

**ClassView  
Project Documentation - Spring 2014**

**TechSmith Corporation Representatives**

Glenn Hoeppner

Bill Scanlon

**Michigan State University Capstone Students**  
Eric Austin  
Rebecca Collins  
Kyle Woodward

**Table of Contents**

1. Executive Summary...……………………………………………………………………..…4

2. Core Functionalities…………...…………………………………………………………….5

Primary usage and abilities of teacher and student users

2.1 Teacher Dashboard

2.1.1 Classes

2.1.2 Assignments

2.1.3 Progress Notifications

2.2 Student Dashboard

2.4 Use Cases

2.4.1 Homework Flow from Assignment to Completion

2.4.2 Archiving a Class

2.4.3 Archiving a Student

3. Design Specifications....……………………………………………………………………..8

Details about the user interface

3.1 User Interface

3.2 User Login

3.3 Teacher Dashboard

3.4 Student Dashboard

4. Technical Specifications...…………………………………………………………………11

Details about technology utilized

4.1 System Architecture

4.2 User Interface

4.3 Teacher and Student Dashboards

5. Development Standards...…………………………………………………………………12

Information on how application will be developed

5.1 Code Conventions and Style

5.2 HTML

5.3 JavaScript

5.4 CSS

5.5 Accessibility

6. Risks and Mitigation...…….......……………………………………………………………13

Potential problems and how they were handled

6.1 Minimal Previous Experience  
6.2 Extent of API Capabilities  
6.3 Client Were Unfamiliar With Technologies Utilized  
6.4 Google's Frequent Changes

7. Future Features.........…….......……………………………………………………………14

Additional features that could improve ClassView

7.1 Groups  
7.2 Sort By Assignment View  
7.3 Single Application  
7.4 Domain Security  
7.5 File Sorting  
7.6 Improved E-mail Notifications

8. Timetable ……………………………………………………………………………………16

9. Point of Contact.……………….…………………………………………………………...18

10. Acknowledgements..………….…………………………………………………………..19

**1. Executive Summary**

The purpose of this section is to briefly summarize the extent of this document.

TechSmith Corporation is a software company located in Okemos, Michigan. It is well known for creating screen capture, recording, and editing software. It also develops software for visual collaboration and learning.

TechSmith works with teachers, principals, superintendents, professors, industry leaders, IT coordinators, parents, and students to learn about the challenges they face and how we can help address them. They are focused on providing software solutions that help at every level of education.

The use of Google Applications for Education, Google Docs, and Google Drive are increasing at a rapid rate, especially K-12. TechSmith’s products are embracing the Google platform as well. This project will help TechSmith explore how we can improve the learning experience for schools that have adopted Google Docs and Google Drive.

ClassView will provide both a teacher and student dashboard view of Google Drive. This project will implement a Chrome Application that provides an alternative view which models the classroom, helps with content management, facilities collaboration on assignments, and exposes learning activity.

**2. Core Functionalities**

This section explains the major components of ClassView and their uses.

**2.1 Teacher Dashboard**  
The Teacher Dashboard allows the user to model a classroom environment and act as a teacher by building classes and assigning homework through Google Drive. Teachers can also interact with students and will be notified when student users have changed the status of their homework assignment. The Teacher Dashboard was built to facilitate learning, organization and progress in the classroom.

**2.1.1 Classes**The teacher is able to create a class and add students to that class. Within these classes a teacher will be able to send assignments or files to individual students. A student will be notified when they have been added and given access to the common folders. They will also be able to view assignments sent by their teacher. When a teacher is done with a class, they can hide it from their dashboard by archiving it. Likewise, if a student drops out of a teacher’s class, they can be archived as well.

**2.1.2 Assignments**Once a student has been added to a class, a teacher can share assignments with them. The teacher can choose to upload an existing file or to create a new Google Document for an assignment. The teacher can then assign that file to any student in that class.

**2.1.3 Progress Notifications**  
In ClassView, students are able to mark their assignments as completed or needing help. Teachers will receive visible indicators when a student has changed the state of their assignment. Teachers know that the student is ready to have their assignment graded when they have marked their assignment as “Completed”. Teachers can also open up a student’s file and have a discussion when the student has asked for help on an assignment.

**2.2 Student Dashboard**  
When a teacher has assigned homework to a student through ClassView, the student can use the dashboard to keep track of their work. Assignments are neatly sorted by class. The Student Dashboard contains features that help students interact with their teacher and get help when they need it. The Student Dashboard will be auto populated with the classes that a teacher has added the student to. When a student has been added to a class, they will receive a notification by email.

**2.3 Use Cases**

**2.3.1 Homework Flow from Assignment to Completion**

1. The teacher chooses a file to upload and a student to share it with.

2. The student receives an email notification of the new file and the file appears in their dashboard with a “New” symbol indicating that the student has not yet viewed the file.

3. The student opens the homework and does the work required.

4. If the student needs help with the assignment, they can click the “Ask for Help” button. There is now an indicator marked under the “Help Needed” column.

5. The teacher then receives an email notification that a student needs assistance with the file. In the Teacher Dashboard, the file is marked as “Help Needed”.

6. The teacher can view the file and once the appropriate assistance is given, click “Resolve”. This removes the “Help Needed” marking from the file in the Teacher Dashboard.

7. The student then will receive an email notification that their assignment has been reviewed by the teacher. In the Student Dashboard the assignment has an indicator under the “Feedback” column.

8. The student may then review any assistance received and ask from more help or complete their work on the file. When finished, the student clicks the “Mark as Complete” button. Their file is then marked under the “Complete” column in both the Student and Teacher Dashboards.

9. The teacher receives an email notification that the file is complete and may grade or assess the student’s work.

**2.3.2 Archiving a Class**

1. To archive a class a teacher must first click “ClassView Settings” in the Teacher Dashboard.
2. Next, the teacher clicks “Archive a Class” and selects the class they wish to archive.
3. Finally, the teacher clicks the “Submit” button and the class is moved to the archive list and the Teacher Dashboard switches to a view of all archived classes.
4. To remove a class from the archive list, the teacher simply clicks “Remove Class from Archive List” instead of “Archive a Class”. Once the class are selected and the teacher clicks “Submit”, the class are now back in the current view of the Teacher Dashboard.

**2.3.3 Archiving a Student**

1. To archive a student, a teacher must first have the class to which the student belongs selected and then click “ClassView Settings’ in the Teacher Dashboard
2. Next, the teacher clicks “Archive a Student” and selects the student they wish to archive.
3. Finally, the teacher clicks the “Submit” button and the student is moved to the archive list and the class switches to a view of all archived students in the class.
4. To remove a student from the archive list, the teacher simply clicks “Remove Student from Archive List” instead of “Archive a Student”. Once the student is selected and the teacher clicks “Submit”, the student is now back in the current view of the class.

**3. Design Specifications**

This section further explores the functionalities of the ClassView dashboards and their design.

**3.1 User Interface**

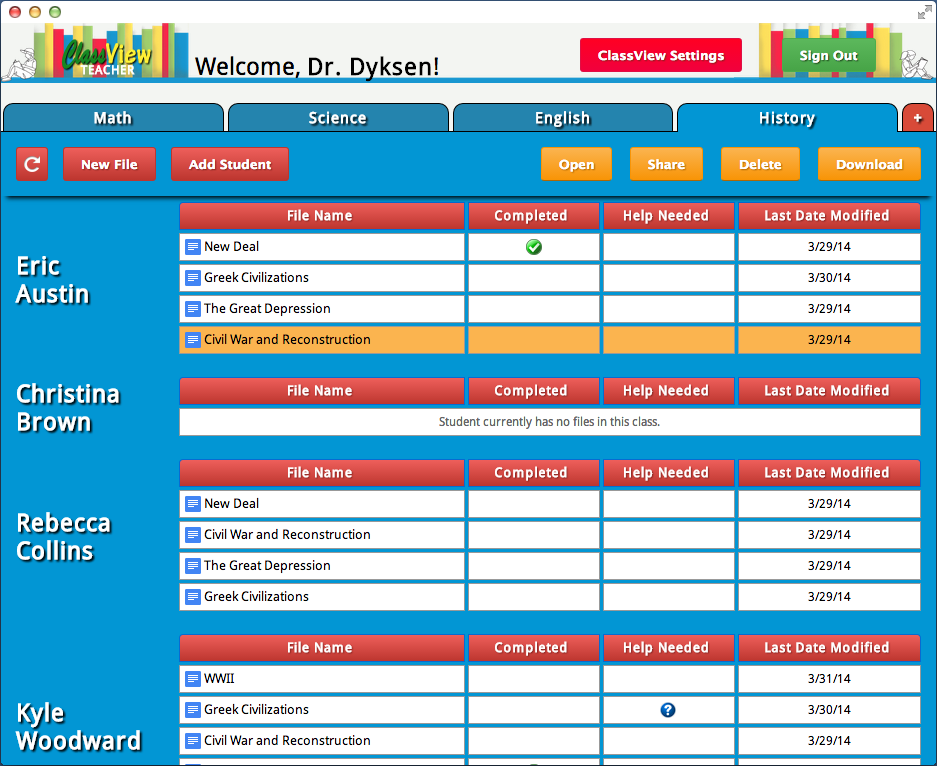
The user interface is a very important part of ClassView. Teachers need to easily be able to navigate and understand how to use ClassView so they can monitor their classroom through it. The user interface also needs to be intuitive enough that children of a wide range of ages (K-12) will be able to understand how to use it.

**3.2 User Login**

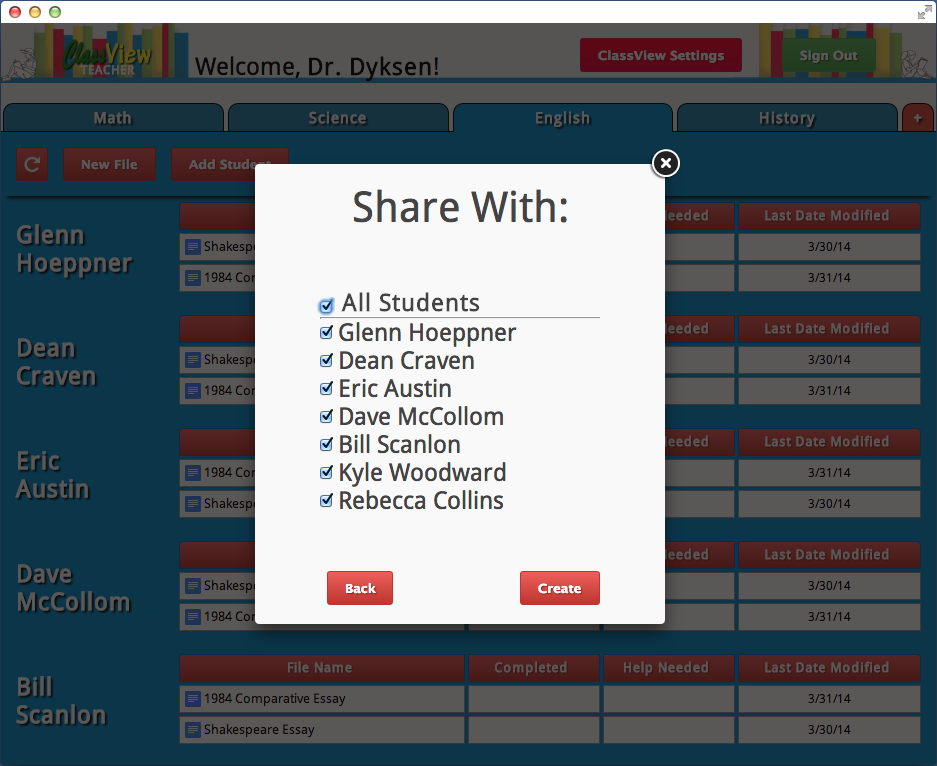
Using OAuth 2.0, when users click the “Sign In” button an authorization token will be requested from Google’s servers and whichever user is currently signed into the Google Chrome will be signed into ClassView.

**3.3 Teacher Dashboard**

On the teacher dashboard, the user is presented with a tabbed view of all the classes that they run. They can click the tab with the plus sign on it to create a new class. Once they have selected a class, they have the opportunity to add students and assignments. As students and assignments are added to the class, they will be displayed on the screen, sorted by student. Teachers can then sort a student’s files by a specific property by clicking the column header. By clicking on a file from the listing, a button bar will appear so teachers can open, download, or delete the file. The button bar also allows teachers to share a student’s file with another student, and lets teachers notify students when they have responded to a student’s request for help.



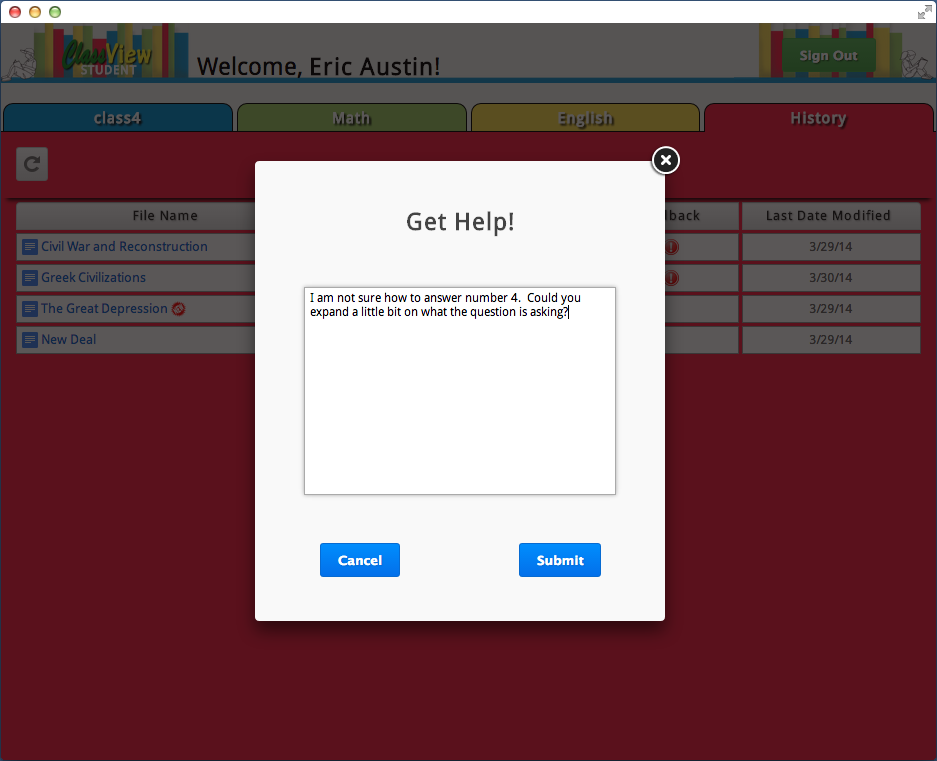
*Figure 3.3.1 - Viewing a class in the Teacher Dashboard*



*Figure 3.3.2 - A teacher’s view of assigning a file*

**3.4 Student Dashboard**

On the student dashboard, the user is presented with a tabbed view of all the classes that they are in. By selecting a specific class tab, all of their assignments within that class will be listed. Students can then sort the files by a specific property by clicking the column header. By clicking on a file from the listing, a button bar will appear so students can open or download the file. The button bar also allows students to ask for help from their teacher on a specific assignment or mark that assignment as complete.



*Figure 3.4.1 - Requesting help on the Student Dashboard*

**4. Technical Specifications**

This section explains the technical aspects that create the program.

**4.1 System Architecture**

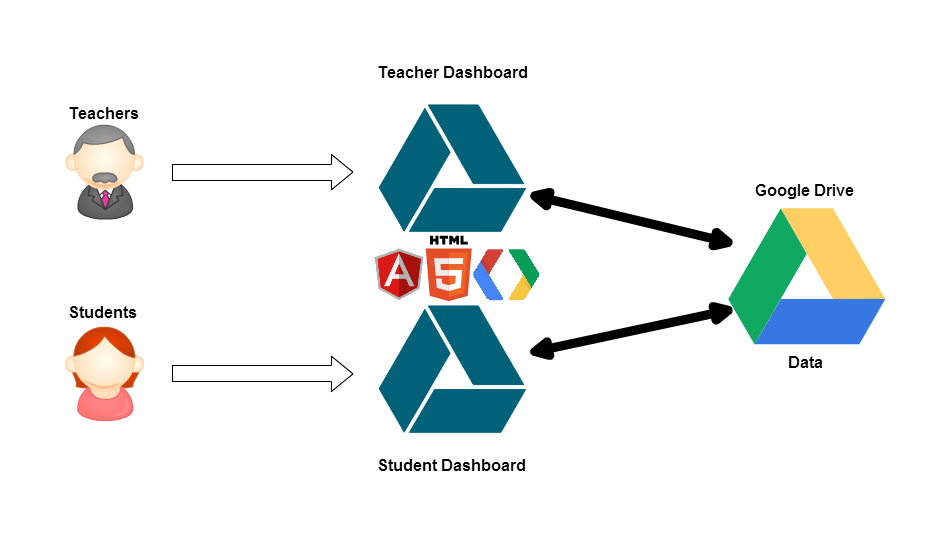
There are three parts to the ClassView system:

1. Teacher Dashboard

1. Student Dashboard

2. Google Drive

Teachers and students will use their respective dashboards to accomplish their work. The dashboards provide an easy to use graphical user interface that allows users to manipulate their files on Google Drive.



*Figure 4.1.1 – System Architecture*

**4.2 User Interface**

The user interface of ClassView is run through two Google Chrome applications, the Teacher Dashboard and the Student Dashboard.

**4.3 Teacher and Student Dashboards**

Both of the dashboards are written in JavaScript and use the AngularJS framework. All the interactions with Google Drive are run through HTTP protocols to the Drive API in the Google Drive SDK. This allows us to create and manipulate files on a user’s Google Drive storage.

**5. Development Standards**  
This section discusses the standards used while developing ClassView.  
  
**5.1 Code Conventions and Style**

- We will follow TechSmith’s coding standards and conventions

**5.2 HTML**

- Must be validated (<http://validator.w3.org/>)

**5.3 JavaScript**

- We will be using Jquery and AngularJS frameworks

**5.4 CSS**

- Must be validated (<http://jigsaw.w3.org/css-validator>)

**5.5 Accessibility**

- Must verify problematic areas in accessibility (<http://www.w3.org/TR/WCAG10/full-checklist.html>)

**6. Risks and Mitigation**

This section addresses potential problems faced by the team and how they were overcome.

**6.1 Minimal Previous Experience**

The team had little to no experience with the Chrome Application APIs, HTML, CSS, JavaScript and AngularJS. It was mitigated by working through various tutorials and code samples to learn.

**6.2 Extent of API Capabilities**  
The risk was related to risk 6.1 in that the group members' lack of knowledge left us unable to determine if all project goals were achievable within the given constraints of the Chrome Application APIs. This was mitigated when alternative solutions were considered by both the group and the TechSmith representatives.

**6.3 Client Were Unfamiliar with Technologies Utilized**Neither of our contacts had previously worked with the Google APIs or the Google Drive SDK. This made getting advice on design decisions difficult. This was mitigated by the team familiarizing itself with some basic information and by getting in touch with another TechSmith employee who had previous experience.

**6.4 Google's Frequent Changes**Originally, the Student Dashboard was just Google Drive loaded in a window with JavaScript and CSS injections to manipulate it how we wanted. The risk in this is that Google frequently updates their services, which would lead to the Student Dashboard breaking without our knowledge. The team mitigated this task by rebuilding the Student Dashboard in a similar fashion to how the Teacher Dashboard was built.

**7. Future Functionality**

This section discusses additional features that would improve ClassView if implemented.

**7.1 Groups**

Initially a requirement for our project, but was downgraded to a nice-to-have feature. We could make use of the Google Apps Groups Settings API to create and update groups with a unique group email address. This group would be treated as a student in the Teacher Dashboard and would be easy to implement. However, the Student Dashboard would add complications as we would have to pull files for the group email if the student was within the group. We would need to find a way to store the group information, which would be difficult at the moment without a backend.

**7.2 Sort By Assignment View**  
While using the Teacher Dashboard, files are sorted by the student that they are assigned to. If we added the option to toggle to a view that was sorted by assignment, teachers would have an easier time monitoring a specific assignment that the class was working on. The teacher could see all of a specific assignment grouped together, instead of having to scroll through every student and looking for that particular assignment.

**7.3 Single Application**When we initially started development on ClassView, we had planned on the Teacher Dashboard and the Student Dashboard having very different functionality. Because of this, we developed them as two separate applications. Along the way, we completely overhauled the Student Dashboard and rebuilt it based off of the Teacher Dashboard. Because of this, there is a lot of reused code between the two applications. If we had all of the ClassView functionality within a single application, when the user clicks login we would want an option to ask the user if they wanted to login as a teacher or as a student. After selecting, the user would be presented with the proper dashboard.

**7.4 Domain Security**The original project description brought up the idea of having teachers monitor their students’ files and receive alerts if that student tried to share their files outside of the school’s domain. As the team progressed in the project, it became apparent that this was outside of the scope of what the team was capable of doing. There are ways to have this domain security, but the practical way to accomplish it would be through an organization (how many schools run on Google), not through an application like ClassView.

**7.5 File Sorting**The application lists files in the order in which they are pulled from Google Drive using the API. We would like users to be able to click on the column headings to be able to sort by that column in ascending or descending order. For example, this would allow users to sort files alphabetically or by most recently modified.

**7.6 Improved E-mail Notifications**The current e-mail notifications are generated through the Google Drive API and while they contain a descriptive message, they still link to Google Drive. Ideally we would want the hyperlink to open the Student Dashboard instead.

**8. Timetable**  
This section outlines the weekly goals of the team.

**Week 1 (1/5-1/11)**

- Received project description  
  
**Week 2 (1/12-1/18)**

- Initial Meeting with TechSmith: 1/13

- Introducing team members to new technologies

- Working through samples and example Chrome Applications

- Working on Status Report  
  
**Week 3 (1/19-1/25)**

- Finalize Status Report  
- Status Report Presentation: 1/22

- Work on Project Plan

- Create mockups to show TechSmith

- Continue working through examples to learn new technology  
  
**Week 4 (1/26-2/1)**

- Finish Project Plan  
- Project Plan Presentation: 1/27

- Create new mockups for teacher and student dashboards

**Week 5 (2/2-2/8)**  
- Student Dashboard loads Google Drive within a webview (iframe)

- Teacher Dashboard displays all files from a logged in user’s Google Drive

**Week 6 (2/9-2/15)**

- Classes loaded dynamically off of Google Drive

- Teachers can now create files from the Teacher Dashboard

- Teachers can create classes

- Prepare for Alpha Presentation

**Week 7 (2/16-2/22)**

- Work on implementing teacher notifications  
- Drag and drop functionality added to assignment upload

- UI is now hidden when a user is not logged in

- Design switched to a new tabbed layout

**Week 8 (2/23-3/1)**

- Teachers can now upload local files as assignments

- Added sharing functionality to new assignments

- Alpha Presentation: 2/26

**Week 9 (3/2-3/8)**

- Students are now represented by shared folders within the teacher’s class folder

- Added button bar to open, download, and delete files from the Teacher Dashboard

**Week 10 (3/9-3/15)**

- Implement progress tracking on assignments

- Automatic refresh added after file assignments and sharing

- Added functionality to assign and share files with the entire class

**Week 11 (3/16-3/22)**

- Class and file assignments from the Teacher Dashboard perfected

- Completely redid the Student Dashboard, now very similar to the Teacher Dashboard

- Added indicator for unviewed assignments in the Student Dashboard

**Week 12 (3/23-3/29)**

- Added buttons to Student Dashboard for request help and marking as complete

- Message displays when a student is in a class but has no files

- Student Dashboard now notifies students of teacher feedback on files

- Instructions for teachers on their first sign in to the Teacher Dashboard

**Week 13 (3/30-4/5)**

- File listings can now be sorted by clicking on column headers

- Beta Presentation: 3/31

- Finish Project Plan

**Week 14 (4/6-4/12)**

- Record project video and audio

- Final bug fixes

**Week 15 (4/13-4/19)**

- Edit project video

- Final bug fixes

- Prepare deliverables

**Week 16 (4/20-4/26)**  
- Project Video: 4/21  
- All Deliverables Due: 4/23  
- Design Day Setup: 4/24

- Design Day: 4/25

**9. Point of Contact**

For further information concerning this document and project, please contact Professor Wayne Dyksen at Michigan State University ([dyksen@cse.msu.edu](mailto:dyksen@cse.msu.edu)).

**10. Acknowledgements**

We would like to acknowledge TechSmith Corporation for their contributions and collaborate to the Senior Design program at Michigan State. Specifically we would like to thank Glenn Hoeppner and Bill Scanlon for dedicating much of their time to meeting with and guiding us through this project.

Furthermore, we would like to thank the Michigan State University Department of Computer Science, Dr. Wayne Dyksen, and Malcolm Doering for their resources, counseling, and support.