

Class Scribe: A Modern Approach to Notetaking

A Technical Report submitted to the Department of Computer Science

Presented to the Faculty of the School of Engineering and Applied Science
University of Virginia • Charlottesville, Virginia

In Partial Fulfillment of the Requirements for the Degree
Bachelor of Science, School of Engineering

John Watkins
Spring, 2020.

Technical Project Team Members

Benjamin Brown
Henry Weber
John Watkins
Rahat Maini

On my honor as a University Student, I have neither given nor received
unauthorized aid on this assignment as defined by the Honor Guidelines for
Thesis-Related Assignments

Signature John Watkins Date 4/12/2020

John Watkins

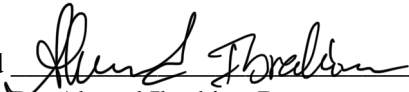
Approved  Date 4/12/20
Dr. Ahmed Ibrahim, Department of Computer Science

Table of Content

Abstract	3
List of Tables (if applicable)	4
List of Figures (if applicable)	5
1. Introduction	5
1.1 Problem Statement	6
1.2 Contributions	6
2. Related Work	7
3. System Design	8
3.1 System Requirements	8
3.2 Wireframes	9
3.3 Sample Code	9
3.4 Sample Tests	9
3.5 Code Coverage	9
3.6 Installation Instructions	9
4. Results	11
5. Conclusions	12
6. Future Work	13
7. References	14

Abstract

A good abstract explains in one line why the paper is important. It then goes on to give a summary of your major results, preferably couched in numbers with error limits. The final sentences explain the major implications of your work. A good abstract is concise, readable, and quantitative.

Length should be ~ 1-2 paragraphs, approx. 400 words.

Abstracts generally do not have citations.

Information in the title should not be repeated.

Be explicit.

Use numbers where appropriate.

Answers to these questions should be found in the abstract:

What did you do?

Why did you do it? What question were you trying to answer?

How did you do it? State methods.

What did you learn? State major results.

Why does it matter? Point out at least one significant implication.

List of Tables (if applicable)

This should be on a separate page but is not required if you have no tables.

List of Figures (if applicable)

This should be on a separate page but is not required if you have no figures.

1. Introduction

This part should be 2-3 simple, easy to read paragraphs. Be sure to include a hook at the beginning of the introduction. This is a statement of something sufficiently interesting to

motivate your reader to read the rest of the thesis, it is an important/interesting scientific problem that your thesis either solves or addresses. You should draw the reader in and make them want to read the rest of the thesis. Think about what made you choose this project to work on!

1.1 Problem Statement

Start with a description of what the organization you are working offers.

- What work they do?
- What problem(s) they faced?
- How do they currently solve that problem as of the beginning of this academic year, before you work on your solution?
- How long does it take?
- What problems does the current way of solving it cause?

Then, include a paragraph or two on how and why your system will be beneficial.

1.2 Contributions

What did you accomplish to contribute to the solution of the problem statement? Which parts were you able to solve or make better. In the subsection, you can mention things like: “we were able to create a web based application which allows the organization’s management team view how many clients used their product”. The rest of this thesis is organized in four sections. In Section 2, related work is presented. Section 3 shows our approach to address the aforementioned problem stated, our web-based application, as well as the system design. The results of our work is discussed in Section IV. Finally, Section VI concludes this thesis.

2. Related Work

The goal in this section is to give a context of what else is out there.

- * What other systems exist that do similar things?
- * Are they custom written, or generic?
- * Why don't they fit the bill?

- Show where your system fits into everything.

3. System Design

First, start by describing the high level goals of your system. A few sentences which quickly describe what each type of users should be able to do. Also, mention what language/framework will you use to develop the system. Why did you choose such language/framework? What license is your code under?

3.1 System Requirements

Using your own words, state the importance of gathering system requirements. Then, include a list of your “minimum requirements”, “desired requirements”, and “optional requirements”. You can state the due date for each type of requirements to be completed/developed.

3.2 Wireframes

Using your own words, state the importance of wireframes. Then, include the wireframes you developed for the system. If you have many wireframes, make sure to use as little space as you can.

3.3 Sample Code

Include a sample of the code you developed. Make sure you include code that is easy to understand and has documentation. This can be three model functions, three views, and three form classes. If you don't have a clear vision on what you should include in this subsection, consult the course instructor (and have your code ready).

3.4 Sample Tests

Using your own words, state the importance of testing. Then, include sample tests with description of what they test.

3.5 Code Coverage

State what package you are using for code coverage. Include directions on how you set it up. Then, include the statistics which the tool gives on your code coverage.

3.6 Installation Instructions

This subsection should contain instructions and detailed steps of how to deploy the code/system on the customer's hosting choice. Assume that the customer has got a new fresh OS install, how would they get the system up and running? Audience is the customer, so do not assume that they have any technical knowledge beyond what you already know they have.

Your installation instructions must include the following:

How to create the account(s) that will host the system?

- a walk-through of the web pages to configure this

How to install any dependent packages?

- apt-get installs, PHP libraries, Ruby gems, or Python libraries

How to upload all the files from the system?

- Have them download a free version of SecureCRT or similar

How to configure those files?

- How to update database.yml with the MySQL username, password, and DB

How to initialize the database?

- Include a schema.sql (or similar) that they can upload to create the tables

How to load any default values into the database?

4. Results

- Did the system solve some or all the problems?
- What are the results of that?
- How does the customer use the system?
- How do other stakeholders use the system?
- This will require ****ACTUAL**** numbers (ideally measured)
- Example: "The scheduler allowed the customer to prepare a schedule in 30 minutes, where it took 3 hours before"

5. Conclusions

What is the strongest and most important statement that you can make from your observations?

If you met the reader at a meeting six months from now, what do you want them to remember about your thesis?

Refer back to the problem posed, and describe the conclusions that you reached from carrying out this work, summarize new observations, new interpretations, and new insights that have resulted from the present work.

Include the broader implications of your system.

Do not repeat word for word the abstract, introduction or discussion.

6. Future Work

Include when appropriate (most of the time)

Remedial action to solve the problem.

Further research to fill in gaps in our understanding.

Directions for future investigations on this or related topics.

7. References

Cite all ideas, concepts, text, data that are not your own

All references cited in the text must be listed

Use APA Citation. A guide can be found here: <https://guides.lib.virginia.edu/stsapa/citex>

Do not use footnotes