EZ Grader - iOS App for Grading Structured Content

Rolandas Burbulis
Department of Computer Science
Golisano College of Computing and Information Sciences
Rochester Institute of Technology
Rochester, NY 14586
rxb4548@cs.rit.edu

***Abstract*—EZ Grader is an iOS app, allowing instructors to grade PDF-formatted structured student assignments from the convenience of their iPad or iPhone devices. EZ Grader relies heavily on PDFKit, framework released for iOS by Apple in WWDC 2017, in order to display and annotate PDF documents. Along with typical PDF annotation features, such as the ability to annotate free-hand and via keyboard text, EZ Grader provides features that are unique to grading assignments, such as per page and per student viewing modes, the ability to add maximum points component of the grade to all students at once, and a tabular view of each student's performance against each assignment question. EZ Grader aims to improve the overall course experience for both instructors and students alike. Digital grading allows for easy correction of mistakes. Automatic tallying of the grades eliminates the potential of mathematical errors when adding up the total grade for each student. The tabular view of each student's performance against each question allows instructors to quickly identify potentially problematic questions. This insight may lead to re-visiting problematic topics in class, grading particular questions on a curve or modifying those questions for future assignments. Finally, the possibility of student dishonesty is reduced, as students can no longer change their answers after the assignment is graded to recover points lost, because instructors have the original graded assignments.**

***Index Terms*—Mobile applications; Educational technology; Computer aided instruction**

I. Background and Introduction

Computer technology transformed and continues to transform what seems to be every aspect of our lives - every aspect *except* for grading student assignments, it seems. From high schools to universities around the world, the picture is generally the same: teachers and professors, pen or pencil in hand, grade paper-based versions of student assignments. Clearly, there are various problems with this approach. Instructors are unnecessarily burdened with needing to carry around potentially heavy stacks of assignments from office to home and back again in order to do the grading. Typographical errors happen when grading, which the instructor may wish to correct. Also, as the instructor is grading, they may realize that a certain question needs to be voided or curved due to many students making the same mistake - possibly due to the wording of the question. If the instructor is using a pen, their only choice might be to cross out incorrect marks and write in new ones. This is unsightly and potentially confusing. The instructor may also make a mathematical mistake when tallying up the student's total grade, which might not be discovered until the assignment is returned to the student. This can be frustrating to both students and instructors alike. When reviewing the results of the assignment with the class, instructors often have to rely on their memory as to which questions were particularly problematic to students. That memory of student performance on specific questions is even further removed when instructors make decisions to re-use the questions in future assignments. Finally, student dishonesty can occur after graded assignments are returned to students, as students can potentially change their answers in order to attempt to recover some points lost, and the instructor may likely have no means to verify what the student's original answer was.

All of these problems can be addressed by moving grading into the digital age. Of course, digital annotation solutions already exist. On one end of the spectrum, there are generic PDF editors out there - they simply allow users to draw and add text to PDFs, which might be sufficient enough to address many of the concerns of the paper-and-pen approach. One such example is the Files app in iOS [1]. On the other end of the spectrum, there are apps such as Schoology [2] out there, which are full blown learning management systems. Schoology has a basic package, which is free, as well as a paid enterprise package. The user first needs to create a Schoology account. Then they need to create a course, which generates a course access code, which is to be shared with the students. Using the provided course access code, the students then create their own accounts and join the course. At this point they can submit their assignments, which then become available for the instructor to grade. This type of an app requires the participation of the entire class - all of the students and the instructor. It also means that the assignments are submitted by the students in a digital format. One can see that this type of a system should work well for homework assignments, for example. But what about quizzes, tests and exams? Instructors will often want to give a paper-based version of the assignment in these kinds of situations, with no electronics allowed to be used by the students. How should the professor grade these paper-based submissions using Schoology? They could scan-in the paper submissions into a digital format. But then what? Clearly, the professor would not want to create dozens of fake

student accounts just in order to ”submit” the assignments for themselves to grade in Schoology.

This paper does not propose replacing paper-based assignments with digital counterparts, which is a topic that is surely going to come up more and more frequently and is worth exploration in and of itself. The author assumes that paper-based assignments will continue for the foreseeable future. The author also assumes that the instructors will be able to scan-in these assignments in order to convert them into a digital format. Printer copiers are generally available at most schools, which are able to bulk scan-in assignments and turn them into PDF documents, ready for digital grading.

This paper introduces a new app, EZ Grader, which ingests these scanned-in PDFs and allows instructors to grade them digitally. This paper begins by briefly introducing the software development environment as it pertains to developing iOS apps. Next, the features of EZ Grader app are described and showcased. The architecture of the app, including the usage of Apple's PDFKit framework is presented next. The paper then discusses the benefits of digital grading and how the overall class experience can be improved as a result. Major challenges when it comes to the development of EZ Grader are discussed next. Finally, the paper concludes with the author's experience with the EZ Grader app and potential topics for future work.

II. iOS App Development Environment

Xcode is Apple's integrated development environment for app development for various platforms, including iOS. Since Xcode only runs on macOS, a Mac machine is also a prerequisite for iOS app development. EZ Grader is written in Swift and its source code is publicly available on GitHub.

III. Requirements and Assumptions

EZ Grader application requirements and assumptions are as follows:

- User will be able to select PDF documents to be graded from their iCloud Drive.
- User will be able to view assignments either per page or per student.
- User will be able to free-hand annotate anywhere on the page.
- User will be able to add keyboard text annotations anywhere on the page.
- User will be able to add grades anywhere on the page. Furthermore, maximum points component of the grade will be added to the same location on the corresponding pages of the assignments of all other students.
- User will be able to erase free-hand annotations.
- User will be able to edit or remove keyboard text annotations and grades.
- User will be able to save annotated assignments, as well as view all entered grades in a tabular fashion.
- It is assumed that all assignments to be graded will have the same number of pages. Error message will be displayed if this is not true.

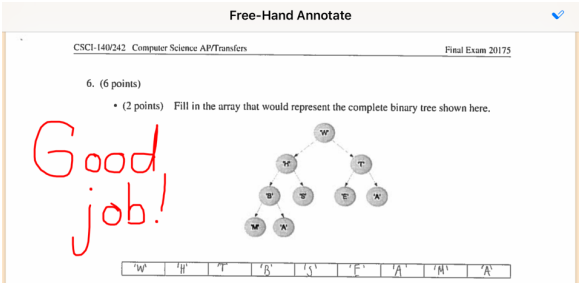

Fig. 1. User in free-hand annotation mode

IV. EZ Grader

EZ Grader is an iOS app, providing the features instructors need to grade structured students assignments on their iOS devices. Structured content assignments are assignments where each student's assignment consists of the same number of pages, and the questions, question order, question positioning, and space to write or select the answers is identical in the corresponding pages of all students. Most exams and quizzes are structured, for example. Homework assignments which students write on their own paper would not be considered structured for the purposes of EZ Grader, as different students can fit a different number of questions on a page.

While the app runs on an iPhone, iPad is the device of choice for EZ Grader, as the larger screen size of the iPad allows for a more comfortable grading experience.

EZ Grader expects assignments which are to be graded to be in the PDF format and for them to be located in the user's iCloud Drive.

A. Grading Modes

One of the features of EZ Grader which differentiates it from generic PDF editors is its grading mode feature. EZ Grader provides the following grading modes:

- 'Per Page' mode - this mode replicates the preferred way that instructors like to grade in when grading paper-based assignments, and it is the default mode in EZ Grader. Page one of all students is presented first, followed by page two, and so on.
- 'Per Student' mode - in this mode, the entire assignment of the first student is presented first, followed by the assignment of the second student, and so on. This mode will often become useful when grading in the 'Per Page' mode, but when the question or the answer spans multiple pages.

User's location in the document is preserved when switching between the two grading modes.

B. Features

EZ Grader provides the standard annotating features that one would expect to find in a generic PDF editor. User can draw on the page using their finger or a stylus, as seen in Figure 1. User can also insert keyboard text annotations by tapping the location within a page where the annotation

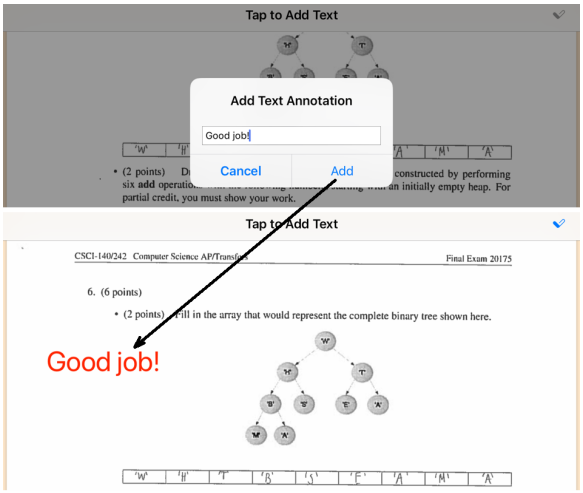

Fig. 2. User adding keyboard text annotation

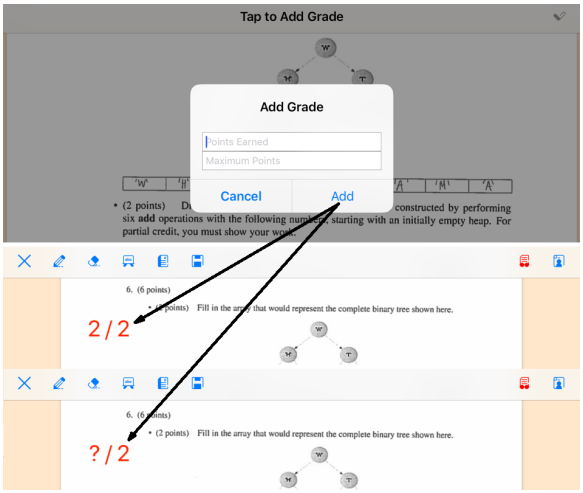

Fig. 3. User adding a grade

should be inserted in, as seen in Figure 2. User can also erase free-hand annotations and edit or remove keyboard text annotations.

Another feature which distinguishes EZ Grader from generic PDF editors is its concept of a grade. Grade consists of two components: points earned and maximum points. As seen in Figure 3, when a user adds a grade to a specific location of a page, the same grade, with its points earned component replaced with a question mark, is added to the same location of the corresponding pages within the assignments of all other students. This is the reason why the assignment to be graded needs to be structured. This feature allows the instructor to need only specify the weight of the question once, as opposed to needing to enter the same weight for the same question for each student. Similarly, when the grade is removed, it is removed from all of the students. Points earned component of any grade can be modified or the grade can be removed as seen in figure Figure 4.

When the user saves the document, the grades for each

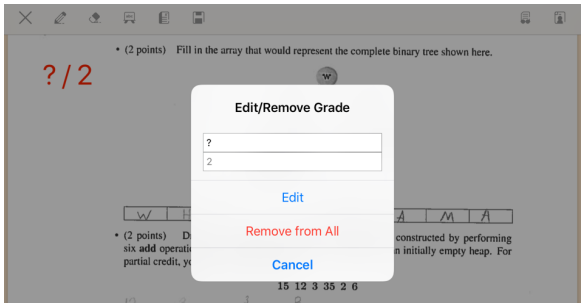

Fig. 4. User editing or removing a grade

Page Number	Question Number	Max Question Points	Anna	Peter	Mike	Rec
Page 2	Question 1	4	4	3	2	1
	Question 2	5	5	4	3	1
Page 3	Question 1	2	2	1	0	1
Page 4	Question 1	2	2	2	1	1
Page 7	Question 1	4	3	4	2	1
Page 9	Question 1	3	1	2	?	1
Page 13	Question 1	4	?	4	3	1

Fig. 5. Grades tabulated per question for each student

question are tabulated per student and saved in a CSV formated file, as seen in figure Figure 5. This file can then be used by instructors outside of the EZ Grader app to calculate metrics desired for the assignment in the instructor's editor of choice.

C. App Architecture

Apple appears to be pushing the use of storyboards for user interface creation, as opposed to creating user interfaces programmatically. As such, the user interface of EZ Grader is created entirely within the following storyboards:

- Launch Screen.storyboard - as the name implies, this storyboard represents the launch screen user interface.
- Main.storyboard - the main storyboard contains the rest of the app, which consists of two views.

Various icons used by the app, such as the main app icon, as well as the navigation bar button icons are located in the Assets.xcassets file.

The views found in the main storyboard are controlled by the following view controllers:

- SelectPDFDocumentsToGradeViewController.swift - this is the controller for the view used to select PDF documents to be graded from the iCloud Drive.
- GradePDFDocumentsViewController.swift - this is the controller for the view which displays the document being graded as a single combined PDF document and allows the user to grade and save the document. When the user is in this view, they are in one of the following five modes:
  - Viewing PDF documents
  - Free-hand annotating
  - Erasing free-hand annotations
  - Keyboard text annotating
  - Adding Grades

GradePDFDocumentsViewController uses a tap gesture recognizer to respond to user taps in the following situations:

- In viewing PDF documents mode - allows the app to check if the user tapped on an existing keyboard text or grade annotation in order to edit or remove the annotation.
- In erasing free-hand annotations mode - allows the app to check if the user tapped on any free-text annotation in order to remove it.

GradePDFDocumentsViewController also responds to user beginning to touch the screen, tracing the screen, as well as lifting the finger or stylus in the following situations:

- Free-hand annotating mode - used to create the shape that the user is tracing on the screen as a Bezier path.
- Keyboard text annotating and adding grades modes - touch begin event provides the touch location where the text annotation or the grade should be added to.

D. PDFKit

EZ Grader makes heavy use of the PDFKit framework in order to display and modify PDF documents. PDFKit was released for iOS in WWDC 2017 [3]. At the core of PDFKit is PDFView, which is the view that holds and presents the PDF document as a PDFDocument type. In EZ Grader, the user first selects the PDF documents to be graded from their iCloud Drive, which are then combined into a single document and presented in a PDFView for grading. When the user toggles between 'Per Page' and 'Per Document' modes, this view is updated to reflect the user's selection by reshuffling the pages in the combined document. PDFDocument type holds the pages of the document, where each page is of PDFPage type. Finally, PDFPage holds user added annotations, with annotation types being subclasses of the PDFAnnotation type. Examples of annotation types include ink - which represents free-hand annotations in EZ Grader, free text - representing keyboard text and grade annotations in EZ Grader, circle, highlight, line and many others.

Annotations have a bounds property, which dictates the location on the page that the annotation will appear on. When the user is adding any kind of an annotation in EZ Grader, the application receives as input the tapped location in the view coordinate space. This coordinate needs to first be translated into the coordinate space of the PDF page that was touched. Fortunately, PDFView makes it easy to both get the page that was touched as well as to translate the touch location from the view coordinate space into the page coordinate space.

As mentioned previously, EZ Grader uses ink PDFAnnotation for free-hand annotations. An ink annotation is represented by a Bezier path, which is essentially just a set of coordinates which are connected to form a curve. When user begins to touch the screen a move to the touched location is performed on the Bezier path. As the user continues to trace the screen, "touch moved" event exposes the new touch location, and a line is drawn from the previous touch location to the current touch location on the Bezier path object. This ultimately creates the curve which represents the user's entire screen trace.

V. Improving Course Experience

EZ Grader improves the course experience for instructors and students in many ways:

- Grading digitally removes the burden instructors have of needing to carry assignments to be graded home and back to the office again. Once the assignments are scanned-in and uploaded to the instructor's iCloud Drive, instructors can use the EZ Grader app to grade the assignments from any location their desire. This can reduce assignment grading time and students can receive feedback sooner.
- Any typographical mistakes or errors made while grading can be quickly erased or corrected, eliminating potential confusion on the part of the student when the assignment is returned to them.
- As each student's grades are tabulated in one convenient CSV file, instructors can import this file into an application of their choice in order to calculate total grades and other metrics desired by the instructors. This next to eliminates the possibility of mathematical errors that sometimes occur when grading paper-based assignments.
- Instructors often desire to go over the results of an assignment with the class. It is not a good use of class time to go over questions that the majority of the class answered correctly. It is therefore wise to prioritize discussing the questions that the students struggled with. In order to determine which questions the students struggled with, instructors would likely need to rely on their memory, or possibly notes taken while grading. The tabulated view of student grades for the assignment eliminates this need, as the instructors can clearly see the maximum grade for each question and each student's grade for that question. Instructors can use this insight to prioritize class time to discuss those questions and the curriculum topics covered by those questions. It is also possible that the question wording is not clear, or maybe the instructor did not spend enough time covering a particular topic. Either way, the tabulated view of student grades allows instructors to see topics and assignment questions that they need to take action on.
- The tabulated view of student grades provides for easy scaling or curving of specific assignment question grades, where instructors just need to modify the maximum points portion for those questions. In general, the single table of all student grades provides the ability for instructors to quickly edit any grade.
- Instructors often do not make copies of graded assignments before returning them to students. This allows for academic dishonesty to creep in. It is possible, especially if the student wrote the assignment in pencil, for the student to correct their answers after the graded assignment is returned to them and ask the instructor why points were deducted for a particular question. In this case, the instructor might have little to no proof as to whether or not the student is being dishonest. Since the

instructor has a digital copy of the graded assignment when using EZ Grader, this possibility is eliminated.

VI. Challenges

Proper development environment setup is key to successful iOS app development. Since iOS development requires a Mac machine as mentioned earlier, a new developer might be tempted to perform development on a virtual machine, the short term cost of which can be significantly lower than that of purchasing a physical Mac machine. However, the performance of the virtual machine will most likely quickly become a major hindrance to productivity. In addition, it is not possible to connect a physical iOS device to a virtual machine in order to run the app under development on them, leaving iOS simulators as the only alternative. Running a simulator on a virtual machine will further reduce performance. In the case of EZ Grader, this would especially be an issue with free-hand annotation feature testing, as the user needs to trace the screen of the device in order to draw on it. Any significant delay experienced when performing this action would make it next to impossible to properly test this feature. In summary, the author does not recommend virtual machine usage for iOS app development.

PDFKit is a relatively new framework for iOS, having just been released in WWDC 2017. As such, it is undergoing changes with subsequent Xcode releases. Xcode version was upgraded in the middle of EZ Grader development in order to support the latest iOS version running the iPad and iPhone devices that were used to test EZ Grader. This both introduced new issues and fixed existing issues. For example, when running on a previous version of Xcode, if the user was viewing the PDF in portrait mode, tapping and holding a particular location of the PDF in order to provide a temporary zoom window would display the zoom window in the opposite orientation. In addition, PDF annotations added to the document would no longer display when the document was saved and re-opened using PDFKit, even though all the annotations would still be visible when viewing the graded PDF using an external PDF viewer. Both of these issues were apparently fixed with an Xcode upgrade. This same Xcode release broke PDFPage copy functionality though - creating a copy of the page, which worked fine before, would now create a blank page. In conclusion, PDFKit is in active development, it being a relatively young framework. App developers should expect to have to deal with various issues as the framework matures.

VII. User Experience and Future Work

This section is intended to summarize the author's experience with the app and provide suggestions for future work. While EZ Grader implements core PDF annotation features that instructors would need, it does not match the experience the user has via generic PDF annotation apps, such as the Files iOS app, in terms of their feature sets. There are also several known issues with the EZ Grader app.

Allowing the user to select the color of the pen and pen width when free-hand annotating would be a nice enhancement. There is also sometimes an issue with the free-hand annotation flickering as the user is drawing it. When the user is free-hand annotating while zoomed in on a certain location of the page, the shape drawn often does not show up until the finger is lifted. This particular issue appears to be a defect with the PDFKit.

As the user is free-hand annotating in the Files app, they can undo their work, where each undoing erases the most recently drawn fully-connected drawing. EZ Grader only allows users to erase the entire drawing. Furthermore, the Files app allows the user to erase the connecting shape that is rubbed with the finger or stylus. EZ Grader relies on the bounds of the Bezier path to detect if the user is touching the annotation in order to erase it. The bounds of the Bezier path is a simple rectangle that is inscribed inside the area bounded by the extreme top, bottom, left and right coordinates of the Bezier path. If the user draws something in the upper left corner of the page and in the right bottom corner of the page, the bounds of the Bezier path essentially become the entire page, therefore tapping anywhere on the page would erase it. This is certainly an area of the app that could be improved.

It would also be preferable to allow the user to change the font and font size when adding keyboard text and grade annotations. Furthermore, it would be desired to allow the user to drag keyboard text and grade annotations to another location of the page.

EZ Grader currently saves graded documents and the tabulated grades file into the documents directory of the app sandbox. In order to view the output artifacts, users must use Xcode to download the app container, which is not something that the average user would want or know how to do. This is really a must do as far a future work is concerned.

There is also currently a bug in EZ Grader, where the user is unable to add annotations past the first page after saving the assignment. Toggling the grading mode appears to fix the issue, but a more appropriate fix is needed long term.

It might also be useful to add a feature to the app to allow instructors to return graded assignments to students via email. This would reduce the time that the students need to wait to receive feedback on an assignment. Without this feature, instructors might also be tempted to reprint all of the graded assignments, doubling paper and printer usage. Instructors might even place these assignments in public places for students to retrieve, introducing privacy concerns. Email feature would eliminate these concerns, further improving the course experience.

VIII. Conclusions

EZ Grader is a step in the direction of moving assignment grading into the digital space, improving the class experience for everyone. While it attempts to implement many of the core PDF annotation features and even introduces features that are not found in generic PDF editors, there is still quite a lot left to be desired, both due to apparent bugs with the underlying

PDFKit framework as well as features not present in the app. It is the author's hope to see the features and bugs outlined in the previous section addressed in the future. This requires users to download the app container

ACKNOWLEDGMENTS

The author thanks Dr. James Heliotis, professor in the computer science department at the Rochester Institute of Technology, for the app idea and his guidance and support during the ideation and implementation phases of the development of the app.

The icons for the various buttons in the navigation bar of the EZ Grader app are designed by Freepik and were obtained from www.flaticon.com free of charge.

REFERENCES

[1] Apple, Inc. (2018). Files (Version 11.3) [Mobile application software]. Retrieved from Apple App Store.

[2] Schoology, Inc. (2018). Schoology (Version 5.0.0) [Mobile application software]. Retrieved from Apple App Store.

[3] Apple, Inc., 'Introducing PDFKit on iOS', 2017. [Online]. Available: https://developer.apple.com/videos/play/wwdc2017/241/. [Accessed: 6- May- 2018].