to the user if he/she needed any help.

* Save button in merge

This button is accessible only after generating a merged file in the merge tool. This button is placed at the top right of the page close to help button (question mark button). It is located there to have a higher visibility to the user to see it.

# 7. Business Rules

N/A

# 8. Future Enhancements

N/A