5-2022

Designing and Evaluating Accessible E-Learning for Students with Visual Impairments in K-12 Computing Education

Earl W. Huff Jr
earlh@clemson.edu

DESIGNING AND EVALUATING ACCESSIBLE E-LEARNING FOR STUDENTS WITH VISUAL IMPAIRMENTS IN K-12 COMPUTING EDUCATION

A Dissertation
Presented to
the Graduate School of
Clemson University

In Partial Fulfillment
of the Requirements for the Degree
Doctor of Philosophy
Human-Centered Computing

by
Earl W. Huff, Jr.
May 2022

Accepted by:
Dr. Julian Brinkley, Committee Chair
Dr. Guo Freeman
Dr. Paige Rodeghero
Dr. Murali Sitaraman

ABSTRACT

This dissertation explores the pathways for making K-12 computing education more accessible for blind or visually impaired (BVI) learners. As computer science (CS) expands into K-12 education, more concerted efforts are required to ensure all students have equitable access to opportunities to pursue a career in computing. To determine their viability with BVI learners, I conducted three studies to assess current accessibility in CS curricula, materials, and learning environments. Study one was interviews with visually impaired developers; study two was interviews with K-12 teachers of visually impaired students; study three was a remote observation within a computer science course. My exploration revealed that most of CS education lacks the necessary accommodations for BVI students to learn at an equitable pace with sighted students. However, electronic learning (e-learning) was a theme that showed to provide the most accessible learning experience for BVI students, although even there, usability and accessibility challenges were present in online learning platforms.

My dissertation engaged in a human-centered approach across three studies towards designing, developing, and evaluating an online learning management system (LMS) with the critical design elements to improve navigation and interaction with BVI users. Study one was a survey exploring the perception of readiness for taking online courses between sighted and visually impaired students. The findings from the survey fueled study two, which employed participatory design with storytelling with K-12 teachers and BVI students to learn more about their experiences using LMSs and how they imagine such systems to be more accessible. The findings led to developing the accessible learning content management system (ALCMS), a web-based platform for managing courses, course content, and course roster, evaluated in study three with high school students, both sighted and visually impaired, to determine its usability and accessibility. This research contributes with recommendations for including features and design elements to improve accessibility in existing LMSs and building new ones.

DEDICATION

I dedicate this dissertation to everyone who believed in me, supported me, and played an instrumental part in helping get to this monumental moment in my life. A special dedication to all the Black and Brown youths who look for examples of people in S.T.E.M. who look like them and wonder if they belong in this space.

ACKNOWLEDGMENTS

This six-year journey would not have been possible without the support of so many people I had the opportunity to meet throughout my studies. First and foremost, I give honor and praise to God; without him, none of this would be possible. Thank you to my parents, Earl Sr and Victoria, and older sister, Natashia, who have had my back since I first started down this path, and consistent prayers over my life.

Thank you to all my wonderful friends that I connected with during my studies. Thank you to the people from the BLK Doc Students group for the community, support, and writing groups over the years. A special thank you to my ‘scholar siblings’, Tonelli and LaShica, for the bond we developed during the journey and for having my back through thick and thin. I want to thank my dissertation committee members of Drs. Paige Rodeghero, Gou Freeman, and Murali Sitaraman for their mentorship and support as my research converted from conception to reality. A special thank you to my chair and advisor, Dr. Julian Brinkley, who took a chance on me as his first graduate student and member of his lab, for his mentorship and guidance during the past three and half years. I want to acknowledge Dr. Juan E. Gilbert for his mentorship and opportunities through the Institute for African American Mentoring in Computing Sciences (iAAMCS). I am incredibly grateful for Dr. Kinnis Gosha, who introduced me to research and helped mold me into a strong scholar. Last but certainly not least, a special thank you to my good friend and long-time mentor, Dr. John Robinson. You took me under your wing and helped me see my potential. I will always be grateful for your support, guidance, words of encouragement, and friendship.

TABLE OF CONTENTS

Page

TITLE PAGE i

ABSTRACT ii

DEDICATION iv

ACKNOWLEDGMENTS v

LIST OF TABLES x

LIST OF FIGURES xi

CHAPTER

I. INTRODUCTION 1

II. BACKGROUND AND RELATED WORK 5

Visual Impairment 5
Programming and Software Development for the Visually Impaired 6
Assistive Technology Solutions 10
Computing Education for The Visually Impaired 13
Conclusion 14

III. EXAMINING THE WORK AND EDUCATIONAL EXPERIENCE OF VISUALLY IMPAIRED PROGRAMMERS 16

Introduction 16
Method 17
Findings 20
Conclusion 31

IV. EXPLORING THE PERSPECTIVES OF TEACHERS OF THE VISUALLY IMPAIRED REGARDING ACCESSIBLE K12 COMPUTING EDUCATION 33

Introduction 33

Table of Contents (continued)

Page

Method 34
Findings.......... 37
Discussion 45
Conclusion 48

V. A DIARY STUDY OF THE TEACHING AND LEARNING EXPERIENCE IN A HIGH SCHOOL PROGRAMMING COURSE 49

Introduction 49
Research Questions 50
Participant Recruitment and Demographics 50
Study Apparatus 51
Procedure 51
Case Study 52
Discussion and Conclusion 58

VI. SUMMARY OF INVESTIGATION 60

Introduction 60
Curriculum and Learning Materials 61
Developer Tools 62
Perceived Advantages of Visual Impairment 63
Online Learning and Electronic Access 64
Discussion and Conclusion 66

VII. A SURVEY OF READINESS FOR AND EXPERIENCE WITH ONLINE LEARNING 69

Introduction 69
Research Questions 70
Participants 72
Instrument 74
Procedure 75
Results 76
Discussion 85
Conclusion 88

VIII. A STORY-DRIVEN PARTICIPATORY DESIGN OF AN ONLINE LEARNING MANAGEMENT SYSTEM 89

Table of Contents (continued)

Page

Introduction 89
Research Questions 92
Method 92
Design Session One 95
Design Session Two 99
Design Session Three 105
Conclusion 107

IX. THE ARCHITECTURE AND IMPLEMENTATION OF A LEARNING MANAGEMENT SYSTEM DESIGNED FOR ACCESSIBILITY 110

Introduction 110
System Overview 110
System Architecture 111
Client Component 112
Server Component 113
Cloud Computing 113
Features and Functionality 114
Conclusion 124

X. THE EVALUATION OF THE ACCESSIBLE LEARNING CONTENT MANAGEMENT SYSTEM 126

Introduction 126
Research Questions and Hypotheses 126
Method 128
Results 132
Discussion 139
Conclusion 142

XI. CONCLUSION 145

Introduction 143
Contributions of The Dissertation 144
Future Work 149
Final Remarks 150

APPENDICES 151

Table of contents (continued)

Page

A. Study 1 Interview Questions 152
B. Study 2 Interview Questions 154
C. Study 3 Student Diary Study Questions.. 156
D. Study 3 Teacher Diary Study Questions 158
E. Readiness for Online Learning Survey 160
F. Participatory Design Student Contextual Story 165
G. Participatory Design Teacher Contextual Story 166
H. Participatory Design Student Conceptual Story 167
I. Participatory Design Teacher Conceptual Story 169
J. Student LMS Evaluation Survey 171

REFERENCES 175

LIST OF TABLES

Table Page

2.1 List of navigation difficulties and number of participants 9

3.1 Study 1 participant demographics 19

4.1 Study 2 participant demographics 36

5.1 Student ratings of course 53

5.2 Teacher ratings of students 54

6.1 Major themes from the investigative studies 61

7.1 Relationship between research questions and measures 71

7.2 Student demographics 74

7.3 Student Readiness for Online Learning Descriptive Statistics 77

7.4 Student perception of importance by demographic 80

7.5 Student perception of confidence by demographics 80

8.1 Summary of key findings from each session 109

9.1 User need and associated system feature(s) 111

10.1 Relationship among research questions, hypotheses, and measures 127

10.2 Student demographics 129

10.3 Task descriptions 132

10.4 Means of difficulty of tasks performed 134

10.5 SUS scores of ALCMS by students 135

List of Tables (cont.)

Table Page

10.6 Mean scores of technology acceptance model constructs................................136

11.1 Major contributions of dissertation..145

LIST OF FIGURES

Figure Page

8.1 Sequential progression of participatory design methodology 95

8.2 HTML semantic element layout 102

8.3 Accordion element for Introduction to A.I. module 104

9.1 ALCMS high-level architecture 112

9.2 User registration form 115

9.3 Login form for the ALCMS 115

9.4 Landing page for the ALCMS with the sidebar Navigation collapsed 117

9.5 Profile picture upload widget 118

9.6 Create course page 119

9.7 Course homepage shows an expanded module 120

9.8 Course roster for viewing users assigned to a course 121

9.9 Create lecture page with a rich text editor 122

9.10 File upload page for lectures 122

9.11 Code editor for practicing web development with a preview pane 124

CHAPTER ONE

INTRODUCTION

According to the US Bureau of Labor Statistics, the computing technology industry is experiencing one of the highest job growths in the United States, according to the US Bureau of Labor Statistics [1]. There is a 24 percent increase in software developer positions between 2016 and 2026 [1]. The rise in technology-related jobs comes with the demand for a pool of qualified applicants to fill such positions. However, there are currently more computing-related openings than qualified candidates [2]. According to Code.org, less than 50,000 Computer Science graduates were available in 2017; but over 500,000 computing jobs were available nationwide [3]. In 2020, an estimated 1 million computing jobs will go unfilled [4]. The problem may lie in the lack of diversity in computing. To this day, the field of computing is dominated primarily by males of Caucasian or Asian descent, while women and minorities are grossly underrepresented. Comparing diversity reports from 2014 to 2019, representation of women in tech jobs saw an increase at Apple (+3%), Facebook (+8%), Google (+6%), and Microsoft (+3%), but still in disproportion to their male counterparts [5]. In the same 5-year span, Black and Latinx representation in tech jobs saw an average increase of one percentage point [5]. New programs and organizations are working to increase awareness and access to computing across the US. Code.org is a non-profit organization that provides resources to K-12 schools in the form of computer science curricula for educators to use for teaching students [3]. The Hour of Code campaign is a global initiative in which participating schools dedicate one hour to coding tutorials and exercises. CS for All [6] is a federally funded initiative providing resources for K-12 educators, researchers, and companies to help provide high-quality computing education to children across the United States.

Blind or visually impaired (BVI) persons, who make up 2.3 percent of the US population, with less than half of whom are of working age (18+) and employed [7], are a population that may benefit from efforts in broader participation. Research suggests that BVI persons may struggle with finding employment because of a) the visually demanding nature of certain tasks and b) the abelist perceptions of employers regarding their competence [8]. However, as the tech industry may benefit from the broader participation of persons with programming skills, employers that initially overlooked BVI persons may see them as potential candidates. Factors may include a) the latest advances in assistive technology (AT) to perform tasks such as walking, wayfinding, and using smartphones; and b) a significant shift towards technology improving proficiency in using a computer, such as screen readers, magnifiers, and braille displays [8], [9]. The prospect of BVI persons working in the computing industry does not come without significant obstacles. One major obstacle may be the accessibility of current tools and environments for writing code. Many of today's code editing software requires performing more visually demanding actions that may put BVI persons at a disadvantage compared to their sighted counterparts. Even with the most capable of screen reading software [10]–[12], the tools' user interface (UI) can be challenging to navigate due to the complexity and lack of accessibility. A second major obstacle may be the accessibility and accommodation in BVI students' training in computing. Prior studies suggest BVI students may not receive adequate accommodations in their learning of computing concepts and programming [13], [14]. As a result, visually impaired students may fall behind in their learning progress. Additionally, instructors may not provide learning materials accessible to students with visual impairments. The lack of support can be detrimental to students' continued progress and deny them a fair opportunity to demonstrate their ability to write software code on the same level as someone who is sighted. Consequentially, this will impact how future employers view the likelihood of hiring someone with visual impairment due to their perceptions about their capabilities as developers.

My research will explore the accessibility of computer science education (CSEd), including existing computer science (CS) curricula and their materials, the learning tools and environment, and the institutional and teacher support for BVI learners. My research will answer two questions:

RQ1: What are the perceptions of blind or visually impaired persons regarding the accessibility of the curricula, tools, and pedagogies in K-12 computer science education?

RQ2: What interventions can improve accessibility for blind or visually impaired persons in K-12 computer science education?

To answer RQ1, I investigated the current state of accessibility in the tools and education for BVI students learning to write computer software. The investigation consisted of three studies across multiple populations to obtain different perspectives on BVI students' current challenges in learning to code. The first study was an exploratory study from BVI persons who work as professional software developers in the tech industry. This reflective investigation involved asking participants about their experience learning the skills necessary for their occupation and their struggles to overcome to become proficient in their craft. The second study focused on K-12 educators with experience teaching BVI students within computing and programming classes. The focus was on strategies employed to teach BVI students, the accommodations provided, the challenges they faced during the lectures, and any tools or additional resources that BVI students used to supplement their learning. The final study focused on high school students with visual impairments enrolled in a computing class. It is vital to observe the use of current software tools within the user's environment to understand the context of its use better. This phase involved a diary study of the teaching and learning process within a computer science course. The findings from the exploratory studies revealed that web-based technologies, electronic materials, and online learning platforms provide the most accessible learning experience for BVI students. For RQ2, I explored the design and development of a prototype computational artifact using user-centered design (UCD) that satisfactorily addresses the concerns raised from the exploration and provides a solution that may facilitate accessible computing education for BVI learners.

CHAPTER TWO

BACKGROUND AND RELATED WORK

Visual Impairment

Visual impairment is defined as the functional limitation of the eye(s) or vision system [15]. Visual impairment can mean loss of visual acuity or field, double vision, difficulties of perception, and visual distortion. People with a visual impairment fall into two main types of vision loss: low vision or blindness. The Center for Disease Control and Prevention (CDC) defines low vision as having a visual acuity of between 20/70 and 20/400 in the better-seeing eye with corrective glasses or contacts or a visual field of 20 degrees or less [16]. Blindness is defined as having a visual acuity of 20/400 or worse in the better-seeing eye with corrective glasses or contacts or a visual field of 10 degrees or less [17]. In the United States, legal blindness is having a visual acuity of 20/200 or worse or a visual field of 20 degrees or less.

There is a multitude of causes for visual impairment. One of the most common causes in the US is refractive errors [16]. Refractive errors include myopia (nearsightedness), hyperopia (farsightedness), presbyopia (inability to focus close up), and astigmatism (focus problems caused by the cornea) [18]. Diabetic retinopathy, a complication resulting from diabetes, is the leading cause of blindness. Age-related macular degeneration (AMD), an eye disorder associated with aging [19], cataract (clouding of the eye's lens) [20], and glaucoma (a disease that damages your eye's optic nerves) [21] are additional common causes of vision loss in the US.

In the US, 2.3 percent of the population has a visual impairment [22], from which 44 percent of non-institutionalized persons aged 21-64 are employed, compared to 79 percent without a visual disability. An estimated 16 percent of non-institutionalized persons aged 21-64 with a visual impairment have a Bachelor's degree, 31 percent have some college experience or have earned an Associate's degree, and the remaining 53 percent either earned a high school diploma or GED or never finished high school [22]. According to the US Department of Education, National Center for Education Statistics, 2.9% of Bachelor's degree recipients in computer and information science fields have a vision impairment between 2015 and 2016 [23]. According to the National Science Foundation, National Center for Science and Engineering Statistics, Survey of Earned Doctorates, only 2.9% of all recipients of doctorates in the mathematics and computer sciences have visual impairments in 2018 [24]. In 2020, eleven states reported on the participation of students under the Individuals with Disabilities Education Act (IDEA) in computer science courses. Only 7.6% of students served under IDEA have taken a computer science course [25].

Programming and Software Development for the Visually Impaired

Programming challenges

There are a body of literature that examines the issue of programming for blind or visually impaired (BVI) persons, although limited research focuses specifically on understanding the challenges they face. Mealin and Murphy-Hill [26] interviewed eight blind or visually impaired professional software developers to learn more about their software development practices, what tools they use, including any assistive technologies, and their attitudes toward software development. The study revealed several findings regarding blind software development practices. For writing code, most participants preferred using simple text editors over integrated development environments (IDEs) due to the simplicity of the user interface (UI) and a shallower learning curve over the complex and inaccessible UI of the IDEs. Although some participants also used IDEs, they may use text editors in tandem for specific tasks such as keeping track of bugs, method headers, and variable names. Regarding assistive technologies, most participants use either a screen reader or a refreshable Braille display to interpret the code on the screen. Concerning using assistive technologies, participants indicated that code navigation and UML diagrams were challenging to use with screen readers. As code navigation is trivial for sighted developers as they can quickly jump to different codebase sections, it is challenging for BVI developers as screen readers navigate linearly, forcing them to read an entire document [26]. For UML diagrams, the participants that encountered them would try to find text-based alternatives or have their teammates verbalize them for better comprehension and interpretation.

Albusays and Ludi [13] surveyed 69 BVI developers to identify challenges in software development. Findings from their study were similar to that of Mealin and Murphy-Hill [26]. Participants used screen readers and refreshable Braille displays to read code and used text editors instead of or in conjunction with IDEs because they are complex and inaccessible. Even with assistive technology technologies, code navigation and UML diagrams were the most significant challenges. Additionally, like Mealin and

Murphy-Hill [26], participants used more basic debugging techniques as the debug features in IDEs were challenging to use.

In a follow-up study, Albusays, Ludi, and Huenerfauth [27] interviewed 18 BVI software developers with at least five years of professional software development experience. They observed them as they completed a series of short programming tasks to understand their profession's challenges and the workarounds they use to accomplish their tasks. Their observation revealed 12 challenges (see Table 2.1) they face. The three most prevalent challenges were debugging, line-by-line navigation, and distinguishing whitespace using a screen reader in indentation-based languages (e.g., Python). Findings of participants' use of software tools and assistive technologies were on par with their earlier survey results [13]. When probed about asking for sighted help, participants were split on the topic, with some listing the embarrassment of needing help to accomplish a task as a reason for avoiding help. In contrast, others voiced the necessity of seeking help from their sighted teammates for specific tasks such as getting an overview of the codebase. In Mealin and Murphy-Hill's study, the participants cite positive experiences and the necessity of seeking help when required in their line of work. Additionally, participants discussed potential future features to help make IDEs more accessible for their needs. Potential features include a hierarchical tree view of the codebase for more straightforward code navigation with a screen reader, auditory feedback, and collapsible, nested code environments in text editors.

Table 2.1: List of navigation difficulties and number of participants who mentioned them from Albusays, Ludi, and Huenerfauth, 2017 [27]

Navigation Difficulties	**Number of Participants**
Debugging: difficulty navigating through the code in the process of understanding a wrong output.	24
Line by line: difficulty navigating through code to locate specific information without having to go through the entire codebase linearly, line-by-line.	23
Indentation: unable to distinguish the level of whitespace using a screen reader in indentation-based languages, e.g. Python.	22
Nesting: difficulty navigating through nested methods, loops, functions, or classes.	20
Back Track: difficulty returning quickly to a specific line (in a lengthy codebase) when reviewing other code statements in various files.	18
Errors: difficulty quickly locating code errors while navigating through lengthy codebases.	14
Scope: difficulty understanding the scope level, e.g. while navigating deeply nested methods or loops.	14
Characters: difficulty perceiving certain characters, operators, and parentheses, e.g. missing some characters while coding.	10
Autocomplete: difficulty accessing the autocomplete feature due to incompatibility with the screen reader.	9
Relationship: unable to distinguish the relationship between code entities within a codebase, e.g. the relationship between a class and its subclasses.	9
Line Numbers: difficulty accessing line numbers in the code editor as they were not designed to be readable by a screen reader, e.g. using PyCharm with VoiceOver.	7
Elements: unable to quickly locate a specific element within a given array, class, function or loop, e.g. locating values or variables.	5

Despite the barriers, as research has shown, BVI programmers can be as proficient in software development as sighted programmers. To examine how blind programmers comprehend code compared to their sighted counterparts. Armaly, Rodeghero, and McMillan [28] conducted an eye-tracking study with blind programmers, similar in procedure to one conducted with sighted programmers [29]. Participants were tasked with reading several methods presented in a Notepad session and submitting summaries and comments about the methods, after which they were taken to an exit survey. An extension script captured the JAWS screen reader's eye-tracking data since all participants used JAWS as their primary screen reader. The eye-tracking data and summaries were compared with data from the sighted programmer study to see how each group prioritizes reading areas of code and significant differences in the quality of their summaries. The study revealed that although both groups have different reading code processes, their ability to comprehend the codebase is identical. To summarize, even with using screen readers to read code in a top-down approach, blind programmers can comprehend and provide quality feedback about code as their sighted counterparts.

Assistive Technology Solutions

Prior research into assistive technologies for blind or visually impaired programmers can be categorized into 1) programming languages and development environments, 2) development environment extensions, and 3) accessible learning environments. Some of the research can be associated with more than one of the groups.

Programming Languages and Development Environments

Prior research focused on developing specialized programming languages and programming environments for the use of BVI programmers. One of the earliest attempts at developing an accessible development environment was Schweikhardt, who developed a programming environment featuring a Braille-APL notation and an APL editor to support programmers using Braille displays [30]. Siegfried, Diakoniarakis, and Franqueiro created a scripting language for BVI programmers to develop Visual Basic forms without the need to use the "point and click" interface to assemble the graphical user interface (GUI) [31]. In another approach to address GUI programming's inaccessibility, Konecki created the Graphical User Interface Description Language (GUIDL), a description language and system serving as an aid for blind programming novices to design GUIs [32]. A user study revealed that blind novice programmers found GUIDL convenient and straightforward for building GUIs [33].

The research explored the application of auditory cues and interfaces as extensions to existing environments and as standalone development environments or languages. Sanchez and Aguayo developed the audio programming language (APL), which employs an auditory interface that programmers interact with using predefined commands [34]. APL was designed for novice blind programmers to learn concepts and algorithms in computer science to apply in problem-solving. Smith et al. developed JavaSpeak, a Java-based editor that provides audio feedback to programmers about program structure and semantics [35]. Sodbeans, a development environment based on the Netbeans IDE, provided auditory cues for Java programmers to improve their code navigation and comprehension skills [36]. A user study revealed that Sodbeans helped increase programming self-efficacy among blind programmers [37].

Development Environment Extensions

Research into assisting blind programmers in specific programming tasks, such as code navigation, has led to the development of assistive technologies as extensions of existing programming environments (e.g., Eclipse, Netbeans). The JavaSpeak prototype from Smith et al. [35] was developed into an Eclipse plugin to provide nonvisual navigation of the hierarchical tree-like structure of a program in the Eclipse IDE [38].

StructJumper is another Eclipse plugin, similar to JavaSpeak, designed to improve code navigation for blind programmers by creating a tree view of the code structure program, including methods and control flow statements, something JavaSpeak does not do [39]. Audiohighlight is another form of code navigation tool as an Eclipse plugin and a web service that utilizes HTML headings to represent code structure [40]. This strategy is familiar to blind programmers as screen readers use virtual cursors to navigate web pages through HTML headings and other structure tags. CodeTalk differs from the previous implementations as it is a plugin for Microsoft Visual Studio [41]. CodeTalk was designed to provide audio assistance addressing the main accessibility issues uncovered in their survey: discoverability (finding IDE features), glanceability (quick glance of the environment), navigability (code navigation/skimming), and alertability (error alerts, debugging information).

Accessible Learning Tools

Research into accessible learning of programming and computing principles has led to the development of several tools designed to be accessible for people with visual disabilities. These tools have come in the form of tangible interfaces, auditory interfaces, or a combination of touchscreen-centric and auditory features. Quorum is a text-based language that enables blind programmers to create complex programs, graphical applications, and games [42]. Initially designed for making text-based languages accessible, Quorum has now expanded to be a fully-featured tool for learning programming by anyone, regardless of disability. Block-based programming such as Scratch [43] and MIT's App Inventor [44] is a popular means of introducing computing to children due to its user-friendly interface and simplified syntax. However, block-based languages such as Blockly [45] are inaccessible to students with visual disabilities due to the visual capacity required to use them. Hence, research has focused on making accessible block-based languages for the visually impaired, such as Blocks4All [46, p. 4], which utilizes touchscreen interaction and provides audio feedback. PseudoBlocks allows programmers to use their keyboard to navigate block programs with speech aid [47]. StoryBlocks extends block-based programming by using 3D-printed tactile blocks on a specially designed workspace to teach students programming to create audio stories [48]. In addition to StoryBlocks, there are other tangible environments for teaching novice programmers with vision impairments. Microsoft's CodeJumper features a set of tangible blocks representing different programming concepts such as variables, conditionals, and iterations to teach children how to write a program [49], [50]. Quetzal and Tern are two tangible programming languages for operating robots [51]. The tangible blocks contain images representing actions to be executed by the robots. A camera connected to a computer reads in the image of the connected blocks and the software interprets the actions to execute based on how the blocks were assembled.

Computing Education for The Visually Impaired

There is growing research focusing on inspiring and teaching computing and programming to blind students. The exploratory study of Mealin and Murphy-Hill [26] brought to light an issue with blind developers not comfortable using IDEs. They theorized that the cause is a lack of education in their use. Existing research has explored blind students earning their degrees in computing, accommodations in their courses, and the need for curriculum changes [14], [52]. Baker et al. [14] explored the educational experience of blind developers who earned college degrees in computing. Among the challenges, the most significant being the lack of accessible learning materials, utilizing inaccessible IDEs, lack of context in teacher instruction, and a lack of support from faculty in their learning. Doustdar explained in his blog post that he continued his education in computer science outside of college due to the high visual demand of the coursework [52]. Prior work explored introducing blind students to computing through programming activities such as chatbots [53], using Twitter [54], robotics [55], and audio [34]. Stefik et al. established an educational infrastructure for teachers to incorporate accessible programming activities in their curriculum for blind students [36]. New initiatives have been created to help lead efforts in increasing the participation of people with disabilities in computer science. AccessComputing [56] provides resources for students with disabilities to successfully pursue college degrees in the computing fields and offers workshops and training services for institutions to help accommodate students with disabilities in their coursework. AccessCSforAll is an NSF-funded initiative designed to 1) build capacity for computer science teachers to better serve students with disabilities through professional development and training and 2) develop accessible tools and curriculum that can be used in the classroom.

Conclusion

This chapter identifies the current state of research in investigating and addressing the challenges blind or visually impaired people face in computing education. While the work described has addressed many facets of accessible computing education, there are gaps in the body of knowledge. Although prior work investigated blind programmers about the challenges in their line of work and their education, there is insufficient research examining their experience across all education levels (K12, secondary and post-secondary) and self-learning practices. There is insufficient research about teachers' teaching strategies in a computing course for students with visual impairments and their challenges in accommodating their needs. Finally, there is little research examining how blind or visually impaired students learn to code in the context of an actual class lecture. Therefore, the first phase of this research will consist of a multi-perspective investigation of how BVI students learn the skills and tools necessary to become developers, what barriers exist that impede their progress, and how such barriers can be addressed. The following chapters will detail the investigation into visually impaired students' learning challenges within computing education from different perspectives.

CHAPTER THREE

EXAMINING THE WORK AND EDUCATIONAL EXPERIENCE OF VISUALLY IMPAIRED PROGRAMMERS

Introduction

The first study of the investigation explores the workplace and learning experiences of blind or visually impaired (BVI) programmers. The goal of the study is to examine the following: 1) how professional programmers who identify as blind or have low vision adapt to their working environment, 2) what challenges they encounter because of their visual impairment, and 3) in what ways they feel their impairment gives them an advantage over sighted programmers. This study explores BVI programmers' prior education and training experiences to understand their methods to learn the skills for their positions. Earlier work explored the challenges of visually impaired programmers in the workplace. Findings include inaccessible developer tools, difficulties in navigating large codebases, and working with unreadable diagrams [26], [27], [57]. Few studies explored the prior education of blind or visually impaired programmers. The research that exists describes some of the challenges faced in post-secondary courses. Such challenges include inaccessible learning materials, a lack of context in the instruction during the lecture, and a lack of peer and faculty support within the department [14]. However, it can be argued that there are additional factors not sufficiently explored concerning the workplace environment of BVI programmers. While there is a body of research into addressing accessibility challenges for BVI programmers through tool enhancements and prototypes, little research focuses on the competencies of

BVI programmers compared to sighted programmers. Armaly et al. compared program comprehension strategies between blind and sighted developers to see how they would differ and if it would result in a significant disadvantage for blind developers [58]. The findings revealed that blind and sighted programmers use similar comprehension strategies and produce similar code summarization quality. The current research will shed light on barriers impacting BVI programmers' productivity and contributions and open more dialog on areas of improvement to provide better accessibility. This work may also spur conversations about improving the education and training for BVI programmers.

Method

I conducted semi-structured interviews to gather in-depth data about 1) the workplace practices of BVI programmers in software development environments, 2) their prior education and training experience, and 3) the advantages of being visually impaired. The following questions guide the current research:

RQ1: What are the challenges that blind or visually impaired programmers face in their place of work?

RQ2: What do blind or visually impaired programmers perceive as ways to improve accessibility in their place of work?

RQ3: To what degree does a visual impairment afford blind or visually impaired programmers an advantage over sighted programmers?

RQ4: How did blind or visually impaired programmers learn to code?

RQ5: What do blind or visually impaired programmers perceive as ways to make learning how to code more accessible?

Participant Recruitment and Description

I recruited participants through email advertisements sent to a mailing list for visually-impaired developers. The advertisements instructed all participants to contact me to schedule the interview. In total, 11 participants were interviewed. Participant demographics can be seen in Table 3.1. There were nine males and one female, with one person identifying as Other, ranging in age from 23 to 50 (mean: 33.7, std.d: 8.4). In terms of employment, seven participants worked full-time while one participant worked part-time. One participant is self-employed, and the remaining two are not currently employed. The average professional work experience is 10.5 years, and participants averaged 5.6 hours a day programming. One participant identified as having low vision, and ten participants identified as blind.

Table 3.1: Participant Demographics

PID	Age	Gender	Race/Ethnicity	Impairment	Position	Years of Experience	Country
P1	39	Male	Caucasian	Blindness	IT Security Specialist	15	USA
P2	23	Male	Asian	Blindness	Student	2	India
P3	34	Male	Caucasian	Blindness	Senior Software Engineer	10	USA
P4	50	Male	Black/Afr. Am.	Blindness	Senior Software Engineer	27	USA
P5	48	Male	Caucasian	Blindness	Senior Software Developer	21	USA
P6	27	Male	Caucasian	Blindness	Freelance Developer	6	Netherlands
P7	28	Male	Caucasian	Blindness	Python Software Developer	5	USA
P8	31	Male	Caucasian	Blindness	Software Engineer	10	USA
P9	26	Male	Latino/Hispanic	Low Vision	Graduate Research Assistant	5	USA
P10	30	Female	Caucasian	Blindness	IT Specialist	5	USA
P11	35	Male	Caucasian	Blindness	Applied Scientist	10	USA

Interview Methodology

I conducted semi-structured interviews, lasting between 25-40 minutes (see Appendix A for interview questions). The interview began with the reading of the informed consent document, during which the participants could ask questions and provide verbal consent to participate. Questions focused on the following critical aspects: 1) navigating the software development workplace as a visually impaired programmer, 2) improving accessibility in the work environment, 3) the learning experience in programming as a visually impaired student, and 4) improving accessibility in computing education. Participants were compensated with a $10 gift card after the research.

Data Transcription and Analysis

A professional transcriptionist transcribed all recorded interviews in preparation for analysis. The research team entered all transcripts into MAXQDA [59], a qualitative data analysis computer program. I used a hybrid thematic analysis, a method used for extracting common themes from textual data. After becoming familiar with the data, two external researchers independently coded ten percent of the transcripts with a set of codes agreed upon in advance. The two researchers then met to compare their transcript sets to form a list of codes agreed upon to use for the remainder of the transcripts. When the coding was complete, I reconciled both sets of transcripts to form a unified version. Any disagreements in coding and categories were settled by me and collectively agreed upon by the team.

Findings

This section presents findings from the interviews, separated into four areas: software development challenges, areas of improvement, prior education, and advantages of visual impairment.

Software Development Challenges

While the participants reported many challenges during the interviews, only the barriers participants perceived as the most significant are reported. The most notable challenges include workplace dynamics, the accessibility of software tools, and specific programming tasks.

Workplace dynamics. All but three of the participants work as part of a team at their place of employment. P2 and P5 work independently (P2 is a college student, P5 is a freelance developer working alone), and P10 is unemployed. Common amongst the discussions was the type of interactions with the coworkers. P1, P3, and P5 explain how they would have their sighted coworkers handle the less accessible and more complicated tasks such as graphical user interfaces or frontend web design:

> *"So, I'll take on some of the more accessible uhm sometimes harder tasks and uh they'll take on some of the uh less accessible and easier tasks." (P1)*

P6 has difficulty when it comes to using team-based collaboration tools such as Trello. As team members continue to fill the task board, it takes more time to read the entire board, which slows down his progress in other areas. P6 mentions how his visual impairment can make certain collaborative activities like pair programming a challenge:

> *"But, you know, like I can't look over someone's shoulder and see what they're doing and... They can describe it to me and.... Generally, in those settings sometimes we have to take it into a meeting room and figure something out together and it's a little hard to follow along..."' (P6)*

P4 and P10 recall experiences of team settings where coworkers often have trouble determining how to send visual information in a readable format by a screen reader. For example, P10 would get an image of code from a coworker, but her screen reader could not read it.

Accessibility of software tools. The accessibility of the tools used in the BVI programmer's line of work can be a critical factor in successfully contributing to their team's objectives. However, it is also one of the most significant barriers. Participants found that IDEs were too complicated and geared more towards visual functionality.

> *"Visual Studio is quite challenging at times because, okay, there's so many things you do in it and it's such a large uh development environment and so many different developers have, or I guess so many different teams in Microsoft have developed into the finished product of Visual Studio that the accessibility is not consistent." (P4)*

As a result, many participants turn to use text editors with simpler user interface (UI) layouts as a replacement or in conjunction with IDEs to overcome some accessibility barriers. For example, P9 likes the Python programming language because it can be written and run using the command line rather than an IDE.

Programming Tasks. Challenging tasks fall into two categories: 1) code navigation and 2) user interface.

Participants brought up the difficulty in navigating through a codebase as their screen readers move primarily in a linear manner. P2 describes the task as time-consuming:

> *"It's difficult to jump from let's say if the code is nested and it's time-consuming to go line by line to some uh specific syntax, so it is time-taking" (P2)*

Other participants (P6, P10, P11) indicated the advantage of sighted programmers moving through the code quicker and having less information to memorize than BVI programmers.

Debugging and code navigation were discussed in parallel with aspects of debugging involving navigating to the error.

> *"debugging and how to navigate to the different parts of. Other people can bookmark some places uh in the syntax so that you can revisit or can make some changes or whatever. But as a visually impaired, I feel it's important, but uh I don't figure out any way to uh to do such thing." (P2)*

Working UI development was a topic discussed by three participants (P1, P4, P5) as something they typically avoid doing because they consider it a highly inaccessible task:

> *"I would say that is interface design. Uhm Obviously, when you are building a program, an application or website, it has to look a certain way and that when, for example I do a lot of websites work, uh it generally comes down to uhm CSS, which is just not something I have the capability of doing up to the same level of my sighted peers." (P5)*

Participants mentioned often leaving the UI work to their coworkers while focusing on the program's logic.

Areas of Improvements

Suggested areas of improvement are categorized into 1) accessibility of tools and 2) quality of life improvements.

Accessibility of tools. Most participants (P1, P2, P3, P4, P6, P7, P9, P11) suggested improvements to existing IDEs for better accessibility. One such suggestion was the simplification of the user interface. Participants did not like the number of controls and added complexity to the interface, hence why many developers would use simple text editors or the command line:

> *"It's a lot of hassle and difficulty for me, so I just want a basic interface to / important feature which people use most of the time." (P11)*

P1 suggested a way to obtain a larger view of the codebase within the IDE for improving his ability to understand the code structure:

> *"Some method sort of getting a bird's eye-view of a large code base. Uhm And being able to get a better sense of uh the organization of it" (P1)*

Quality of life improvements. Regarding the quality of life (QoL), P1 suggested video tutorials for people with visual impairments for using a specific tool:

> *"Sort of there's not like the equivalent of the guy with the YouTube video uhm showing ow you would use a program uh from a non-screen-reader perspective." (P1)*

P7 and P10 also made suggestions for documentation and built-in help functionality for learning the shortcuts and commands to use tools such as IDEs.

> *"Documentation in the first place. Like uh providing uh the shortcuts and the hotkeys for the different commands in different areas together with uh the main instructions..." (P7)*

P6 mentioned the challenge of introducing a new tool and the additional cognitive effort required by BVI programmers to learn it. Compared to sighted programmers, it may take longer for BVI programmers to learn a tool:

> *"Every new tool you introduce, like, it slows you down, especially when you're using a screen reader compared to if you're sighted. I mean, you have to learn it. For anyone. You have to spend time learning. But every tool kinda has its own keystrokes and you have to make different like adjustments for the screen reader. And it gets to be a lot of mental overhead." (P6)*

Participant P5 expressed concern about the current attempts to improve accessibility. His concern was that some companies develop accessible technology solutions without understanding BVI programmers' needs and preferences.

> *"That people try to make things accessible without doing research first. And it's just way too overdone and actually worsen the situation more than it helps, which is uh not good either. So yes, definitely input for and from visual disability uhm, I guess, suffering people should definitely be included when these designs are made uhm if that's possible, yes." (P5)*

P5's concerns are shared amongst the disability community in that companies attempt to create assistive technologies without including input from the end-users. As more BVI programmers enter the software engineering workforce, more consideration in the universal design of software tools with input from the population will be needed to create a more inclusive work environment.

Prior Education

Motivation for pursuing computing. Motivations for pursuing a career in computing arose from interest or early attempts at learning to code during childhood. At least 6 of the participants mentioned their interests manifested through early exposure to computers, playing games, or coding.

> *"I always toyed around with the computer when I was ten, eleven, twelve. Started getting into programming, made the mistake of trying to teach myself C when I was about thirteen. Uhm I was able to get up to pointers, but pointers didn't make sense to me at all. The problem-solving is what got me into it really, the problem-solving aspects, I would say." (P8)*

P3 and P10 revealed pursuing computing because of the surplus of job opportunities in the market and the salary of programmer positions.

Methods of learning. Nine of the participants went to a college or university for a degree in either Computer Science (P1, P3, P6, P7, P8, P9, P10), Informatics (P5), or Information Systems (P4). The languages they learned varied; however, some of the more common languages and markup among the participants include HTML, Python, Java, and C++. In terms of developer tools, most of the participants used Visual Studio and simple editors such as Notepad++.

Challenges. The participants' challenges can be grouped into three areas: learning materials, developer tools, and in-class instruction.

Nine of the participants (P1, P2, P3, P4, P6, P7, P8, P10, P11) mentioned struggling with the lack of accessible materials and having difficulty with the materials' visual elements used in the class. Participants mentioned trouble using the course textbooks, mainly when they contained graphics or math formulas that their screen readers could not detect.

> *"When I got into the college classes, specifically like the algorithm and data-structures class, some of that stuff was displayed visually, which still more complicated. And then, being able to visualize things, like Big O notation and how O(N) versus O(N^2) is actually a huge difference, was somewhat difficult." (P3)*

The tools used in their courses also played a role in their challenges in learning how to code. Participants P2, P7, P10, and P11 struggled with the IDEs that were inaccessible at the time of their learning. Such tools may be more accessible now than at the time when participants were using them.

> *"Well, at first it was nasty, as I told you, because they taught us C++ and C# and such stuff, and in order to use it you had to use Visual Studio at the time. That was the standard and it wasn't accessible all the time…" (P7)*

Participants also mentioned their professor's tendency not to provide context to what they are teaching from the materials. P4, P9, and P10 cite how professors will reference a graphic or concept from a textbook, PowerPoint presentation, or something on a whiteboard but fail to describe the content in enough detail to understand what they are referencing. For example, a teacher may allude to an image on a slide but not describe the image, as in the case for P4:

> *"But sometimes they actually say, "Well, look at this picture", you know, and they'll word their phrases inside their paragraphs which will not give you insight about what is actually in that picture or image."*
> *(P4)*

P10 shares how instructors will post code on a PowerPoint slide and walk through it but fail to verbalize each line's structure and location, making it near impossible for blind or visually impaired students to follow the code. It demonstrates how teachers may lack the experience or knowledge of providing instruction to visually impaired students.

Workarounds. In overcoming their challenges, most participants looked to resources outside of the classroom, such as online websites with tutorials and electronic versions of their textbooks. Participants P2, P4, P5, P6, P8, and P11 looked for online materials such as web-based tutorials or articles explaining their concepts in class. Some participants leveraged the assistance of other people or services for learning the more challenging topics or finding additional resources. P1 gets assistance from online communities, from the general population of program-l for blind or visually impaired programmers to more specific communities such as PostScript for database development. P3 had friends who learned the materials from his course and helped to explain the concepts to him. P6 got help from his professor and accommodations from the disabilities office.

Improvement to CS Education. Two themes were common from responses regarding improving computing education and programming concepts for visually impaired persons: accessible learning materials and accessible learning tools. Every participant mentioned the necessity of materials to be accessible by screen readers or easily translated into an accessible format. Materials included printed textbooks with electronic variants, images and graphics with descriptive texts, and online documentation.

> *"I think that if every image, every picture, every graph in the text is actually provided in a way that is consumable by someone who is visually impaired. That's a major win." (P4)*

> *"well-formatted HTML with uhm, you know, headings and tables and (2) lists and such. Or uhm yeah, HTML or EPUB. So, some form of / of well-formatted, well-structured uh electronic text." (P1)*

Some participants (P1, P5) suggested curricula that should provide materials in a wide variety of formats that people with disabilities can access. Additionally, the shift towards instruction using videos may not be helpful if the video lacks the explicit context of instruction when demonstrating code.

> *"And another thing that I see a lot in these video-based courses is: a lot use of demonstratives. So, you'll get phrases like "If I take this here and paste it down here, then that area will come up. And if you do this, then it's fixed," and of course by this point a blind person has no idea what's even happening anymore. So, these are things that content creators should definitely keep an eye on. If they don't, they're gonna to exclude a lot of people. And this is actually sort of a parallel to what teachers used to do in a classroom. They'd point at the whiteboard and pretty much do the exact same thing, the big difference being that it's a live situation, so you can raise your hand and be like, "I don't know what's happening. What are you pointing at?" With a video this is just not the case. You'd have to comment and hope that someone gets back to you, which is not a very uh proper way to do teaching, I would say." (P5)*

P10 would like to see accessible materials available before a lecture rather than after. She found it difficult to follow the lectures without the materials, but the professor would wait until after the class to make accessible versions to distribute. P11 explains that instructors should plan for accessible learning materials from the start rather than producing them as needed.

The accessibility of developer tools has always been a concern for blind or visually impaired programmers, and the issue is still prevalent for students learning to code. P5 mentioned accessibility challenges using Codeacademy, a website offering courses in programming in various languages [60].

> *"...the Codecademy comes to mind, that use inaccessible uhm widgets for their code editor for example, which makes it hard for uh blind people to actually use the free services that they offer. This is being worked on, but this shouldn't even have happened in the first place."*

P6 would prefer courses on programming to use simple editors such as Notepad++, which are easier to navigate with a screen reader than the more complex UI of IDEs. P11 would prefer to use the command-line interface (CLI) to learn how to code. In P5's opinion, developer tools should be designed to be accessible initially instead of trying to 'fix' them after their deployment.

> *"Currently accessibility is very much an afterthought with a lot of different, like, tools. Pretty much all of them really. There has to be a shift from like fixing this afterwards to fixing this when it's actually being created."*

This finding may suggest that developers of editors should consider all potential users of their product, including those with a disability. To ensure accessibility of editors on the launch, companies may consider an inclusive design approach, involving all potential users' input into such products' design.

Advantages of Visual Impairment

I asked participants about perceived advantages their visual impairment may give them over sighted programmers. The most common perceived advantages mentioned were 1) better memorization and understanding of the code structure, 2) greater focus on the details. Participants P3, P5, P6, P7, P8, and P10 believe that because they cannot rely on sight to reference code from previous lines, they build a mental model of the code structure. As a result, they develop a better understanding of the codebase they are working on than their coworkers:

> *"In that sense, yeah. I can retain vast quantities of information in my head uh about the code structure and what it's doing and some of the actual names of functions and so on." (P8)*

P4, P6, and P11 mentioned being more focused on the task than their coworkers. P11 believed his lack of sight helped to remove distractions that may affect sighted developers because of the additional sensory input:

> *"Yes, I mean the first thing is as a blind person I get more focused, so not distracted with a lot visual things. Because uh vision gets a lot of input and it's easy to get distracted. So, first it's focus" (P11)*

P5 pointed out that memorization extends to understanding the code and the many keystrokes and shortcut keys to perform his job. He emphasizes that keyboard shortcuts may be an advantage to the use of a mouse depending on the task:

> *"A lot of the sighted developers I work with don't know as many of the keystrokes that I do because I literally need them to do my job and they*

can use the mouse. And it can be argued that a mouse is faster. It will depend on the task you're doing, I think." (P5)

P4 expresses how his work habits and attention to detail result from his impairment and that it gives him a "one-up in many situations" because he can "focus more on the task at hand." P1's sentiments mirror P4, in which, because of his lack of sight, he can focus more on understanding the code and pick up on the details typically missed by his coworkers.

"I excel at looking at a small uhm area with a lot of detail." (P1)

From the responses, the participants feel that their lack of sight removes the distractions of visual stimuli that can impact sighted developers' productivity and, therefore, allow them to focus more on the code and better comprehend its structure. Research shows that blind or visually impaired programmers can perform at similar levels in program comprehension and summarization as sighted programmers, despite the differences in their reading strategies [58].

Conclusion

The findings from the interviews helped to answer each of our research questions. For RQ1, the main barriers encountered in the workplace are 1) workplace dynamics, 2) the inaccessibility of specific software tools, and 3) specific programming tasks too challenging for BVI programmers. For RQ2, the most significant suggestions involve 1) simplifying the user interface for IDEs, 2) quality-of-life improvements such as supporting documentation for using specific tools, and 3) including input from blind or visually end-users in the design of accessible technology solutions within software tools. Responses for RQ3 suggest that programmers can memorize and understand their codebase and keystrokes and commands. Additionally, blind or visually impaired programmers are more focused, have greater concentration, and pay more attention to detail than their sighted counterparts without much or any visual stimuli to distract them. For RQ4, many participants were introduced to programming during childhood and then went to college to earn a degree in a computing-related field. Every participant encountered challenges learning how to code, such as finding accessible learning materials, working through inaccessible editors, and instructors' inability to adapt their instruction for them. For RQ5, The most common suggestions include more accessible learning materials, improving accessibility in using code editors, and better accommodations for instructors' lectures.

In conclusion, this study identified factors that hinder blind or visually impaired programmers' performance in technical challenges or workplace practices. Programmers suggest that more usable and accessible software tools and offerings in training and documentation for using them would enable them to contribute equally compared to their sighted coworkers. From an educational perspective, current educational materials and practices are inadequate for students with visual impairments and negatively impact their learning. These findings suggest that current curricula should be reviewed and modified to offer flexible formats of the same learning materials that accommodate students with or without a disability. Lastly, future research should consider how the advantages of visual impairment may be explored to influence the design of development tools for blind or visually impaired programmers. In the next chapter, the research explores the state of computing education from the view of teachers of visually impaired students.

CHAPTER FOUR

EXPLORING THE PERSPECTIVES OF TEACHERS OF THE VISUALLY IMPAIRED REGARDING ACCESSIBLE K12 COMPUTING EDUCATION

Introduction

In Chapter 3, I interviewed 11 blind or visually impaired (BVI) programmers regarding how their visual impairment impacts their work in software development, the biggest challenges in their line of work, and their past education and training experiences. Findings from the interviews revealed that BVI programmers face obstacles in the form of a) workplace dynamics, b) software development tools that are inaccessible, c) primarily visual tasks they often delegate to their coworkers, and d) limited training or documentation for using tools. Regarding their past education, similarly, all participants reported having to find external resources to supplement their classroom instruction as the learning materials were not very accessible. Further exacerbating the challenge, participants described how teachers failed to contextualize their teaching when pointing to concepts written on a board or presented on a slideshow. These findings are consistent with past work examining the challenges faced by BVI programmers in using current developer tools and performing specific programming tasks [26], [27], [57]. The findings related to BVI programmers' education are consistent with recent work exploring obstacles in BVI students' educational attainment [14].

The study in Chapter 3 serves as the starting point of a more in-depth exploration into BVI persons' growing challenge in becoming programmers. A common issue from the first study and prior literature is BVI persons' education in computing and how current pedagogies, curricula, and materials may significantly disadvantage them compared to sighted students. This second study attempts to get to the root of the problem by exploring the current state of computing education (CEd) from the educators who instruct blind or visually impaired students. Such educators are known as teachers of the visually impaired (TVI) [61]. TVIs play a crucial factor in the success or failure of BVI students' academic outcomes. TVIs must discover, create, or adopt a computing curriculum that will work given their students' ability, provide materials in an accessible format, and use learning tools to work with assistive technologies. Prior work examined how factors such as environment, closeness to mainstream technology, cost, and the reinforcement of disability identity impact what technologies TVIs use in their classroom [61]. This study explores CEd more broadly to include topics involving the curriculum used, the availability and accessibility of learning materials, strategies used to create an accessible learning environment, and challenges for both the students and the teachers that need to be addressed.

Method

Semi-structured interviews were conducted to gather in-depth data from K12 computer science (CS) teachers regarding their teaching programming approach to their students and the challenges inherent in their role. The questions asked focused on lecture preparation, programming languages and tool use, and the areas in which they believe blind or visually impaired students excel and struggle. The findings from the study help to answer the following research questions:

RQ1: What are the challenges faced by teachers in instructing blind or visually impaired students to write code?

RQ2: What challenges do teachers see in blind or visually impaired students learning to write code?

RQ3: How do teachers adapt their curriculum and teaching style to serve the needs of blind or visually impaired students learning to write code?

Participant Recruitment and Demographics

I recruited participants using snowball sampling. Advertisements were sent out via email to individuals and an organization focusing on making K12 computing education accessible for people with disabilities. From the initial advertisements, they were sent to people interested in participating. Interested participants reached out to me, after which I scheduled interviews within one week. Seven (four males and three females) participants were interviewed. All participants taught at least one computing class with one or more students with a visual impairment. Teacher ages range from 30 to 65 years, with an average age of 50. Four of the participants teach at a school for the blind or visually impaired, while two teach for a public school system. Table 4.1 shows the demographic summary of the participants.

Table 4.1: Participant demographics

PID	Age	Gender	Race	Grade Levels Taught	Years of Teaching Experience
1	53	Female	White	8th, 9th, 10th, 11th, 12th	15
2	65	Male	White	2nd, 3rd, 4th, 5th, 6th, 7th, 8th, 9th, 10th, 11th, 12th	24
3	64	Male	White	9th, 10th, 11th, 12th	10
4	30	Female	White	9th, 10th, 11th, 12th	1
5	38	Female	White	6th, 7th, 8th, 9th, 10th, 11th, 12th	12
6	57	Male	Asian	3rd, 5th, 6th, 7th, 8th, 9th, 10th, 11th, 12th	5
7	43	Male	White	5th, 6th, 7th, 8th, 9th, 10th, 11th, 12th	5

Procedure

Semi-structured interviews were conducted, lasting between 20-60 minutes. Participants were emailed the informed consent document before the interview and allowed to ask the research team questions before giving their consent to the study. Participants also completed a short demographic questionnaire. Questions from the interview can be found in Appendix B. Participants were compensated with a $10 prepaid gift card.

Data Transcription and Analysis

A professional transcriptionist transcribed all recorded interviews in preparation for analysis. The transcripts were analyzed using MAXQDA [59], a computer program for qualitative data analysis. The data was analyzed through thematic analysis, a method used for extracting common themes from textual data. After becoming familiar with the data, two external researchers independently coded ten percent of the transcripts with a set of codes agreed upon in advance. The two researchers then met to compare their transcript sets to form a list of codes decided upon to use for the remainder of the transcripts. When the coding was complete, I reconciled both groups of transcripts to form a unified version. Any disagreements in coding and categorization were settled by me and collectively agreed upon by the team.

Findings

Interviews were conducted with seven TVIs to understand their experience better preparing for and instructing blind or visually impaired students in computing courses. I categorize our findings into seven broad themes.

Curriculum and Topics.

In addition to computing principles, participants also taught courses in using a computer (P1, P4), assistive technologies such as Braille readers and specialized keyboards (P1), and math (P3, P4, P5). Curriculum use varied among three different approaches. P1 uses the curriculum from CodeHS [62], which provides lessons for teaching students computing concepts and programming such as JavaScript and Python. P2, P6, and P7 use the Quorum programming language [8], an evidence-based language accessible for people with disabilities. P3, P4, P5 use the AP Computer Science curriculum for their course [3]. Of note, P6 does not teach a class to students but instead provides supplementary materials and instruction to students who desire to pursue a computing career. It is a constraint of his position as a special education teacher for a public school system.

P1 teaches a broad range of topics, including programming, computer applications (i.e., Microsoft office), and web navigation, as his course focuses more on general computer technology use. P2 follows a similar approach to P1, but by incorporating computing concepts and problem-solving more broadly within the sciences and mathematics. P3 follows the curriculum path of the AP CS course but admitted that the programming aspect of the course was challenging to do because of the switch to distance learning:

> *"I want to say we did chapters 1, 2, and 4 on the AP computer science curriculum. So, chapter 3 was programming, but we were remote and I said, "Oh, this is gonna be too complicated to try to do remote", so I skipped over it."*

P4, P5, P6, P7 followed the topics provided by their respected curricula.

Languages, Developer Tools, and Assistive Technologies

While there were variances in what language participants used to teach, all participants at one point in time have used Quorum, citing its accessibility for their blind or low vision students.

> *"the thing that I liked about Quorum was that it was built for a community of uhm students that are blind. And so, they did a lot of things to make coding more accessible for them" (P3)*

Aside from Quorum, other languages or markup used included HTML (P1, P2), Swift (P3), Java (P1, P5, P6), and Python (P1, P4, P6).

Participants using Quorum utilized either Sodbeans or the Quorum website's online editor for software tools. Sodbeans is an integrated development environment (IDE) based on the Netbeans IDE, which implemented audio interfaces to increase BVI programmers' accessibility [37]. However, the Quorum team released Quorum Studio [63] to replace Sodbeans as the desktop-based IDE. P3 uses Apple Swift Playgrounds for teaching robot programming using the Sphero Ball [64]. Other editors used include Notepad, WordPad, TeachText. Most participants prefer online editors as the curriculum of choice provides their editors for students to use rather than downloading and installing them on computers. Teachers also like it because they are more compatible with screen readers.

In terms of assistive technologies, participants reported students using screen readers such as VoiceOver (Mac), JAWS, or NVDA; Braille displays, tactile maps, and magnification. P2 noted that most of his low-vision students used magnification, while blind students used screen readers and corded keyboards.

Lecture Presentations and Accommodations

Participants indicated how they adapted and modified existing curricula to fit their students' needs when asked about their lecture preparation and presentation. P1's process involves role-play; she goes over the lessons as a BVI student and identifies what materials should be modified, discarded, or left as-is:

> *"...I go in and imagine that I am blind and, of course, so everything has to be done via screen reader and keyboard shortcuts... So, I'm always looking for simplicity. I go in and do the exercises, do the programming exercises as if I were a blind person, to see what problems I would run into. And then I wouldn't allow the stuff that just doesn't work. And uh that's how I've developed my curriculum"*

P2 adjusts materials for low vision by using "illustration panels" since they have some usable vision while, for the blind students, he converts the materials to Braille. P2 also provides instructional guides for his students, which provides the most used keyboard commands or snippets of code to use as a reference. In a similar approach to P1, P3 goes through the AP Computer Science curriculum and removes lessons that he considers very hands-on and would be too complicated for his students. P3 plays videos from the curriculum and supplements them with homework assignments. P5 uses the online curriculum on Quorum, and, therefore, all the materials are online as her classes are blended (both in-person and distance learning). To help students with code navigation issues, P5 has students commenting throughout the code to indicate the start and endpoints of functions and code blocks. Additionally, students may use a separate file to keep track of their code locations:

> *"Uhm I've also had a couple of students who would keep a separate file open where they'll put uhm like the name of whatever block of code it was and then the line numbers that they intend it to be on, so that they can go back and kind of check that little cheat sheet of "Oh, I'm looking for", you know, "print screen" or whatever they've made and they can see what line they should be able to hop to find it." (P5)*

P6 follows the curriculum from Quorum but uses Google Docs as a medium for his students to take notes and complete homework assignments. P7 focuses on lecture materials using audio programming as it is the most exciting aspect for his low vision and blind students. P7 recalls a time doing robot programming with his students, which the low vision students like; however, the blind students found it less attractive because they are not able to see the robot's actions:

> *"I've also found that with totally blind students that, like, with the robot stuff sometimes isn't as engaging either. Because it's basically, you know, you're programming a robot to do what you want and they can't SEE the robot doing that, you know?"*

Other participants mentioned taking different approaches in teaching students with low vision compared to students who are blind, specifically when using different mediums to present materials.

Learning Competencies and Challenges

We asked TVIs about topics they find their students understand well and what tasks they can perform with little difficulty. The participants' most common responses were understanding the use of conditionals (if, if-else statements), repetition (loops), logic statements, and variables. In P1's course, she expressed how students understood discrete points of information reasonably well:

> *"...the main thing I found is that if there are discrete points of information. Having a table database. Uh uhm And I really haven't taught SQL, but that concept of tables seems to be very uh - how shall we say? the blind seem to uh understand that very well because of the discrete points that the table describes itself."*

In P2's class, his students can comprehend and perform robot programming, given how he teaches it in a linear style.

We asked participants about topics or tasks they found their students to struggle with the most. P1, P2, and P7 noted visual elements such as graphics on a page as the biggest challenge for their students. P3 emphasizes that writing code to be more difficult than block programming, noting the nuances of the syntax that programmers must abide by for each language and how it may be problematic for people with low vision or blindness.

> *"I find actually writing the code to be much more difficult than to do the block coding, especially for students with low vision...Uhm Everything just takes for my students uhm a little bit longer. So, if I start writing the code, it's gonna take so long... Like, you don't capitalize the first letter but you capitalize the second word, and open parentheses or close parentheses."*

P1 and P5 also identified challenges in code navigation, code placement, and debugging problems for students. P5 expressed that although students may know the programming concepts, such as conditionals, they struggle with placing code within the codebase's proper scope. P5 feels current development tools do not perform well in indicating the positioning code constructs (e.g., correct placement of curly braces to indicate the be code block).

Concerns for BVI Students

We questioned participants regarding concerns about the current state of computing education for BVI students and how it may affect their interest in pursuing a career in computing. The biggest problem pointed out was the lack of accessible learning materials available. Participants argued that BVI students are at a disadvantage to their sighted peers because the current materials are not as accessible and require more time to read over and comprehend. All participants were consistent in pointing out how many of the curricula for teaching how to code do not provide sufficient materials that meet their students' needs.

> *"Yeah, the curriculum, I mean, I keep going back to the curriculum, but that's the main thing, you know. A guy like myself, I don't have time to develop a curriculum for the blind." (P1)*

P2 expresses concern about students not learning the 'precursor' skills to learn to code. He noticed that students learn computer skills at different times, which may be problematic if they intend to learn programming with insufficient background knowledge in using a computer. P6 suggests that curricula should incorporate pre-learning materials to prepare students for the lecture as BVI students require more time to process information. P6 and P5 believe that in traditional classrooms with sighted and BVI students, the difference between the group's information processing may hurt the BVI student's confidence, efficacy, and potential interest in computing.

> *"So, they get behind and behind, and that's where starts and then, if they had enough, you know created, the students, the younger students would just lose their interest in learning and then just drop out. And that has been happening a lot of times in the education system."*

In discussing the challenges that students face in using the software tools, navigating the user interface was the biggest concern among all the participants. Participants noted that screen readers would have difficulty picking some aspects of the interface depending on the interface and may miss menus or other controls. P1 suggests that UIs should be more simplistic or provide a small set of keyboard shortcuts that students can remember.

> *"...sometimes it takes as much just to teach an IDE as it does to teach the actual content or the language. Uhm Just to be able to navigate and*

> *use the interface. Uhm uh memorization of tons of keyboard shortcuts and so on and so forth. Simplification is the key."*

Participants find that their students have difficulty navigating the screen even with web-based editors because the user interface design is not optimized for screen readers.

Suggestions for Improving Accessibility in K12 Computing

It was important to know from a teacher's perspective what improvement could be made for computing education accessibility for the student and the teacher. One area was the development tools used with the curriculum for their class. P1, P2, P5, and P7 mentioned improved UI navigation for the tools. They witness their students struggling to navigate between menus and find the desired window. P2 and P6 similarly express concern about the lack of consistency with how specific screen readers process and navigate IDEs.

> *"Each screen reader, reading program, accesses the computer system and the code behind it somewhat differently. And so, for the normal developer who develop a program totally accessible for all of them is impossible." (P6)*

P6 believes developers of IDEs should incorporate accessibility testing of their platforms using a screen reader to check for potential issues when BVI persons use them. However, he also recognizes the increased workload for the developers and believes alternative solutions should be explored.

> *"So, particularly, you know, in an ideal world I wish that every program developer uses the screen reader to program the software uh and then test it out with a screen reader. But that's a huge, you know, man-hour. That kind of a work would double or triple the developer and programmer's time to produce one good visual program."*

P3 suggests physically accessible supplements to help BVI students better use the tools. For example, she recalled using tactile maps with Swift Playgrounds to help students recognize specific screen areas to use the interface.

The second area of consideration is the availability of resources for the teachers to supplement the instruction. P2 and P7, who work at a public school, mentioned the difficulty of providing their resources and equipment for the computing courses they teach. They would suggest that schools incorporating computing in their curriculum should invest additional resources (e.g., computers, software licenses, mobile devices). P3 suggests lesson plans that provide explicit instructions for teaching the concepts of the lecture. He feels that curricula do not provide enough information to carry out instructions to the class and present the topics effectively to students. P5, on the other hand, believes that languages like Quorum have support for teachers to learn how to teach students to use the editor with a screen reader, citing several schools that have issued videos and tutorials on such topics. P6 vouches for pre-learning materials to which BVI students should gain access in advance of upcoming lessons. Such materials would help BVI students stay on par in preparation with sighted students.

Discussion

This study examined BVI students' experiences learning to code through the lens of teachers of the visually impaired. The interviews uncovered the work required to adjust existing curricula to accommodate their students' needs (RQ3). Teachers find the biggest obstacle for both the students and themselves to lack accessible learning materials, which falls on the teachers to produce usable and accessible computing education (RQ1, RQ2). This study also uncovers more about topics and tasks in which BVI students tend to perform well and the advantages their visual impairment offers compared to sighted and low vision students compared to blind students. The following sections offer areas that educators and researchers should investigate and improve to remove these obstacles.

Accessible Course Materials and Curricula

Based on participants' responses, a significant barrier in computing education for the blind and visually impaired is course materials that are not accessible. It is critical to ensure equitable access to learning materials and that teachers are equipped to administer such materials to their classes to provide a balanced learning environment for students of all dis/abilities. The pursuit of such materials should not rest entirely on the teachers to provide them as they may not have the expertise to design them and increase their workload for delivering accessible instruction for their classes. The development of fully accessible computing curriculum materials requires researchers, industry partners, software developers, educators, and specialists on disabilities to pool their resources and produce artifacts that are both usable and accessible for student learners [14]. There is also a need to review current computing curricula and consider how specific aspects of the lessons place blind or low-vision students at a disadvantage. Such an effort would require significant time, human resources, and resources generally, which, depending on the organization, may not be readily available. The development team behind the Quorum language and Quorum Studio IDE adapted the AP Computer Science Principles curriculum from Code.org. The lessons have been tailored to be accessible for people with disabilities and to use Quorum [63]. This work can potentially serve as a model for other curricula to adapt and extend their own for improving the accessibility of their lessons. There is a need for additional research into barriers in developing new accessible curriculum and materials and designing and deploying more cost-effective solutions to reduce roadblocks for BVI students.

More Resources for Teachers

Teachers of the visually impaired in K12 education need more support in their computing class. Such support may come in the way of up-to-date computers and devices, software licenses, assistive technologies for their students, or supplemental learning materials. Two participants mentioned facing challenges in the classes related to outdated or missing equipment. Institutions dedicated to providing computing as a course should invest more of their budget in ensuring teachers are equipped with the necessary technology to enable a better teaching experience and learning experience for the students. The challenge in such an investment may be affected by factors such as budget, student interest in computing, and the teachers trained or willing to teach computing.

Development Tool Accessibility

A critical barrier for BVI students in learning computing is the lack of accessible development tools. IDEs are too complicated for BVI people to utilize fully, and most editors do not offer more accessible methods of navigating codebases. Even with tools such as StructJumper [39], AudioHighlight [40], and CodeTalk [41] that help in improving code navigation in IDEs, such tools are limited to a few platforms (Eclipse, Visual Studio). Many other programs have not largely been addressed in becoming more accessible. To this date, Quorum Studio is the first and only evidence-based program developed to be accessible for people with disabilities. However, some teachers have noted students encountering trouble navigating the interface. Additional research is needed to explore further ways to enhance accessibility in existing tools and develop new tools that are usable for people of all dis/abilities.

Conclusion

This study examined teachers' teaching experiences of visually impaired students in computing classes, the challenges that come with their profession, and what they see as barriers for BVI students to learn how to code. The study revealed how teachers often must modify existing curricula and materials due to their lack of accessibility, increasing their workload. Although teachers mention their students' competencies in specific topics and perform specific programming tasks, their learning progression is behind sighted students due to problems using current development tools that are not readily accessible, even with assistive technologies. We also see issues specific to blind students vs. those with low vision and how it may negatively impact their interest in CS. Teachers offer suggestions on improving computing education for teachers and students, including more learning resources for teachers and better, more accessible equipment. The next chapter presents and discusses a diary study where data is collected from high school students with visual impairment and their teachers in a computing course.

CHAPTER FIVE

A DIARY STUDY OF THE TEACHING AND LEARNING EXPERIENCE IN A HIGH SCHOOL PROGRAMMING COURSE

Introduction

The findings from studies 1 and 2 provided insights into the blind or visually impaired persons' educational experience taking computing courses. Consistent across both studies, BVI persons find current developer tools inaccessible even with screen reader technology. BVI persons find most learning materials presented in an inaccessible format and ineffective teaching methods from an educational perspective. As a result, BVI persons find their learning progression limited, potentially dissuading them from pursuing a computing career. Study 2 provided additional details from the teachers' perspectives. Consistent across the participants, teachers often modify existing computing curricula by removing modules deemed inaccessible to students or modifying materials into more accessible formats. Teachers often choose tools that best fit the environment and tasks and what works best for their students, consistent with previous work for choosing technology for computing courses [65]. The advantages of visual impairment, such as a greater focus and attention to detail and better memorization of code structure and keyboard commands, were also consistent in both studies.

Studies 1 and 2 used interviews to collect data from participants. Although helpful in capturing information from past experiences, interviews may incur response and recency bias. Also, data from recollection may lack details about any one particular experience. Thus, the third study utilized a method for capturing data at the point in which an event or experience occurs. The last study of this investigation will use a diary study method for capturing periodic data from teachers and students in the context of a high school programming course. A diary study method is a field study for collecting *in situ* longitudinal data about participants' behaviors, activities, or experiences [66]. The primary difference between a diary study and other field studies is that participants are the creators and collectors of their data [67]. Diary studies are most useful in the following conditions: 1) when the researcher(s) does not have to be presented for data collection, 2) for capturing data over an extended period, and 3) when it is ideal to avoid presentation effects that may cause participants to behave differently in the presence of the researcher [66], [67]. The necessity for the diary study comes from the COVID-19 pandemic, which enforced social distancing procedures.

Research Questions

The study of Chapter 5 seeks to answer the following research questions:

RQ1: What are ways blind or visually impaired students learn to write code?

RQ2: What are the ways teachers instruct blind or visually impaired students how to code?

Participant Recruitment and Demographics

Participants were recruited by contacting teachers and administrators at schools for the blind or visually impaired interested in participating. Study materials, approved by the institution's IRB, were sent to the participating sites for their superintendents' review and approval. Two schools originally agreed to participate in the study. However, one school could not participate due to complications caused by the COVID-19 pandemic.

The participating school, located in the northeastern US, includes one teacher and two students. Due to the low sample size and the possible revelation of their identification, the teacher and students' demographics will not be disclosed.

Study Apparatus

The study consists of participants completing and submitting diary entries supplied to them in Qualtrics surveys. The use of online surveys is ideal, given that they are proven to be accessible for screen reader users. There are four sets of surveys, two for teachers and two for the students; the demographic questionnaires, and the diary entry prompts. The diary entry prompt for students asks questions to assess the difficulty in understanding the day's lecture, completing any assignments given, and teacher instruction quality (see Appendix D). The teacher's diary entry prompt asks questions assessing their understanding of their students' comprehension of lecture and accompanying materials and their presentation of lecture concepts (see Appendix C).

Procedure

Once school and district officials for the participating sites approved the study, instructions were provided to the teacher, including the questionnaires to complete, the time frame to complete and submit each diary entry, and obtaining parental consent for participating students. The teacher was informed that the diary study would last for one month. After one month, the data collection may end, whether the course ends. The teacher first sought consent from the parents of the potential participating students before the study. Only students with consent from their parents participated. Demographic information was collected from participants through online Qualtrics surveys, separate for teachers and students. Data collection began when the teacher moved into the programming portion of their class. Students were to create and submit their entries after each lecture or after completing and submitting a homework assignment. Teachers submitted their diary entries after each lecture. At the end of the one-month data collection, students were compensated with a $25 gift card; teachers were compensated with a $20 gift card.

Case Study

The following subsections describe the month-long diary describing the teaching and learning of programming to students with visual impairments. In this course, the teacher taught programming using Hypertext Markup Language (HTML) and Cascading Stylesheets (CSS) to create web pages. The course is taught remotely using the curriculum from Code.org [3]. Each week provides the teacher's perspective on the students' progress with the lectures and assignments and their perspectives on the teacher's lectures and assignments. As part of the diary entry, students rated the difficulty of the assignment, lectures, and learning materials on a 5-point scale (1 = poor and 5 = excellent for lecture and assignment format, 1 = challenging, 5 = easy for the rest), as shown in Table 5.1. Teachers rated their students' comprehension of the lecture and assignment, as shown in Table 5.2.

Table 5.1: Students' rating of course lecture, materials, and assignments

Score	**Student**	**Week 1**	**Week 2**	**Week 3**	**Week 4**
Lecture Format	P1	5	5	5	0
	P2	4	5	N/A	3
Assignment Format	P1	4	5	5	5
	P2	5	5	N/A	3
Programming Language	P1	5	5	5	5
	P2	5	5	5	3
Editor/Tool	P1	4	5	5	5
	P2	5	5	N/A	4
Overall Class	P1	5	5	5	3
	P2	5	5	5	4
Overall Assignment	P1	5	5	5	N/A
	P2	4	5	N/A	3

Table 5.2. Teacher's rating of student performance during the course on a 1 (easy) to 5 (challenging) scale

Week	Students' perception of lecture	Students' perception of learning materials	Students' perception of assignment
1 (HTML headings & Debugging)	1	1	3
2 (Ordered and unordered lists)	1	1	1
3 (HTML webpage project)	N/A	N/A	1
4 (Styling with CSS)	3	3	5

Week One

In the first week, students learned how to code in HTML and debug errors in the code.

Teacher. The teacher used lessons from Code.org's curriculum to present the lecture. Code.org provides an online code editor for students to practice coding during the lecture. Students were assigned to find an error in the code (locate a p tag with a missing opening angle bracket) and change the HTML headings' size. According to the teacher, creating HTML elements (e.g., <p> to create paragraph elements) was easy for students to understand. However, when it came to the lesson's debugging aspect, the teacher disclosed that the students had trouble locating the editor's error. The color used to indicate the error is red, which was a problem for one of their students as they have a color deficiency.

> *"The debugging. I think they are still getting used to the code.org website. How to visually manage it. Also, the error code is in pink and one of my students uses reverse color."*

The teacher also expressed concerns about the potential visual fatigue as students are debugging. It may take visually impaired students longer to spot a small error, such as a missing angel bracket for an HTML tag.

> *Coding is very visual. I am worried they will have visual fatigue from all the work. You have to find the small error of a missing sign."*

Students were able to complete the assignment.

Students. The students confirmed they were learning about HTML headings and debugging in their entries. Both students found the lecture and the learning materials easy to use and understand. While the students found the assignment easy to do, they mentioned difficulty reading the code due to their impairment. Student P2 has a color deficiency and found it challenging to find the code's error, depicted in red.

> *"Sometimes it was tricky to see that some of the characters were red instead of black. It was more of a visual thing." (P2)*

Student P1 found the code's text small and thin, making it difficult to read the code lines.

> *"There was nothing that was really difficult except that it was a little hard to see. The text was really thin." (P1)*

Overall, the students felt the lecture and assignments were reasonably easy to understand.

Week Two

Teacher. The topic for the second week was on ordered and unordered lists. Like the previous week, lessons were presented through Code.org modules. The teacher assigned a review assignment of last week's lecture on changing the size of headings and debugging code. They believed the students were more comfortable with the concepts and could easily complete the assignment. Also, students had to create an ordered and unordered list. The students did not face much difficulty in completing the assignment. However, the teacher indicated students were facing some trouble writing code from scratch, a different approach from the previous week's assignment that provided code to the students.

> *"Adding code from scratch. They needed some guidance because this was the first time the lesson was more open-ended."*

Despite the difficulty, the students were able to complete the assignment.

Students. The students felt the lesson on the ordered and the unordered list was easy to comprehend and then implement in their assignment. Student P1 credited their ability to complete the assignment to the teacher's clear and understandable instructions.

> *"The instructions for the assignment made the assignment very easy to understand." (P1)*

Student P2 admitted to copying and pasting the code and changing the words to avoid having to rewrite the necessary HTML tags (i.e., <ol> for ordered lists, <ul> unordered lists, <li>) to create the lists. P2 notes that using Zoom Text helped to invert the code's colors to see the codebase better.

> *"not really if I use Zoom Text to invert the colors I can see everything well." (P2)*

Overall, both students did not encounter any challenges in the week's lecture.

Week Three

Teacher. During week 3, the teacher assigned a project where students built their web page to demonstrate their understanding of HTML. The webpage must encompass everything from lessons in weeks 1 and 2. The teacher noted that while one student easily accomplished the task, the other student struggled initially because they briefly forgot the syntax of HTML. However, both students were able to complete the project for the week.

Students. Only student P1 submitted an entry for this week, and P1 could complete the project with no challenges encountered. As there was no lecture corresponding to week 3, P1 did not offer their thoughts.

Week Four

Teacher. For the final week, students learned about CSS, the stylesheet markup for styling webpages. The teacher used Code.org's module on CSS. The students were assigned to debug CSS code and create a CSS stylesheet to link to an HTML file to style the webpage. The teacher mentions that while the students performed well on the debugging task, they had a challenge in creating a CSS file from scratch and linking it to the HTML file.

> *"When given a blank page to write css, and link it into HTML they found that challenging."*

A challenge the teacher noticed for the students learning CSS is using semicolons (;) to complete a styling rule.

> *"All the references to semicolons, etc was confusing. It was a lot to absorb at once."*

Overall, the teacher believed the students could complete the assignment, despite the difficulties encountered.

Students. Student P2 found using CSS to style a web page initially confusing as they were trying to understand where to add the styling rules.

> *"I found it a little confusing where to put the new code. Like for one of the activities they made me write the code on a whole new blank page. Then they wanted me to change the color of an already written out paragraph so it was kind of confusing." (P2)*

Student P1 would have preferred if the activity for styling the webpage was separate from the user interface, where all the buttons are located:

> *"It would be easier if the activity I'm doing is separate from the buttons I have to click." (P1)*

Overall, both students could complete the assigned tasks for week 4.

Discussion and Conclusion

Findings showed the students' ability to learn coding depending on the choice of language(s), the tools to use, and how the instructor assists with their learning. Students could adapt to the lecture's challenges and accompanying assignments and complete their assigned tasks during the four weeks. Despite achieving their learning outcomes, there were some instances of accessibility challenges for the students, such as small and thin text and lack of identification of errors in the code. However, students rated the lessons, materials, and assignments as easy to understand and perform. The teacher rated the students as understanding the learning materials and working through the assignment's tasks from the teacher's perspective. As the course took place through online instruction, the teacher may have been comfortable using the online resources from Code.org as the materials would be readable by screen readers. Supporting the learning materials from Code.org is the availability of an online code editor allowing for students to practice the coding lessons without the need for an external tool. Hence, online learning may be a potential pathway to introduce more accessible computing education for blind or visually students for both in-school and out-of-school learning.

The small sample size limited the study, with only one teacher and two students participating. Even with a one-month data collection period, the small sample size resulted in few data points to produce generalizable findings. Additionally, data collection began when classes switched to programming lessons and, thus, skipped potential lessons on computing concepts that students may have found easy or challenging to learn. Students may have misinterpreted questions involving rating the homework assignment and in-class assignment. For instance, student P1 indicated a zero (0) (N/A) for questions about the in-class assignment format and overall difficulty when the rating may have been for the homework assignments.

The diary study enabled a glimpse into the educational setting for students learning to code using online learning tools. From the entries, both the teacher and their students feel the learning materials and tools provided were usable for their learning and, in turn, were satisfied with the method of instruction. In the next chapter, all three studies are summarized to inform the essential themes discovered in the investigation.

CHAPTER SIX

SUMMARY OF INVESTIGATION

Introduction

The previous three chapters (3, 4, and 5) detailed three different studies investigating the accessibility of computing education (CEd) from three different perspectives. Chapter 3 focused on interviews with professional programmers with visual impairments to understand how their visual impairment 1) impacted their ability to contribute to their work environment and 2) their education and training to become programmers. Chapter 4 focused on interviews with K-12 teachers of visually impaired students to seek their perspective on how students learn using current computing tools and curricula and what strategies teachers use to provide an accessible learning environment for their students. Chapter 5 was a diary study of students learning to code in a virtual classroom setting and recording their experience going through the lectures and working through the coding lessons. All three studies contributed findings that help frame a larger picture of what factors serve as barriers for teachers and students in CEd and a possible solution to address some or all of such barriers. Table 6.1 summarizes the critical themes from the multi-study investigation with contributing studies. The following sections describe these themes in greater detail. These findings help to answer RQ1 of my research:

RQ1: What are the perceptions of blind or visually impaired persons regarding the accessibility of the curricula, tools, and pedagogies in K-12 computer science education?

Table 6.1: Major themes from the three investigative studies and which studies contributed to each theme (S1 = study 1 (Chapter 3), S2 = study 2 (Chapter 4), S3 = study 3 (Chapter 5))

Theme	**Contributing Studies**
Curriculum and Learning Materials	S1, S2, S3
Developer Tools	S1, S2, S3
Advantages of Visual Impairment	S1, S2
Online Learning and Electronic Access	S1, S2, S3

Curriculum and Learning Materials

In all three studies, the accessibility of computing curricula, including supporting materials, was a significant theme that impacted students' progression with visual impairments and teachers. Participants in S1 brought up in their reflection how they often looked for additional resources through online tutorials and materials in electronic format to supplement their learning due to the inaccessibility of the resources provided in their courses. The biggest challenge is the graphics and other visual elements that contained no description or alt text, making it difficult for screen readers to describe the graphic correctly. Teachers in S2 voiced similar concerns about current materials supplied in the curricula they use. Some of the teachers mentioned how they would modify materials to suit their students' needs and preferences, depending on the severity of their impairment. Sometimes they cut down the curriculum if they feel the lesson would prove to be inaccessible. The teachers fear that without accessible materials, students with visual impairments may feel they are not progressing at the same pace and level as sighted students and may feel they are not adequately supported. Such sentiments were echoed by participants in a study from Baker et al. concerning undergraduate CEd [14]. In S3, entries from the students showed a different perspective on their learning experience. According to the students, they enjoyed their learning experience using the lesson modules from Code.org's curriculum. There were only a few issues viewing the online lessons (small, thin text and color of code error for people with color deficiencies). However, overall, the students comprehended the concepts and completed the assignments for each week. In S2, a few teachers used a version of the CS Principles course that can be used with the Quorum programming language to provide a more inclusive and accessible learning experience [68], [69]. This adaption may serve as a starting point for improving other computing curricula to be more inclusive in their design and student experience.

Developer Tools

Developer tools, particularly code editors and integrated development environments (IDEs), were another theme from all the investigate studies that serve as barriers for blind or visually impaired students in computing education. Programmers in S1 reflect on how the IDEs used at their jobs are not accessible, impacting their performance. Programmers spoke of navigating the complex user interface (UI) of IDEs because they were not designed to facilitate navigation with a screen reader. Often, programmers utilize simpler code editors with fewer features but with minimal menus to navigate. Findings from S2 corroborate the findings from S1. Teachers mentioned how certain editors and programming software could be difficult for the students to use because they do not work well with their assistive technologies. Even online editors have been a challenge in certain instances for students (S2, S3). The issues described in the investigative studies echo similar findings from research seeking to understand difficulties for blind or visually impaired programmers [14], [27], [37], [39], [40], [57], [70], [71]. Such tools' design represents the assumption that the *typical* developer has a full 20/20 vision and does not use assistive technologies. Current tools should undergo a redesign using an inclusive design approach. Future tools should be designed with such an approach from the beginning to include more people with visual impairments in computing.

Perceived Advantages of Visual Impairment

From the first two studies (S1, S2), there were common mentions of the advantages of being visually impaired over sighted developers. Both studies alluded to the notion that having low or no vision may help with specific programming tasks, even better than programmers with full vision. Participants from S1 noted how their lack of vision removed the distraction of external visual stimuli, enabling greater concentration on the task at hand, such as code comprehension. Relying on screen readers to navigate code, line by line, helps them detect code errors better than programmers with sight, who may overlook more subtle errors when quickly skimming, according to the participants. Also, participants claim they have a better mental model of code structure and memorization of shortcut keys, mainly due to necessity as their limited vision forces them to keep such information in their memory. In S2, teachers noted their students' greater focus on understanding code and memorizing concepts. While these advantages help blind or visually impaired developers execute certain tasks, memorizing code structure and keyboard shortcuts may negatively impact their cognitive load. Using a complex IDE may require memorizing many keyboard shortcuts to navigate to the code editor, select a line of code, copy and paste code, trigger specific features, and other common tasks. Voiced in S1 and S2 is the need for simplifying the UI, which may also include reducing the number of shortcuts needed to operate an editor.

Online Learning and Electronic Access

There were mentions of BVI persons accessing electronic resources from the Web, watching online tutorials, or using Web-based code editors in all three studies. In S3, the teacher conducted the course online using video calls and the code editor from Code.org. These instances are examples of online or electronic learning (e-learning). E-learning leverages internet technologies and online multimedia to provide learning resources and instruction to students [72]. E-learning can be any method of delivering education electronically, such as online videos, video calls, conference calls, and interactive websites. E-learning continues to surge in the U.S. as a popular form of distance education. In 2018, over six million students enrolled in distance education courses in degree-granting institutions [73]. In 2019, 46% of teachers in degree-granting institutions had taught an online course [74].

Distance education is not a new phenomenon; research has traced its roots back as early as the 18th century [75]. The first form of distance learning was correspondence education; teachers mailed instructions and assignments to students to complete, who then returned materials for grading and feedback [75]. Correspondence education was the earliest form of providing educational opportunities for those who did not have access to local educational institutions. The early 1900s saw advancements in distance education with the radio in 1921 and the television following after [75], [76]. Online learning became a reality in the 1980s with the rise of computers with internet capabilities. The University of Phoenix was the first institution to establish an online campus in 1989 [77]. With the introduction of the World Wide Web in the early 90s, online learning became a widespread avenue for people worldwide.

Online learning offers several benefits over traditional classroom instruction: flexibility of teaching and learning, accessibility (in terms of accessing learning content and communication), and the asynchronous nature of the learning, allowing students to control the pace of their learning. Online education is also supported by a variety of electronic tools such as video conferencing (i.e., Zoom or Google Meet), online discussion forums (i.e., Piazza), and collaboration tools (i.e., Google Jamboard, Notion). Some notable disadvantages to online learning include potential technology issues, students becoming less attentive while on the internet, reduced student-teacher interaction, and a lack of student feedback [78]–[80]. Research also suggests that collaborative learning and teacher rating perceptions are lowered in online learning [78]. Additional concerns about online learning may impact people with disabilities more than others. For blind or visually impaired (BVI) students, switching to electronic platforms and materials brings questions about whether they are accessible with screen readers, assistive technologies that read visual displays. This is a significant concern for educators, especially when comparing the needs of blind vs. visually impaired students [81]. Prerecorded videos can also be detrimental for BVI students without context for describing certain topics. For example, in a programming course, if the teacher references a line of code, pointing to the line on a whiteboard or slide without mentioning the line number or position, it will make it difficult for BVI students to understand where the code exists in the codebase. Using non-descriptive images in materials can also disadvantage BVI students and those with learning disabilities such as dyslexia. Live teaching formats can be challenging for the deaf or hard of hearing if no live captioning is provided [82]. Even with live captioning, language barriers from the teachers may affect captioning accuracy. Research suggests how the social element of learning is more beneficial in traditional classroom environments for students with disabilities than in online environments. The remote nature of online learning negatively impacts the sense of inclusivity by students with disabilities, and it makes them feel uncomfortable when engaging with the class [83].

Discussion and Conclusion

This chapter summarized the findings from the three investigative studies. It produced themes that generalize the most critical aspects of understanding accessibility in CEd for students with visual impairments. The main themes extracted from the findings include *curriculum and learning materials*, *developer tools*, *advantages of visual impairment,* and *online learning and electronic access*. From my investigation, improving accessibility in online learning is the most plausible approach towards improved learning experiences for BVI learners and, therefore, is the direction I take in answering RQ2:

RQ2: What interventions can improve accessibility for blind or visually impaired persons in K-12 computer science education?

In the following chapters, I discuss the process of designing, developing, and evaluating an online learning management system (LMS). An LMS is an online platform for administering, reporting, tracking, and managing courses or training programs. Some of the basic features of an LMS include but are not limited to the following: a) managing courses or programs, b) managing student or personnel roster, c) maintaining records of academic or professional performance, d) administering educational or training lessons and evaluations, and e) generating reports. The following studies will help to answer the following questions:

R1: What design requirements are necessary for blind or visually impaired students to leverage an online learning platform to learn to program?

R2: What design requirements are necessary for teachers of blind or visually impaired students to leverage an online learning platform to teach programming?

R3: In what ways does an online learning platform improve the usability and accessibility in online learning programming for students with visual impairments?

I will develop the prototype LMS following a user-centered design (UCD) approach to answer the questions above. UCD is a design philosophy that emphasizes taking a product or system's end-users' input and feedback into every phase of the design process [84]. Products or systems developed using UCD may offer a better user experience (UX), resulting in greater user satisfaction [84]. There are generally four phases in UCD. The first phase is to collect data to understand how the context in which the technology is used. In the second phase, the data will be analyzed to generate a series of user requirements that will inform the development of several design solutions (phase three) or prototypes evaluated in the fourth phase to see if it meets the users' satisfaction. The exploratory studies (chapters 3, 4, 5) were part of the first phase of UCD. The next chapter details a survey to collect data from post-secondary students, both sighted and visually impaired, to learn about their experience taking online courses, their perceived confidence in taking such courses, and the perceived advantages and disadvantages of online learning.

CHAPTER SEVEN

A SURVEY OF READINESS FOR AND EXPERIENCE WITH ONLINE LEARNING

Introduction

My studies exploring accessibility of computing education for blind or visually impaired (BVI) students revealed several critical findings considerably impacting their journey towards pursuing a career in computing. Existing computer science (CS) curricula are largely still inaccessible for BVI students because of 1) lecture materials that are not offered in a variety of accessible formats (Chapters 3 & 4), 2) topics not taught in a way that BVI persons may follow or comprehend (Chapter 4), and 3) instructors who do not have the institutional support or prerequisite training to teach CS to BVI students adequately (Chapter 4). Inaccessible code editors and development environments not usable by screen reader users further compound the lack of accessibility in CS classrooms (Chapters 3 & 4). A positive finding from the exploratory studies was the perceived advantages and benefits of visual impairment, mainly better concentration, focus, memorization of code structure, and attention to detail (Chapters 3 & 4). Another, more critically potent finding was the consistent use of online learning as a medium to support accessible learning in CS for BVI students. In all three studies, students, developers, and teachers expressed how electronic resources such as online PDFs, cloud-based services (e.g., Google Docs), and learning management platforms (i.e., Code.org) made learning CS concepts more accessible easier to use. In Chapter 5, the context of the diary study was a distance-learning course using the editor and curriculum provided by

Code.org [3]. While research has examined the use of online learning for students in different forms (MOOCS, for example) and different students, there hasn't been sufficient research into its accessibility, particularly for students with visual impairments. There also is a lack of research looking at differences in perceived ease or difficulty in online learning between sighted students and students with visual impairments. Before seeking to design online learning spaces to include people with visual impairments, additional research is required to understand how such students feel about the prospect of taking fully online courses and how their experiences may compare or contrast with sighted students. In this chapter, I detail a survey study that sought to learn more about perceptions of online learning and the past experiences of students with sight and visually impaired students.

Research Questions

The following research questions guide this study:

RQ1: What competencies do blind or visually impaired students perceive as important for their readiness for online learning?

RQ2: What competencies do sighted students perceive as important for their readiness for online learning?

RQ3: What are blind or visually impaired students' perceptions of their confidence in their readiness for online learning?

RQ4: What are sighted students' perceptions of their confidence in their readiness for online learning?

RQ5: What are student perceptions of the advantages of online learning?

RQ6: What are student perceptions of the disadvantages of online learning?

RQ7: What challenges are faced by blind or visually impaired students in their experience taking online courses?

Research questions RQ1, RQ2, RQ3, and RQ4 will be answered by the responses to the Student Readiness for Online Learning instrument by Martin et al. [85]. Research questions RQ5, RQ6, and RQ7 will be answered from the open-ended questions, with questions specifically for blind or visually impaired students answering RQ7. Table 7.1 describes the relationship between the research questions and measures used.

Table 7.1 Relationship between research questions, hypotheses, and measures.

Research Question	**Measures**
RQ1: What competencies do blind or visually impaired students perceive as important for their readiness for online learning?	Student Readiness for Online Learning
RQ2: What competencies do sighted students perceive as important for their readiness for online learning?	Student Readiness for Online Learning
RQ3: What are blind or visually impaired students' perceptions of their confidence in their readiness for online learning?	Student Readiness for Online Learning
RQ4: What are sighted students' perceptions of their confidence in their readiness for online learning?	Student Readiness for Online Learning
RQ5: What are student perceptions of the advantages of online learning?	Open-ended UX questions
RQ6: What are student perceptions of the disadvantages of online learning?	Open-ended UX questions
RQ7: What challenges are faced by blind or visually impaired students in their experience taking online courses?	Open-ended UX questions

Participants

Recruitment

Participants for the study had to meet the following criteria, based on the group they would fall in: for sighted students, they needed to be at least 18 years of age; for the BVI students, they had to identify as having a visual impairment and be at least 18 years of age. Participants must be current or former students at a college or university, and it was not required to have earned a degree. In addition, participants had to have taken at least one fully online course. Questions were added to the survey to screen participants to ensure they met the criteria for one of the groups. Electronic advertisements were dispersed through listservs, social media platforms, and snowball sampling. Interested participants would follow the link embedded in the advertisement to participate in the study.

Demographics

In total, 592 responses were recorded, of which 196 were used for analysis as the rest were found to be data entered through online bots. From the data, 72 (36.7%) were students identifying as blind or visually impaired, with 66 (91.7%) responding as low vision and the remaining six as blind. Participant ages ranged from 18 to 61, with a mean of 24.3 and a standard deviation of 6.3. Regarding gender, 116 (59.2%) were men, 74 (37.8%) were women, 4 (2%) identified as non-binary or third-gender, while the remaining 2 (1%) preferred not to reveal their gender identity. In terms of racial/ethnic makeup, 78 (39.8%) of respondents identified as White, 39 (19.9%) as Black or African American, 33 (16.8%) as American Indian or Alaska Native, 21 (10.7%) as Hispanic or

LatinX, and 9 (4.6%) as Asian. Regarding school classification, 56 (28.6%) responded as Seniors, 34 (17.4%) as Sophomores, 31 (15.8%) as Postbaccalaureate, 24 (12.2%) as Juniors, and 20 (10.2%) as having a Graduate Certificate. On average, respondents have taken six online courses with a standard deviation of 5.7. Regarding the assistive technologies (AT) used by the BVI respondents, screen readers and magnification were among the most often cited (many of the participants use more than one AT, such as screen reader and magnification). The full demographic breakdown can be seen in Table 7.2.

Table 7.2: Student demographics (N = 196)

Variables		**Mean/Frequency and (Percentage)**

Age		Mean = 24.31
		SD = 6.34

Gender	Men	116 (59.2%)
	Women	74 (37.8%)
	Non-binary/third-gener	4 (2%)
	Prefer not to say	2 (1%)
Race/Ethnicity	White	78 (39.8%)
	Black/African American	39 (19.9%)
	American Indian/Alaska Native	33 (16.8%)
	Hispanic/LatinX	21 (10.7%)
	Asian	9 (4.6%)
Academic Classification	Freshman	9 (4.6%)
	Sophomore	34 (17.4%)
	Junior	24 (12.2%)
	Senior	56 (28.6%)
	Post-baccalaureate	31 (15.8%)
	Graduate Certificate	20 (10.2%)
	Masters	14 (7.1%)
	Doctoral	8 (4.1%)
Vision	No	124 (63.3%)
	Yes	72 (36.7%)
Assistive Technologies*	Screen reader	41
	Magnification	51
	Braille display	10
	Other	3
	Do not use assistive tech	5

Online Courses		Mean = 6
		SD = 5.7

* Total does not equal number of visually impaired students as some may use more than one assistive technologies

Instrument

The survey was constructed by adopting the Student Readiness for Online Learning (SROL) instrument used by Martin et al. [85], which examined student perception of their readiness for online learning. The instrument consisted of 20 statements, with five statements belonging to each of the four competencies: *Online Student Attributes*, *Time Management*, *Communication*, and *Technical*. Each statement was answered twice on a 5-point Likert scale; once for perceptions of importance (1 = not at all important, 5 = very important), and again for perceptions of confidence (1 = not at all confident, 5 = very confident). Additionally, there were open-ended questions regarding perceived advantages and disadvantages of online learning, communication and learning management platforms used, and any challenges encountered. There were additional questions for BVI students asking about the impact their disability may have had on their online learning experience and what, if any, accommodations were provided. The survey can be found in Appendix E.

Procedure

When respondents click on the link in the ad, they are taken first to a one-question survey where they can read the informed consent explaining the purpose of the study, their role in the study, risks, and benefits. Respondents who wish to participate will enter the email address to answer the question and click “Next” to proceed to the actual survey. The email address is used for the drawing to receive a $50 gift card. The survey was designed for respondents to complete in approximately 20 minutes. Once they submitted their responses, their role in the study was complete. Two random drawings were conducted, one for sighted and one for the visually impaired respondents. Each winner was contacted and provided their compensation electronically.

Results

In this section, I discuss the findings of the survey. First, I describe the analysis of comparing the means of the responses by perceived importance and confidence for each competency of the SORL. Second, I describe the analysis of the demographic factors and student perception of importance and confidence in each competency for online learning.

Student Perception of Importance and Confidence of Online Learning Competencies

I observed the means of the responses for each statement separated by perceived importance and confidence (Table 7.3). All the means were above a 3.0 but below a 4.0. Students rated each competency as neither unimportant nor important and indicated as neither unconfident nor confident in their ability for each competency. The means suggest students were indifferent in how they perceived the importance of each competency to succeed in online learning and uncertain about their readiness in taking online courses.

Table 7.3: Student Readiness in Online Learning Descriptive Statistics

Student Readiness for Online Learning Competencies	**Importance M(SD)**	**Confidence M(SD)**
Online Student Attributes		
Q1 Set goals with deadlines	3.45(1.35)	3.58(1.25)
Q2 Be self-disciplined with studies	3.75(1.27)	3.63(1.16)
Q3 Learn from a variety of formats (lectures, videos, podcasts, online discussion/conferencing)	3.63(1.24)	3.66(1.17)
Q4 Be capable of following instructions in various formats (written, video, audio, etc.)	3.68(1.22)	3.7(1.1)
Q5 Utilize additional resources to answer course-related questions (course content, assignments, etc.)	3.74(1.19)	3.8(1.13)
Mean(SD)	3.65(0.95)	3.67(0.87)
Time Management		
Q6 Devote hours per week regularly for the online class	3.61(1.19)	3.56(1.22)
Q7 Stay on task and avoid distractions while studying	3.87(1.22)	3.51(1.18)
Q8 Utilize course schedule for due dates	3.61(1.21)	3.7(1.09)
Q9 Complete course activities/assignments on time	3.8(1.24)	3.84(1.16)
Q10 Meeting multiple deadlines for course activities	3.81(1.14)	3.72(1.19)
Mean(SD)	3.74(0.85)	3.67(0.83)
Communication		
Q11 Use asynchronous technologies (discussion boards, email, etc.)	3.65(1.08)	3.58(1.26)
Q12 Use synchronous technologies (Webex, Collaborate, Adobe Connect, Zoom, etc.) to communicate	3.6(1.16)	3.63(1.17)
Q13 Ask the instructor for help via email, discussion board, or chat	3.82(1.16)	3.7(1.17)
Q14 Ask classmates for support (accessing the course, clarification on a topic)	3.5(1.23)	3.59(1.26)
Q15 Discuss feedback received (assignments, quizzes, discussion, etc.) with the instructor	3.65(1.25)	3.59(1.15)
Mean(SD)	3.65(0.84)	3.62(0.88)
Technical Competence		
Q16 Complete basic computer operations (e.g., creating and editing documents, managing files, and folders)	3.69(1.19)	3.76(1.22)
Q17 Navigate through the course in Learning Management Systems (e.g., Moodle, Canvas, Blackboard, etc.)	3.69(1.24)	3.8(1.17)
Q18 Participate in course activities (discussions, quizzes, assignments synchronous sessions)	3.78(1.15)	3.73(1.2)
Q19 Access the online grade book for feedback on performance	3.78(1.15)	3.65(1.23)
Q20 Access online help desk/tech support for assistance	3.72(1.27)	3.78(1.21)
Mean(SD)	3.73(0.85)	3.75(0.89)

Two repeated-measures ANOVA for importance and confidence were used to determine differences amongst the competencies. For the importance competencies, the assumption of sphericity was violated (W = 0.93, p = .02). Therefore, the degrees of freedom were corrected using Greenhouse-Geisser estimates of sphericity (ε = 0.95). There was a statistically significant difference between the competencies (F(2.85, 555.75) = 3.01, p = .03, η^2 = .003). Post hoc analyses revealed no statistically significant differences between the competencies. These findings suggest that while perceptions of importance differ by the competency, students did not feel that one domain was more important than the other. For the confidence competencies, there was a statistically significant difference (F(3, 585) = 3.32, p = .02, η^2 = .003). Post hoc analysis revealed that students were more confident in technical competence (M = 3.75) than communication (M = 3.62).

Additionally, I conducted repeated-measures ANOVA for importance and confidence to determine differences between competencies for sighted students and visually impaired students separately. Observing perception of importance from sighted students, the assumption of sphericity was violated (W = 0.87, p = .005); therefore, the degrees of freedom were corrected using Greenhouse-Geisser estimates of sphericity (ε = 0.92). There was a statistically significant different between between the competencies (F(2.76, 339.48) = 3.44, p = .02, η^2 = .004). Post hoc analyses revealed no statistically significant differences between the competencies, suggesting that sighted students did not feel one domain more important than the other. For perception of confidence there was a statistically significant difference (F(3, 369) = 5.71, p < .001, η^2 = .006). Post hoc analysis suggest that sighted students were more confident in the technical domain (M = 3.74) than their online attributes (M = 3.6), time management (M = 3.59), and communication (M = 3.55). Observing perception of importance from blind or visually impaired students, the assumption of sphericity was violated (W = 0.84, p = .04); therefore, the degrees of freedom were corrected using Greenhouse-Geisser estimates of sphericity (ε = 0.92). There was not a statistically significant difference between the competencies (F(2.76, 195.96) = .394, p = .76, η^2 = .001). For perception of confidence, there was not a statistically significant difference between the competencies (F(3, 213) = .366, p = .78, η^2 = .001). These findings suggest that blind or visually impaired students did not perceive one competency as more important than the other nor felt more confident in any one competency.

Demographic Factors and Student Perception of Importance and Confidence in Online Learning Competencies

I analyzed the student perceptions of importance and confidence in the competencies for online learning by demographic factors. The factors considered for analysis were gender (as man and non-man), race or ethnicity (as white and non-white), education (as undergraduate and graduate), and vision (as yes and no). Given that there are multiple correlated outcome variables (importance and confidence), I conducted MANOVAs to examine the difference in perceptions by student characteristics. You can see the descriptive statistics of the perceptions of importance and confidence across all competencies by demographics are shown in Tables 7.4 and 7.5, respectively.

Table 7.4 Student perception of importance

Demographic	Student Attribute Mean(SD)	Time Mgmt Mean(SD)	Comm Mean(SD)	Technical Mean(SD)
Gender				
Man (N =116)	3.55(0.86)	3.55(0.85)	3.52(0.9)	3.6(0.9)
Non-man (N = 80)	3.86(0.85)	3.83(0.79)	3.76(0.85)	3.96(0.84)
Race				
White (N = 78	3.57(0.76)	3.66(0.72)	3.59(0.84)	3.7(0.8)
Non-white (N = 118)	3.75(0.93)	3.67(0.9)	3.64(0.92)	3.78(0.95)
Academic Standing				
Undergraduate (N = 123)	3.65(0.91)	3.62(0.91)	3.58(0.95)	3.69(0.96)
Graduate (N = 73)	3.72(0.8)	3.75(0.69)	3.68(0.77)	3.84(0.77)
Vision				
No (N = 72)	3.8(0.66)	3.79(0.66)	3.74(0.77)	3.75(0.74)
Yes (N = 124)	3.6(0.97)	3.59(0.91)	3.55(0.94)	3.74(0.97)

Table 7.5 Student perception of confidence

Demographic	Student Attribute Mean(SD)	Time Mgmt Mean(SD)	Comm Mean(SD)	Technical Mean(SD)
Gender				
Man (N =116)	3.55(0.86)	3.55(0.85)	3.52(0.9)	3.6(0.9)
Non-man (N = 80)	3.86(0.85)	3.83(0.79)	3.76(0.85)	3.96(0.84)
Race				
White (N = 78	3.57(0.76)	3.66(0.72)	3.59(0.84)	3.7(0.8)
Non-white (N = 118)	3.75(0.93)	3.67(0.9)	3.64(0.92)	3.78(0.95)
Academic Standing				
Undergraduate (N = 123)	3.65(0.91)	3.62(0.91)	3.58(0.95)	3.69(0.96)
Graduate (N = 73)	3.72(0.8)	3.75(0.69)	3.68(0.77)	3.84(0.77)
Vision				
No (N = 72)	3.8(0.66)	3.79(0.66)	3.74(0.77)	3.75(0.74)
Yes (N = 124)	3.6(0.97)	3.59(0.91)	3.55(0.94)	3.74(0.97)

*Gender***.** There was a statistically significant difference between male and non-male student respondents on the perception of importance of online readiness and confidence in their readiness for online learning, particularly for student attributes (Wilks' Lambda = 0.97, F(2, 193) = 3.21, p = .04, partial eta squared = .03), time management (Wilks' Lambda = 0.96, F(2, 193) = 3.93, p = .02, partial eta squared = .04), and technical competencies (Wilks' Lambda = 0.95, F(2, 193) = 5.26, p = .01, partial eta squared = .05). Post hoc discriminant function analysis (DFA) suggested that non-male students rated their competencies higher than male students, in terms of perceptions of importance and confidence.

Race. Across all measures, responses from non-white students were higher than white students. However, there was no statistically significant difference between white and non-white students on the perception of the importance of online readiness competencies and confidence in their readiness for online learning.

Academic Standing. There was no statistically significant difference between undergraduate and graduate students on the perception of the importance of online readiness competencies and confidence in their readiness for online learning.

Vision. Across all measures, responses from blind or visually impaired (BVI) were higher than sighted students. However, there was no statistically significant difference between BVI and sighted students on the perception of the importance of online readiness competencies and confidence in their readiness for online learning.

Qualitative Analysis

The survey contained several open-ended questions about students' experience in online learning to gain deeper insights into the advantages and disadvantages of taking online courses. There were additional questions for students who identified as having a visual impairment to understand how their disability impacted their online learning experience. I conducted a qualitative analysis using In Vivo Coding to observe the trends within the responses.

Advantages vs. Disadvantages. The main themes stemming from students' responses regarding perceived advantages of online learning include *convenience, flexibility, comfort and relaxation,* and *saving time*. Students found online learning convenient for them as they do not have to leave home to go to class, all the course materials are online, and lectures were recorded and can be watched at any time. The latter response also applies for flexibility as students felt the asynchronous nature of learning provided a self-paced learning environment, as indicated from one student response:

> *"Flexibility to attend class and complete assignments from anywhere."*

Regarding perceived disadvantages to online learning, the recurring themes include *communication/interaction, concentration/focus,* and *eyesight*. Students expressed the challenge in communicating with students and teachers online, especially synchronously. A common concern was the inability to seek help from teachers consistently, and it makes help-seeking difficult and may impact students' learning progression.

> *"It is harder to get that connection with your professor. A lot of my professors gave us busy work and when we needed help they were unavailable to help us. It was just hard to reach out to the older professors who were not updated with technology because they never really used technology like that in their in-person courses."*

Another disadvantage to online learning from the students' perspective is how easily distracted they become because they are not in a physical classroom. Students expressed how they find it difficult to concentrate on the lectures and to do work. Learning from home may cause students to become complacent and not pay attention to the lecture due to environmental distractions.

> *"It's easy to get distracted by things around you."*

Students indicated that looking at screens for learning can negatively affect their eyes. Particularly, students expressed having sore eyes from watching a screen for extended periods, making it difficult to continue participating in online classes due to eye strain.

Biggest Online Learning Challenge. Students reported that some of their biggest challenges in online learning were 1) accessibility, 2) staying focused, 3) technology challenges, and 4) the instructors. The BVI students struggled with inaccessible content posted on learning platforms and voicing their needs as someone with a disability to the teachers and classmates online. Students often have to find workarounds to stay on pace with their sighted classmates. Similar to responses to disadvantages of online learning, the ability to remain focused in an online classroom while in one's home environment. Students expressed how easy it is to get distracted or disinterested in the lecture and simply turn off their camera to do something else.

> *"sometimes content and projects were not accessible to JAWS and NVDA and having to find work arounds to meet those projects and assignments, sometimes even requiring sighted assistance."*

> *"Having to stay focused and avoid distractions while being at home."*

Students responded with challenges in contacting or seeking help from their instructors online, or the instructors would increase their workload due to constraints in the semester schedule. While in the classroom, students can ask teachers questions in real-time if they do not understand something in the lecture; other times, students can go to the teacher's office to seek help after class. The students responded that teachers are less available in online courses, making it harder to get their help.

> *"Can not ask the teacher at any time, resulting in some knowledge did not understand."*

Additionally, students felt it was difficult to connect with their classmates and make friends online, creating a sense of isolation from everyone else.

> *"Biggest challenge that was staying motivated to learning the materials for class. Because the social interaction was missing, it was difficult to form strong bonds with my classmates to complete task."*

> *"Biggest challenge is spending too much time on something and not being able to easily communicate/ cooperate with a professor, teaching assistant, or class mate. Being alone and isolated for most of the day warps my perception of time."*

Many respondents expressed having technical problems that made consistent access to online courses challenging, such as internet connectivity problems or issues with their electronic devices. Students cited concerns that such issues can impact their academic progress.

> *"My biggest challenge is connectivity. I always fear to miss class for connection error. I purchase mobile data."*

Challenges for Blind or Visually Impaired. When asked about the impact of their visual impairment on their online learning experience, students responded with accessibility, reduced learning progression, strain of the eyes, and isolation. Most of the students expressed the lack of accessible learning materials, assignments, and projects, which hindered their progression in the class. Further exacerbating the challenges is instructors unfamiliar with the needs of BVI students and their inability to accommodate their needs. It forces the students to become more independent learners as a result.

> *"there is more time dedicated to learning compared to my sighted counterparts. Finding work arounds to challanges and projects at hand."*

> *"It is a huge disadvantage since it motivates the segregation from students or even instructors, the poor accessibility are additional battles that we need to fight every semester with each course."*

Low vision students who use their usable vision to watch the screen report on the strain on their eyes for an extended period of watching a screen. The strain makes it challenging to pay attention to the lecture or do their work, further contributing to the reduced learning progression.

> *"I had to get blue light glasses because looking at the computer for an extended period of time caused me headaches. I have a previous history of concussions and it brought back this side effects."*

Discussion

Importance vs. Confidence for Online Learning

The survey revealed perceptions of the importance of online learning competencies and confidence in online learning competencies by college/university students, both overall and separated by visually impaired and sighted students. Findings show that students did not find any of the four competencies (online attributes, time management, communication, and technical) any more important than one another. Students equally find all four competencies important to succeed in taking online courses. Students did express higher confidence in their technical competency than the other competencies. The technical competence assesses students’ ability to perform basic computer operations, navigation learning management systems (LMS), and online access to their grades. Students may find such tasks straightforward to do with some initial training. Examining the perceptions specifically for sighted and BVI students, the findings show differences in perceptions of importance and confidence for sighted students but not for BVI students. Sighted students were more confident in their technical competence than the other competency. BVI student perceptions were indifferent between the competencies. The means across the measures were between a 3 and a 4, indicating between neither important nor unimportant and somewhat important and between neither confident nor unconfident and somewhat confident. These findings suggest BVI students find all the competencies somewhat important for their success in online learning but are unsure about their confidence in taking online courses. For BVI students, additional factors may contribute to their success in online learning not covered in the survey, such as accommodations by their instructors, home environment, and proficiency in using their assistive technology to use a computer. These factors have a more significant impact on BVI students than sighted students, explaining the differences in their perceived confidence.

Benefits and Challenges of Online Learning

Students expressed how online learning provides many benefits over traditional classroom learning. The most prevalent benefits were the flexibility and convenience of learning from home or other locations. Students working full-time or having family or professional obligations would find online learning a flexible option because there is no requirement to be in a physical classroom. Many for-profit institutions use the fully online classroom format because it is a popular avenue for professionals to earn their degrees while not sacrificing their careers. Participants mentioned notable drawbacks to online learning, the most common being the lack of social interaction among classmates, seeking help from their instructors, and difficulties retaining focus. Students noted that the online environment is not conducive to making friends and connecting with classmates for help with classwork. Of note, students find it is more challenging to contact teachers virtually; students find they are less accessible online than in the classroom. While learning remotely offers flexibility in learning spaces, such flexibility can also introduce distractions, as indicated by the students. Students felt learning in their home environment made it challenging to remain focused on the lecture and work on their assignments because of their environment, such as family. As online learning involves accessing the internet, students may be tempted to access other websites such as social media, forums, and news. Another challenge is some students' technological barriers, such as the lack of reliable internet connection and technical issues with their devices. Students from low socioeconomic status may not have the appropriate infrastructure to access remote learning resources, disadvantaging them.

Examining responses from BVI students, challenges in online learning revolve around the accessibility of course content, accommodations from instructors, and connecting with classmates. As with other research, course materials that are not accessible are the most significant barrier to BVI students progressing in their classes, whether graphics without descriptive text or untagged PDFs, making it difficult for screen readers to navigate them. This barrier stems from instructors who may not have experience with BVI students or have no knowledge of making materials accessible. Further exacerbating the problem is when the instructors are not readily available to assist, as noted by the respondents. Coupled with the challenges in connecting with their classmates virtually, the students feel a sense of isolation, becoming independent learners and finding alternative resources to learn the concepts taught in class.

Conclusion

This chapter presents a survey study of post-secondary students' perceived readiness for online learning, sighted and blind or visually impaired. I examined how students perceived what attributes they found important to their online learning success and what attributes they feel they are most confident in taking online courses. The study shows that there are notable differences in how sighted and BVI students perceive their own competencies in learning remotely and the benefits and challenges in online learning. In the next chapter, I detail the codesign of a learning management system with BVI students and K-12 teachers with experience teaching CS to visually impaired students.

CHAPTER EIGHT

A STORY-DRIVEN PARTICIPATORY DESIGN OF AN ONLINE LEARNING MANAGEMENT SYSTEM

Introduction

In the previous chapter, I analyzed survey data from post-secondary education students who were sighted or blind, or visually impaired (BVI) to learn about their readiness for online learning and their prior experiences taking fully online courses. The results identified an area of competence where sighted students felt confident in taking online courses; their proficiency in using online learning technologies such as grade books, chat or messaging, or learning management systems (LMS). BVI students identified no such areas of confidence for taking online courses, which suggests that possible factors not covered in the survey impact their readiness for learning remotely. An area of concern mentioned by students, expressed more deeply by BVI students, was the lack of accommodations from the instructors in terms of providing learning content that is accessible and not being as available for help as traditional in-classroom instruction. If not addressed, these barriers can profoundly widen the learning gap between sighted and BVI students. However, resolving such barriers require learning about the instructor's challenges in providing the necessary accommodations for students with visual impairments. The software used for online learning may be a limitation of what instructors may be able to offer to students. Therefore, more research is necessary to examine how instructors and students view existing learning management systems in terms of usability and accessibility and to uncover how such platforms might be designed to facilitate better the teaching and learning experience in a virtual environment. This chapter details how I leverage participatory design (PD) with BVI students and K-12 teachers of visually impaired students (TVI) to provide critical input on how an LMS can be designed to be accessible for BVI learners and support teachers and create accessible course content.

Participatory Design and Storytelling

Participatory design (PD) is a common and popular method in human-computer interaction (HCI) for collecting end-user input and applying it to the design of a product or system [86]. In PD, the end-users serve as codesigners with the expertise to inform about a product or system’s design. This approach involves an interactive process, starting with a low-fidelity prototype, eventually being redesigned to a high-fidelity, highly functional system. There is evidence of using PD in educational spaces, involving teachers and students in instructional technology design [87], [88]. However, as many PD activities may be visually driven, using stimuli such as scenarios, storyboards, or prototypes may be disadvantageous for visually impaired persons. Research has shown how using stories may help end-users define the design and context of using a product or system [89]–[92]. In the literature, storytelling often uses visual cues such as storyboards to tell stories; however, stories can be presented verbally, suggesting that it may be an accessible form of PD for participants with disabilities [93].

Co-constructing Stories

Co-constructing stories is a PD technique for eliciting in-depth feedback and suggestions [94]. This method is based on the idea that users are best suited to judge a new concept and offer helpful feedback when they think about past experiences [94]. The technique uses two phases: sensitization and elaboration. During the sensitization phase, participants are introduced to an initial story that introduces the product's context and prepares them for dialogue. The designer asks questions to evoke past experiences from the participants. During the elaboration phase, the designer introduces a story with the product in question within the context introduced in the sensitization phase. At the end of the story, the designer elicits feedback from the participants. Then, the participants are tasked to envision themselves as the story's main character, thinking aloud about how they envision future experiences and aspirations of the product. The outcome of this technique is a series of co-constructed stories about the use of the product.

For example, Buskermolen and Terken [94] used co-constructing stories with a group of design students to collaboratively design a design studio concept featuring a multi-touch table and interactive wall display. The authors started with a story about a design student preparing for a group meeting, presented through a video with animations and a voice-over. When the story ended, participants imagined what might happen at the meeting between the main character and their friends. The participants were asked to draw on their past experiences attending group meetings to answer the question. Additionally, a sketch of the meeting room with a table, a whiteboard, and a flipchart was provided to draw out the situation as they were describing it. Then, participants were shown another video of the main character meeting with their friends in a meeting room with a multi-touch table and interactive wall display. When the story ended, participants were asked what they liked and disliked about the story. The authors then asked the participants to imagine themselves as the main character and what they would do in the story. The authors provided the participants several copies of a sketch of the meeting room with the multi-touch table and interactive wall display to draw on as they described their stories. From the participant stories gathered after the second story, the authors collected detailed information about the setting of the concept, the people's perceived habits, suggested methods and materials, and potential outcomes for the concept. Not only did the authors receive feedback about their concept but also suggestions for future development.

Research Questions

The following research questions guide this study:

R1: What design requirements are necessary for blind or visually impaired students to leverage an online learning platform to learn programming?

R2: What design requirements are necessary for teachers of blind or visually impaired students to leverage an online learning platform to teach programming?

Method

Participant Recruitment

I sent recruitment ads to listservs for computer science (CS) educators with instructions to fill out the sign-up form if interested. I sent recruitment adds to a listserv for blind or visually impaired developers with a similar sign-up form for interested participants to complete. For teachers to be eligible to participate, they had to have experience teaching computer science at the K-12 level to blind or visually impaired and have taught a fully online/remote course. For students to be eligible, they had to identify as blind or visually impaired, be a current or former college or university student, and have taken at least one fully online or remote course. The course must be one taught at a college or university; self-paced, standalone courses from websites such as Udacity, Udemy, or Pluralsight did not count.

Participant Demographics

There were two design groups for the research; one group for BVI students and one for K-12 TVIs. For the TVI group, there were two codesigners. One teacher identified as blind and used a screen reader as their assistive technology; the other was sighted. On average, both codesigners have taught seven blind or visually impaired students a year. On average, the participants have six years of experience teaching K-12 CS and 2.5 years teaching visually impaired students. Between the two teachers, the group has taught six fully online courses. Languages used by the group include Java, HTML/CSS, and Swift.

In the student group, two identified as blind and used screen readers and braille displays as assistive technologies. The one student identified as having low vision and uses magnification (i.e., ZoomText). Two of the students were undergraduates (one sophomore and one junior), and one student was pursuing a graduate certificate. The students have taken 42 fully online courses (mean = 14).

Study Stories

One part of conducting co-constructing stories is the actual narratives used to facilitate discussions around the design of the system. Two stories are required, one for each phase (sensitization, elaboration). I created two sets of stories, one for each group, for the design sessions. I drew on the informative literature and research I conducted on accessibility in computer science education (CSEd) to craft stories that reflect the experiences of taking an online course from the teachers' and students' point of view (POV). The first story, called the context story, introduces the context in which the proposed system will be used, setting the background for the story's main character as they prepare to use the proposed technology. The second story, called the concept story, introduces the actual technology conceptually used by the main character. This story is incomplete by intent to allow codesigners to build on the story as part of the activity to inform the system's design and functionality. Teachers' context and concept stories can be found in Appendices G and I, respectively. Students' context and concept stories can be found in Appendices F and H, respectively.

Procedure

The study procedure was similar for both groups. I conducted three design sessions with each group, scheduled bi-weekly, over six weeks. I coordinated with each group to find a common date and time to hold the sessions. All sessions were conducted over video conferencing using Zoom. I used Zoom as it is a more popular and accessible platform for people with visual impairments. All participants were familiar with using Zoom and its functionalities. Each design session lasted between 30-60 mins. Figure 8.1 provides an overview of the progression of the design sessions and development of the prototype.

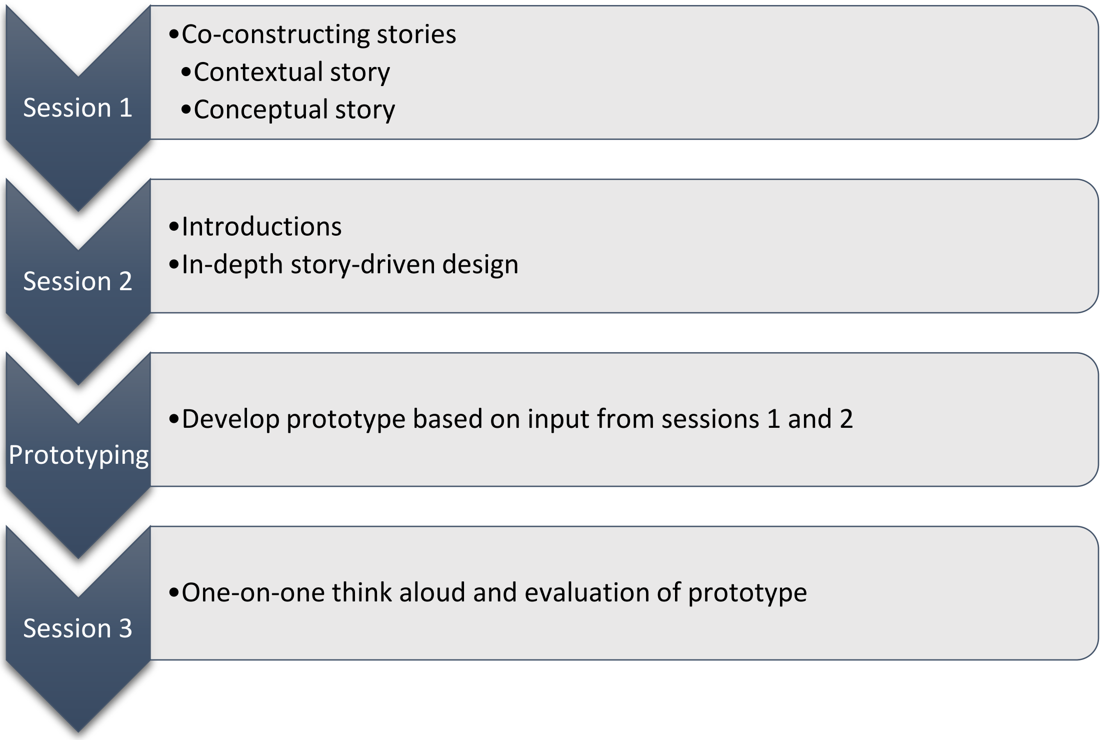

Session 1
•Co-constructing stories
•Contextual story
•Conceptual story
Session 2
•Introductions
•In-depth story-driven design
Prototyping
•Develop prototype based on input from sessions 1 and 2
Session 3
•One-on-one think aloud and evaluation of prototype

Figure 8.1: Sequential progression of participatory design methodology

Design Session One

The goal of the first design workshop was to introduce the concept of the LMS to the groups by using stories crafted to reflect each groups' experience in using existing learning platforms. I used co-constructing stories as a medium to provoke participant reflections on their teaching/learning journeys in online learning and think about early design and function suggestions. The procedure was similar for both groups. The session began with introductions and an icebreaker exercise for members to become acquainted with one another. Afterward, I read the informed consent document explaining the study's objectives, the codesigners' role in the study, and other relevant information. Upon obtaining verbal consent from the codesigners, I began the video recording of the session and explained the session's agenda and the stories to be used. I began the sensitization phase with the contextual story to introduce the context in which the proposed LMS would be used. The perspective of the story differed based on the group. After reading the story, I asked the codesigners if any part of the story resonated with their own experiences and in what ways. After the group exhausted their responses to the questions, I moved to the elaboration phase, introducing the conceptual story. The story introduces the concept of the LMS as a continuation of the contextual story. After reading the conceptual story, I asked the codesigners what they liked and did not like about the story. In the last part of the phase, I tasked the codesigners to reimagine the story with themselves as the main character and describe how the story would be similar or different from their perspectives, including the design and functionality of the LMS. At the end of the phase, I allowed codesigners to ask questions or comment on the workshop.

Findings – Teachers

After reading the contextual story, I asked teachers if any part of the story was relatable to their experiences. Both teachers expressed they related strongly with the story and talked about their first-time teaching course fully online to blind or visually impaired students. The teachers confessed to needing time to adjust to the dynamics a virtual learning environment introduced, such as making digital versions of learning content, the choice of communication tools, and available code editors suitable for web-based coding. T2, who is sighted, recalled a time when she taught a course and was not aware ahead of time that some of her students had visual impairments. It was her first time interacting with BVI students, and she did not know what questions to ask to provide accommodations. She did mention having the support of teaching assistants who helped facilitate communication between her and the BVI students.

When we talked about institutional support, the teachers had two different levels of support and resources. T1 provided his own curriculum, learning materials, and tools. He used the curriculum from Code.org for teaching HTML and CSS, using the Web Lab code editor. T1 uses the Google Suite of technology to support online learning, using Google Classroom for managing course content, Google Meet for synchronous lectures, and Google Docs for code editing. An issue T1 had with using the Web Lab editor because the screen reader could only read the code in the editor but could not edit the code. As a workaround, T1 provided the code to the students in Google Docs to edit the code and then copy and paste it into the editor. Another issue was the editor's inability to edit file names as the web page did not recognize any of the actions executed by T1's screen reader. He would often have to get a sighted person to change the file name for him. Other than the challenges described earlier, T1 appreciated the availability of the video materials provided by Code.org, and most of the course materials were reasonably accessible. On the other hand, T2 had more institutional support. T2 was provided a curriculum for the course with materials predesigned to be accessible for the BVI students. Additionally, T2 was supported by teaching assistants who would help plan the lectures and facilitate communication with BVI students. For her class, T2 used JGrasp, an integrated development environment (IDE) that automatically generates code visualizations, and JCreator, an IDE for writing code in Java, which students found accessible.

After reading the conceptual story, I asked teachers about their opinions of the story and how they would envision it to be different from being the main character. Due to internet connectivity issues, T2 could not participate in the conceptual story activity. T1 found the story relatable to his experience using similar learning platforms and working with inaccessible course materials. T1 provided some suggestions as he imagined himself in the story. He wanted to view the platform's user interface from the student's point of view. He reflected on his experience using Google Classroom in having to create a dummy student account to view the created web pages from the student's side. Another suggestion was communicating with students through the platform using a live chat or messaging feature.

Findings – Students

After reading the contextual story to the students, I asked them what aspects they could relate to. S3 did not feel he related to the story because the main character was blind, and he did not feel the challenges they faced were similar to that of that character. S1 related somewhat to the experience in the story; they reflected on their experience in a classroom that was primarily in-person but included an option for online lectures. The time he used the online option, S1 found the platform to be reasonably accessible. However, S1 mentioned that the lecture materials were not accessible.

After the conceptual story, I asked students their thoughts about the story—S1 related strongly to finding media accessible with screen readers. S1 recalls when he was taking a self-paced course in which the content the creators provided were not accessible, and he discontinued the course as a result. S1 also recalls the times during college when professors transitioned their courses from in-person to virtual but did not provide their content in a way that BVI students could use them. During the discussions, graphics, images, or other illustrations without descriptive text were the most significant issues. All three students concluded that instructors did not make much effort in providing images with text to describe them. S3 recalls having come across code as an image, which frustrated him as he could not copy and paste it into an editor to see how it worked. S3 preferred for instructors to share the actual code with the class. S2's biggest frustration was the lack of accommodations from instructors. She recalled asking an instructor for the lecture materials prior to the start of class and was denied. It made her feel like she had to work even harder than the sighted classmates to stay on par with them. They unanimously picked online learning when asked if they preferred online learning or traditional in-classroom instruction. The benefits of online learning included having access to digital content, which is more accessible than physical content, not being physically in the classroom, and easier to schedule help from tutors or instructors. The one drawback brought up was that it was more challenging to make friends in a virtual environment.

Design Session Two

I continued with the co-constructing stories for the second design workshop but focused on the conceptual story (elaboration phase). For this workshop, I reread the concept story, stopping at critical junctures in the story where the main character is using the proposed learning platform and asking questions to the group. Group members would collaboratively discuss the functions or actions in question and how they may change to best fit the needs of BVI students or the teachers. This phase will generate more concrete design and functionality requirements through the discussions.

Findings – Teachers

One of the first suggestions from the teacher group was user accounts. The teachers agreed that, at minimum, all user accounts should allow for the user's name, username, email address, password, and profile picture. There were some security and privacy considerations regarding student accounts. Students under the age of 13 need to get signed permission from parents to use the internet and web-based technologies in school. Specific personal information may not be stored on the platform—these measures are regulated by the Family Educational Rights and Privacy Act (FERPA) [95]. T2 expressed a desire for students to add other attributes that would help teachers better understand them, such as their preferred pronouns, likes/dislikes, and preferred name. Additionally, T2 wanted the ability to add notes to the student's profile, visible to only the teacher, to keep specific information about a student's needs on hand.

Another critical area of discussion was the layout and navigation of the platform. The group elaborated on several critical aspects for making each page on the platform accessible. Such considerations included legible typography, large buttons, high color contrast, placement of buttons closer to the side or bottom of the page (this is so that the buttons do not disappear when someone zooms in on the page). Customizing specific page aspects such as background color, font size, and button colors were mentioned as valuable features. The group thought multimodal output for describing page content would be helpful for students. An example would be a built-in page reader that would enable users to press a button, and the content of the page would be read by an internal text-to-speech (TTS) synthesizer, thus increasing consistency in how the content is read. T2 explains that the reader would benefit BVI students and those who suffer from learning disabilities such as Dyslexia.

The last suggestion was a code editor built into the learning platform. They express the idea of using a code editor on the platform rather than finding a separate editor. Along with the editor, the teachers mentioned some additions to support accessibility. Such features include a resizable editor, being compatible with screen readers, and highlighting errors in code.

Findings – Students

The suggestions from the students centered around the user interface (UI) and navigation of the platform. Students recalled having difficulty navigating existing LMSs such as Blackboard because of the heavy nesting of menus and buttons to traverse to find the course and related content. They stressed a simpler and cleaner UI to reduce the time it takes to find course materials. Additionally, keyboard shortcuts will help navigate a screen reader faster than cycling through the controls. S1 mention that HTML headings (i.e., <h1>, <h2>) and the appropriate leveling of headings are critical in identifying different sections of a page. HTML5 introduced semantic elements in replacement of an existing method in which web designers often create DIV (<div>) elements and apply the id attribute with the name of the element (example: <div id=”nav”>). Figure 8.2 shows the semantics elements and the hierarchical alignment in a webpage. The students expressed that using more semantic HTML elements can improve identifying page structure for a screen reader.

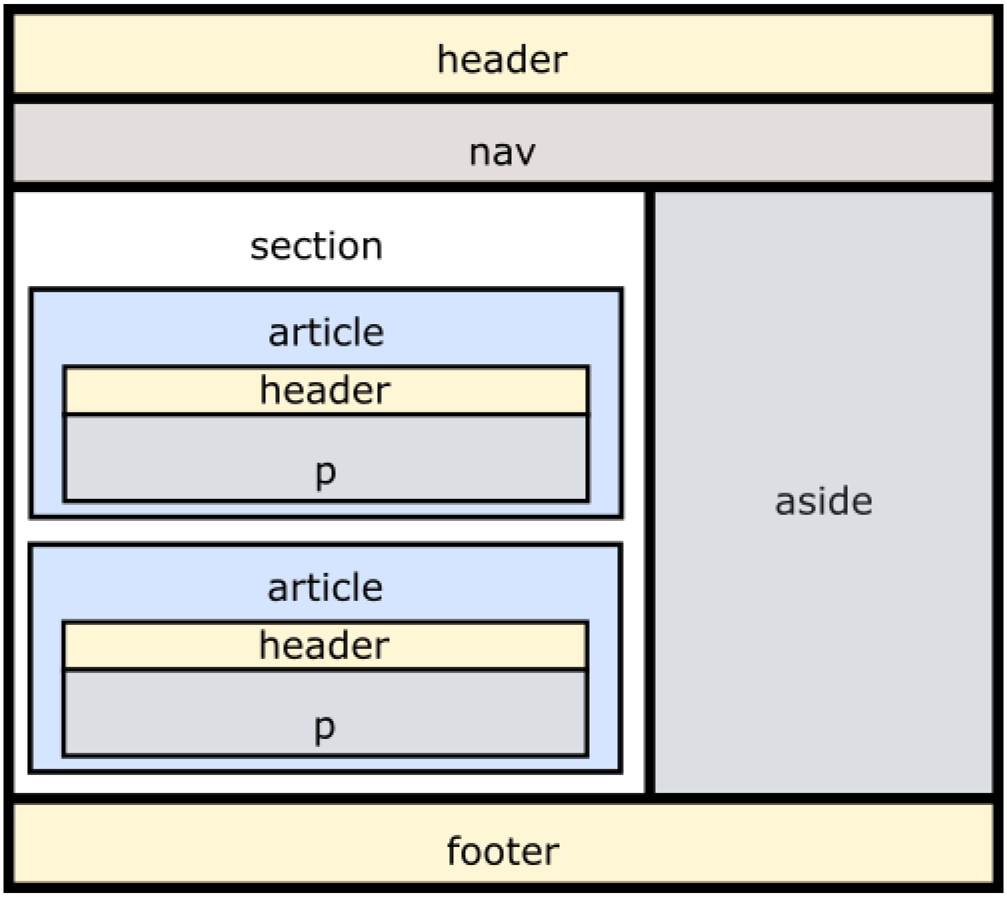


Figure 8.2: HTML5 semantic element layout

Discussions regarding lecture materials hosted on LMSs revolved around the images without text descriptions. Students mentioned it is not an issue if the lecture is primarily text-based as the text may describe the images; however, lectures containing many images may make the lecture more challenging to comprehend. Students wish instructors have the ability to add alt text to images they upload to their lectures to improve the accessibility of their content. Another complaint is the difficulty of navigation from the course home page to the lecture content, reflecting on their experience using Blackboard. The students wanted a more straightforward method to reach the lectures without having to find the correct menus or buttons. A suggestion made by S2 was to make lectures accessible through the modules as an accordion. An accordion HTML element shows the element's title with a dropdown button that, when pressed, creates a dropdown menu of items. In this context, the element title would be the module's name, and clicking on the button would create a dropdown menu containing the corresponding lectures. (see Figure 8.3). All students advocated for web applications to use semantic HTML elements to describe the structure of the web pages.

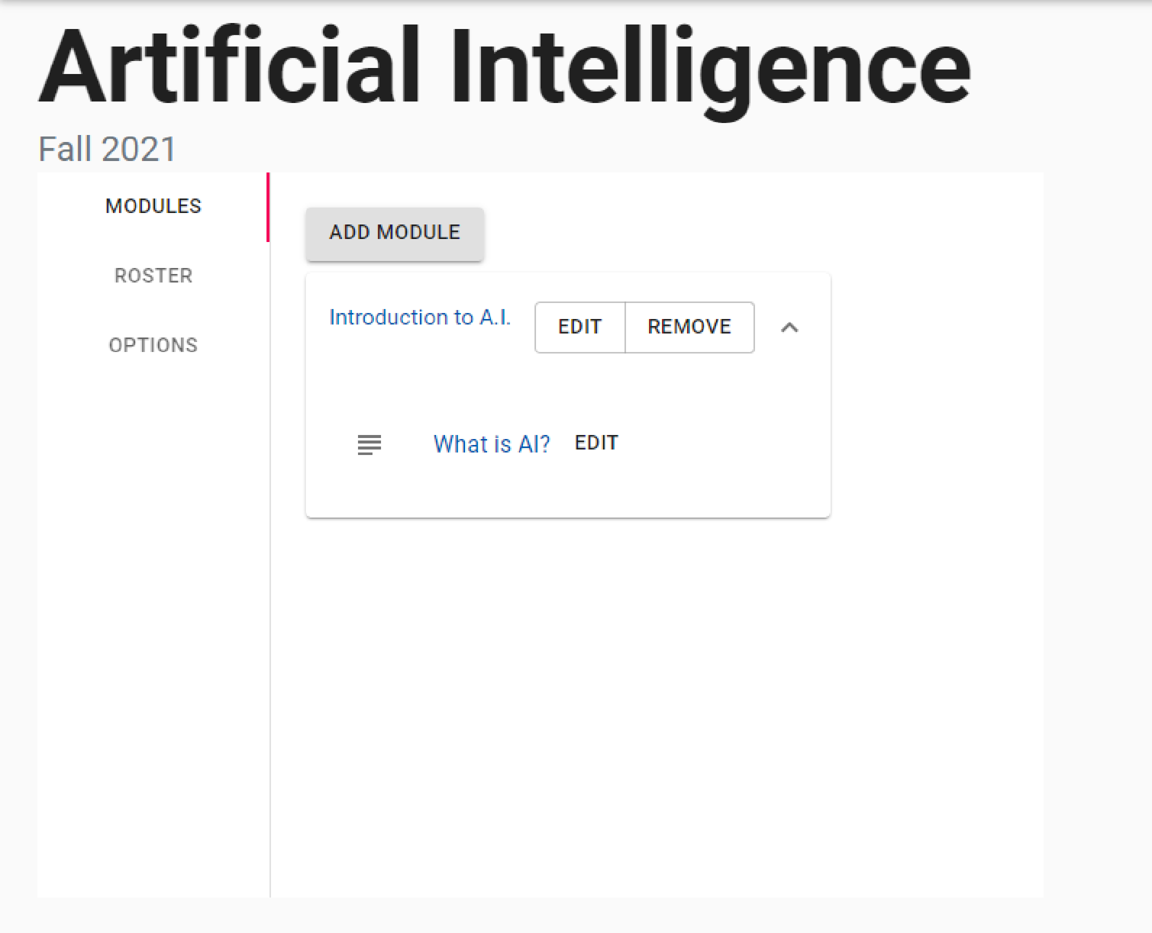


Figure 8.3: Accordion element for Introduction to A.I. module.

The students liked the idea of having a code editor built into the platform for practicing code lessons. Additions to the code editor for enhancing functionality included, but were not limited to, autocompletion, showing a list of errors along with alerts and coding hints. The most critical aspect for the editor was its compatibility with screen readers, and their assistive technologies should be able to read and edit code in the editor.

Design Session Three

Using the input from the first two design sessions, I developed a first iteration prototype of a learning management system (details of implementation in Chapter 9) for groups to evaluate in the third and final design session. For this workshop, I engaged with group members in one-on-one sessions where each participant had the opportunity to use the platform and complete tasks. Participants engaged in think-aloud sessions, speaking their thoughts about the platform's various features as they completed the assigned tasks. At the end of the session, I asked participants for final feedback regarding the LMS.

Findings – Teachers

In both sessions, teachers had trouble creating an account for the system due to the unclear error message for creating a password for the account. There was a disconnect between what users thought were the requirements for a complex password and the system's expectations. They were able to eventually create an account and enter the home page of the LMS. Teachers had to navigate the page to get a feel of the layout and UI. Both teachers found navigation to be straightforward; they liked the simplicity of the layout and minimalist design. T2 suggested making the page elements responsive to the screen's width as she noticed the main section of the home page did not adjust when moved from a small screen to a larger screen. Teachers liked adding a profile picture; however, they would like the ability to add alt text. The teachers felt that creating and managing the course roster, creating courses, modules, and lectures seemed straightforward and had little difficulty completing these tasks. T1 (blind) liked the inclusion of keyboard shortcuts for making navigating the text editor easier. T1 tried to click a hyperlink on an accordion element to go to the module home page, but the accordion dropped down the list of lectures instead. The error occurred when he was using the JAWS screen reader. When T1 switched to using the NVDA screen reader, he could click on the hyperlink to go to the module home page. Overall, both teachers found using the prototype to be straightforward and accessible. They found the ability to manage the course roster, adding courses, modules, and lectures easy and without much guidance. The group complimented the minimalist design of the UI and believed that BVI users would appreciate the more straightforward controls and menus. Teacher T1 admitted that he does not have much experience with LMS but believes it would be a useful and accessible tool for hosting course materials for students.

Findings – Students

For the student evaluation portion, student accounts were made for them to log into the platform and access courses assigned to them. Walking through the LMS home page, all the students were consistent in their thoughts that navigating the UI was easy. They found the UI simple, clean, and free of unnecessary controls. Student S2 found that the course list's layout makes traversing the course list straightforward. Student S3 suggested that the top bar on the home page remain fixed to remain at the top of the page and not scroll down with the page. Another suggestion was to allow the sidebar navigation to be collapsible. S3 expressed that screen real estate is essential for a person with low vision; having as much screen available when zooming in is desirable. Additionally, the sidebar should automatically disappear as the screen width decreases (i.e., viewing the page on a mobile device).

While navigating their user profiles, the students liked the ability to add a profile picture. Student S1 mentioned they would like more explicit labeling of buttons for uploading photos and adding alt text to their photos. The students could navigate through the course assigned to them and follow the controls and menus to find the lectures. The students were tasked with reading over the lecture, which contained an image with alt text. Additionally, they had to summarize the lecture and a description of the image. All three students were able to summarize the lecture content and image accurately. The most significant area of improvement was the built-in code editor. I tasked the students with copying and pasting provided code in the editor panes and read aloud what the preview window showed. The two blind students (S1, S2) had issues with their screen readers interpreting the code within the editors after pasting them. For unknown reasons, the screen readers could not read the code. Also, S2 mentioned it was difficult navigating between the three editors because there was no way of telling to which editor the screen reader was pointing. S2 suggested adding a button to enable the preview webpage feature rather than the preview automatically appearing after editing. Also, she suggested adding a heading to indicate the preview pane (which is located underneath the code editor). S3, who has low vision, did not run into the same issues with the code editor as he does not use a screen reader. Overall, the students were impressed and liked the prototype LMS for its simplicity, ease of use, accessibility, and code editor (even with its notable issues).

Conclusion

This study examined the experiences of K-12 teachers and blind and visually impaired students using learning management systems while taking online courses to understand sources of frustration with existing platforms. Table 8.1 summarizes the key findings from each deign session. The most significant barriers mentioned included a complex UI that makes navigation challenging for screen reader users, the lack of support for teachers, the inability to view the course content from a student’s perspective, and the lack of semantic structuring of the web pages. The latter issue pertains to web page design not aligned with the Web Content Accessibility Guidelines (WCAG) standards for making websites and applications accessible for people with disabilities. Instructors’ lack of providing course materials in accessible formats and not accommodating requests from visually impaired students were additional barriers to the learning progression of the participants.

Another goal of the study was to explore how learning platforms could be redesigned to be more accessible for BVI persons. Using co-constructing stories, a combination of participatory design and storytelling, I learned about the essential design requirements for improving usability in learning management systems. Student requirements included the improved application of HTML semantic elements for better depiction of web page structure, minimal and simple UI for easier navigation, and better use of screen space to maximize page layout for low vision users. Teacher requirements included flexibility in customizing course content layout, attaching files to lectures, viewing the course content from a student’s point of view, and communicating with students through a messaging or live chat feature on the platform. Both groups liked the idea of a practice code editor built into the platform for practicing coding lessons without downloading and installing independent coding software.

Table 8.1: Summary of key findings from each design session

Session	Key Findings
Session One	- Visually impaired students prefer online learning over in-classroom instruction - Primary challenges for visually impaired students in online learning mostly falls on accessibility of learning materials and instructor accommodations - Most LMSs have design flaws in their user interface from a navigation perspective
Session Two	- Teachers desire customizability in creating courses and course content - Teachers want to communicate with students through the LMS - Students desire minimal UI and proper use of semantic HTML in web pages - Navigation and screen real estate most critical for accessible user experience
Session Three	- Teachers and students perceived the prototype as usable and accessible - Simple and clean UI was the biggest positive of the experience - Code editor needs some fine-tuning to work better with screen readers - Teachers want to attached files to their lectures

In the next chapter, I elaborate on implementing the prototype learning management system developed and revised with input from the participatory design sessions. I detail the features of the system as well as the tools, frameworks, and libraries used in its construction.

CHAPTER NINE

THE ARCHITECTURE AND IMPLEMENTATION OF A LEARNING MANAGEMENT SYSTEM DESIGNED FOR ACCESSIBILITY

Introduction

This chapter details the architecture and implementation of the accessible learning management system (ALCMS), designed with input from the co-constructing stories participatory design study from chapter 8. The chapter discusses relevant features, functionality, and design choices.

System Overview

The ALCMS was initially designed during the storytelling-based participatory design sessions with two design teams: K-12 teachers of visually impaired students (group one) and blind or visually impaired (BVI) students (group two). Each group offered input on how the system should be designed and what features need to be included to make it accessible with assistive technologies. Table 9.1 lists the user needs and how they are addressed in the system feature/implementation. The prototype's purpose is to serve as a central hub for managing courses, the course's roster, and course content. Teachers can create an account, create courses, add students to the course, and develop their lecture content. Students can log into the system (after their teacher creates their account), navigate to their assigned courses, and view the course's modules and lectures. A simple code editor for practicing web development is available for both user groups. In the following sections, I detail the architecture behind the system and the implementation process of the available features.

Table 9.1: User need and associated system feature(s)

User Need	**System Feature**
Minimalist and accessible user interface	Integrated UI framework with accessibility support
Upload and attach files to lectures	File upload section provided for each lecture page
Customize course content for improved accessibility	Rich text editor with options to add images (with alt text), tables, links, and can edit HTML source code if needed
Built-in code editor to practice coding	A simple three-pane code editor for practicing web development
View course content from student view	UI of course lectures designed to be similar, regardless of role
Built-in page reader to support native text-to-speech functionality	Not addressed
Messaging/live chat feature for on-platform communication	Not addressed

System Architecture

The ALCMS is a web-based system designed around web-oriented principles and uses a service-oriented and web application architecture. In particular, I developed the system as a single-page application (SPA. Unlike traditional web applications in which the client (web browser) requests a new page from the web server, in a SPA, the webserver sends data to the client machine, and the client browser changes the user interface (UI) dynamically. Using asynchronous JavaScript and XML or AJAX, the client machine communicates with the webserver to request the data necessary to serve the appropriate web content based on the user's needs. Because of this behavior, a SPA can serve content with few interruptions, resulting in efficient dynamic client rendering of web pages. Figure 9.1 shows a high-level overview of the web application architecture for the ALCMS. The system consists of a client component, a server component, and a cloud component.

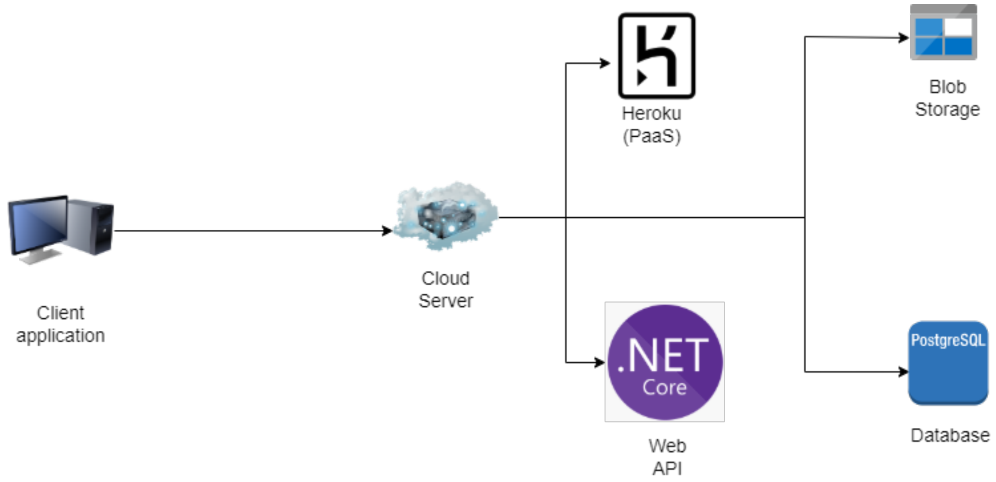


Figure 9.1: ALCMS high-level architecture

Client Component

The client-side application was designed using JavaScript, a web-based scripting language for client-side programming. For this system, I used TypeScript [96], an extension of JavaScript that adds type checking and enables static type definitions for providing additional validation of information. I found these features necessary for ensuring data from the webserver matched the type of data requested from the client. The UI for the system was designed using ReactJs [97], a JavaScript-based library that uses a component-based approach to building encapsulated, stateful UI aspects. React has a strong community of support in plugins, frameworks, and other libraries for extending application functionality. To enable quicker prototyping of the UI, I used a UI framework called Material-UI to create components more efficiently. Material-UI is a framework designed to work with ReactJs to create React components more manageable and faster [98]. Material-UI also provides a gallery of design themes users can use as a base and customize as needed. A crucial factor in choosing this framework is its accessibility support. Material-UI was developed to provide accessible components for screen-reader users and has been rated highly for meeting WCAG 2.1 standards, particularly in focus control, color contrast ratio, and WAI-ARIA compliance [99], [100]. Together, these front-end technologies form the core of the client-side architecture for a web application to be accessible with assistive technologies.

Server Component

The server component of the system serves as a back-end API service to receive and respond to requests from the client. The back-end component's core is ASP.NET CORE, an open-source web development framework for building web applications using the .NET framework [101]. ASP.NET CORE has different variants for designing applications within a specific architecture, such as the Model-View-Controller (MVC) and Web API frameworks. The system uses the ASP.NET CORE Web API to create a RESTful API with which the client will communicate fetching and serving the required data. Additionally, a suite of development packages extends the ALCMS, such as the Entity Framework for mapping domain-specific objects to relational data in databases and Identity for user authentication (login, registration) support.

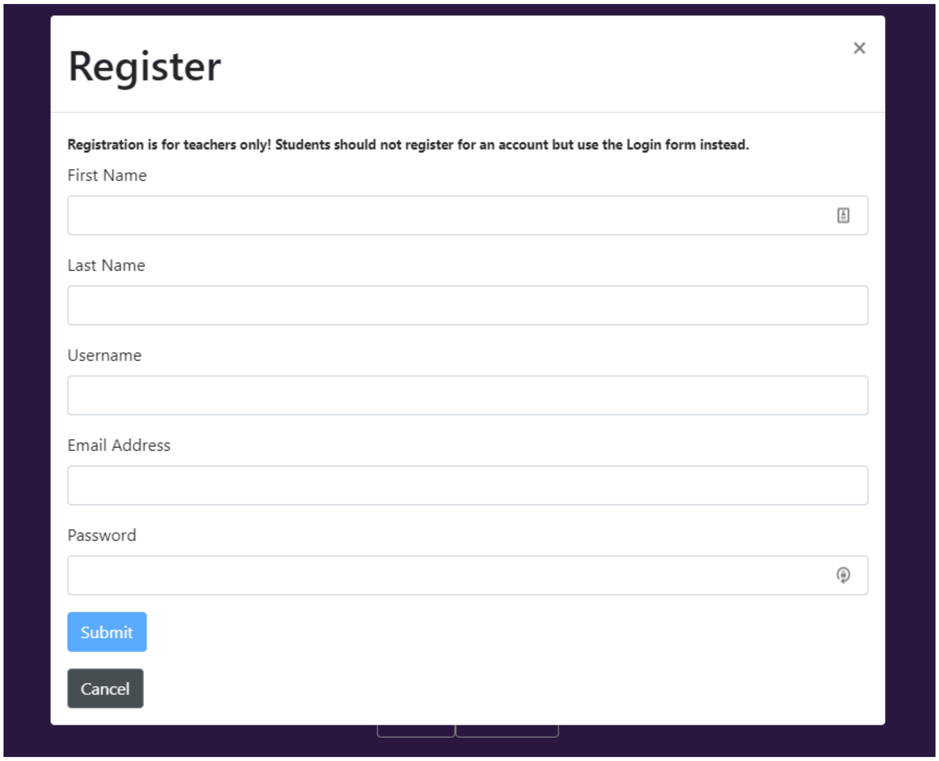
Cloud Computing

A web-based system requires a platform as a service (PaaS) for hosting, deployment, and maintenance. The ALCMS is hosted using Heroku [102] as it supports hosting services for various applications such as Python, Node, and ASP.NET CORE. Additionally, Heroku offers free database support for a myriad of databases, in my case, PostgreSQL. While Microsoft Azure offers a broader range of services and features for hosting and security of applications, Heroku offers a simpler integration process and is more cost-effective for this application. One feature used from Microsoft Azure is the Azure Blob Storage service, which enables unstructured data storage (i.e., text files or images).

Features and Functionality

Account Creation/Login

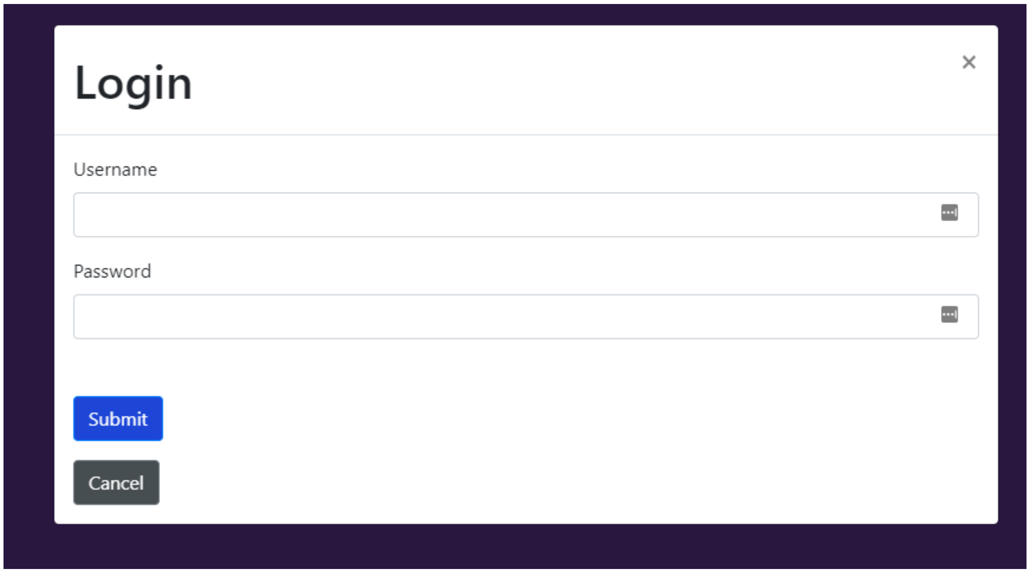
Teachers who use the system can register their accounts when they begin. By design, only teachers have the permission to create an account; student accounts are created by the teacher for the class they assign. Required information for account creation includes their name, a username, an email address, and a unique password (as shown in Figure 9.2). Teachers and students can log in to the system using their username and password (Figure 9.3). Both forms are designed to capture the screen reader's focus in use so the user can navigate through the form more quickly.



Register

Registration is for teachers only! Students should not register for an account but use the Login form instead.

Figure 9.2: User registration form for creating teacher accounts.

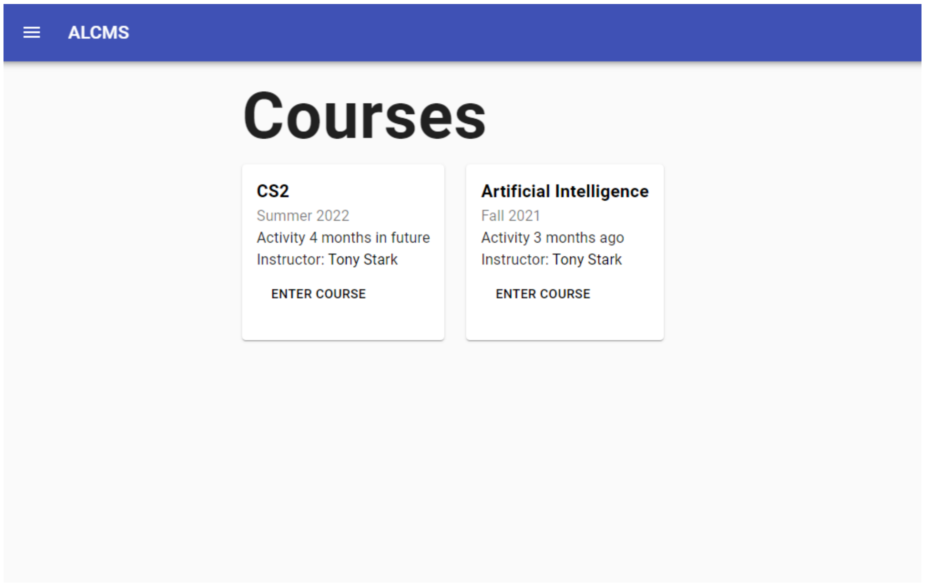


Login

Figure 9.3: Login form for the ALCMS.

Main Page

Once an account is created or the user logs in, they are taken to the home page of the LMS. The home page has two sections: the sidebar navigation and the main content area. The sidebar navigation is the primary navigation menu for the ALCMS, consisting of the user's username and several buttons for accessing different menus. The menus include the user's profile, the list of courses available to the user (shown in the content area by default), the Create Course menu (for teachers only), the code editor, and the log-out option. By default, the sidebar is expanded; however, it can be collapsed manually, as suggested by low vision students during the participatory design study, or if the screen width decreases past a specific threshold (see Figure 9.4 for the sidebar collapsed). The landing page is designed to be similar for teachers and students, except for the presence of the Create Course menu button. The Courses page shows the list of courses that the user has access to, either because the teacher created it or the student was assigned to that course.



ALCMS

Courses

CS2
Summer 2022
Activity 4 months in future
Instructor: Tony Stark
ENTER COURSE

Artificial Intelligence
Fall 2021
Activity 3 months ago
Instructor: Tony Stark
ENTER COURSE

Figure 9.4: Landing page of the ALMCS with the sidebar navigation collapsed.

User Profile

Users can view their own profile and add or change their profile picture on the user profile page. There are two tabs available: the profile tab, which shows the user's profile picture, name, grade (if a student), and role (teacher or student), and the picture upload menu for adding/editing profile pictures. Users can upload an image using the profile picture tab by browsing their computer or clicking and dragging it into the upload widget box. They can crop the image afterward and then click on upload (see Figure 9.5). Screen reader users can navigate each option since each section of the upload feature is a button with labels for identifying their role. Users can change their profile picture if they have more than one image uploaded.

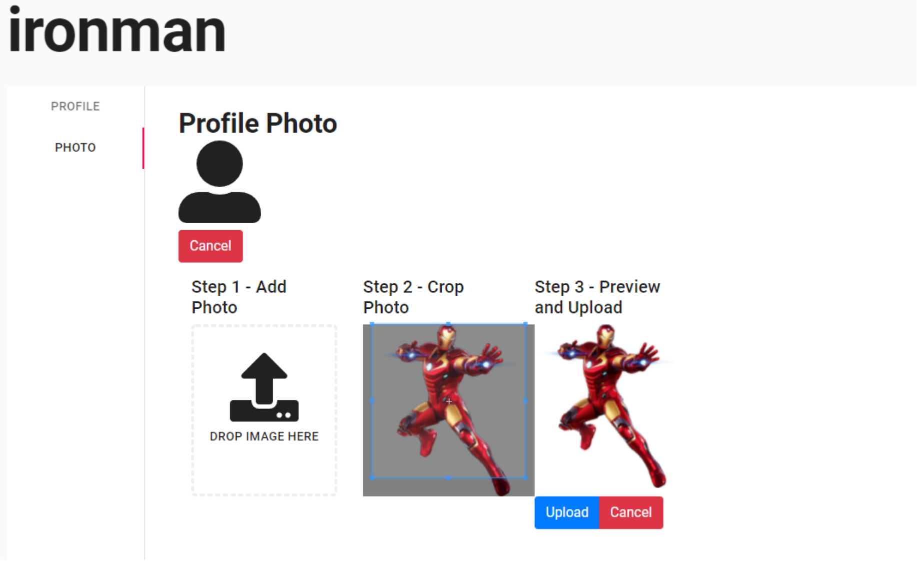
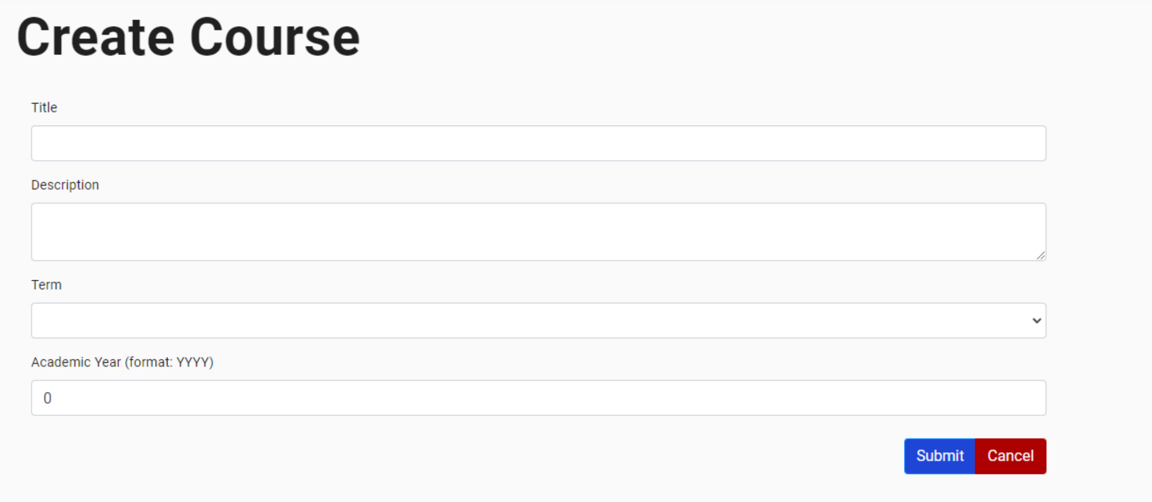


Figure 9.5: Profile picture upload widget.

Create Course

Teachers get access to create a course, as seen in Figure 9.6. Teachers provide a title, description (optional), the term (Fall, Winter, Spring, Summer), and the year. Once the course has been created, they are taken to the course home page to begin adding students to the roster and adding course content.



Create Course

Title

Description

Term

Academic Year (format: YYYY)

0

Figure 9.6: Create Course page

Course Management

Once inside the course homepage, users are greeted by two (or three if a teacher) tabs. The first tab shows the course modules, including the associated course lectures. Figure 9.7 shows the module “Introduction to AI” as an accordion element which, when expanded, shows the lecture “What is AI?” Users can click on the module link to go to the module homepage or click on the lecture in the expanded dropdown to go immediately to the lecture home page (a feature expressed by the BVI student group).

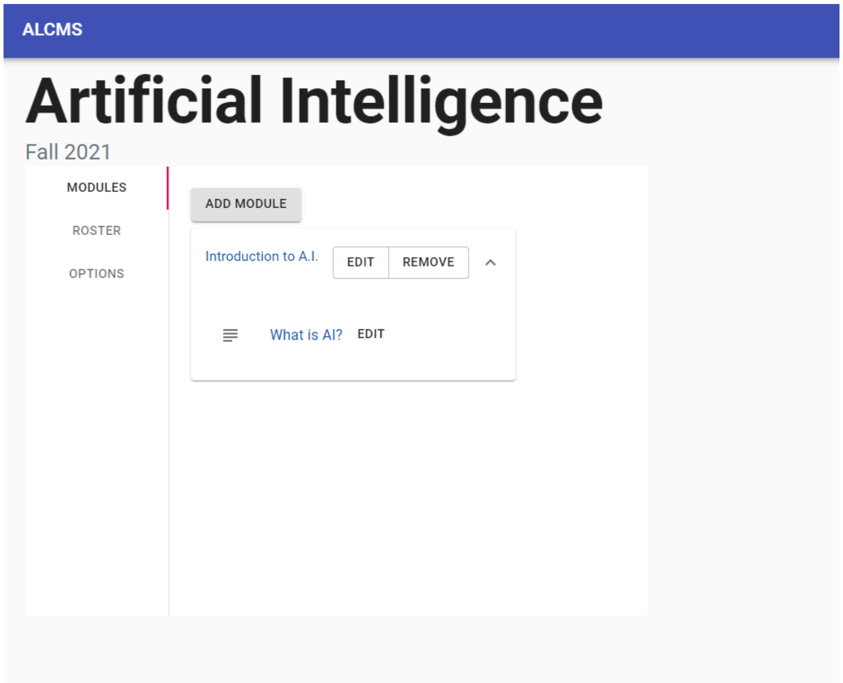


Figure 9.7: Course homepage shows an expanded module and a lecture in the dropdown menu.

The second module shows the course roster in a table format. Teachers and students can view the users assigned to the course and their profiles; however, the teachers can add students to the roster and remove them (Figure 9.8). The table was designed with the role of “table” to aid screen reader navigation per the guidelines of the Web Application Initiative – Accessible Rich Internet Applications (WAI-ARIA) [103]. The third tab, available only to teachers, is an options tab that only has the ability for the course owner to close/open a course. A closed course is not visible to students, but it is to the course owner.

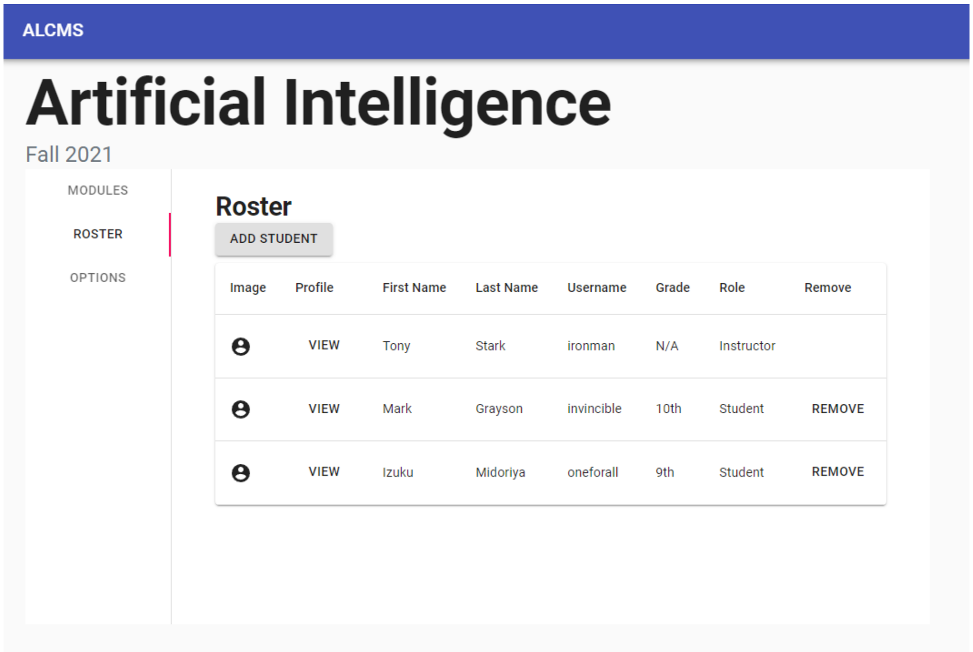
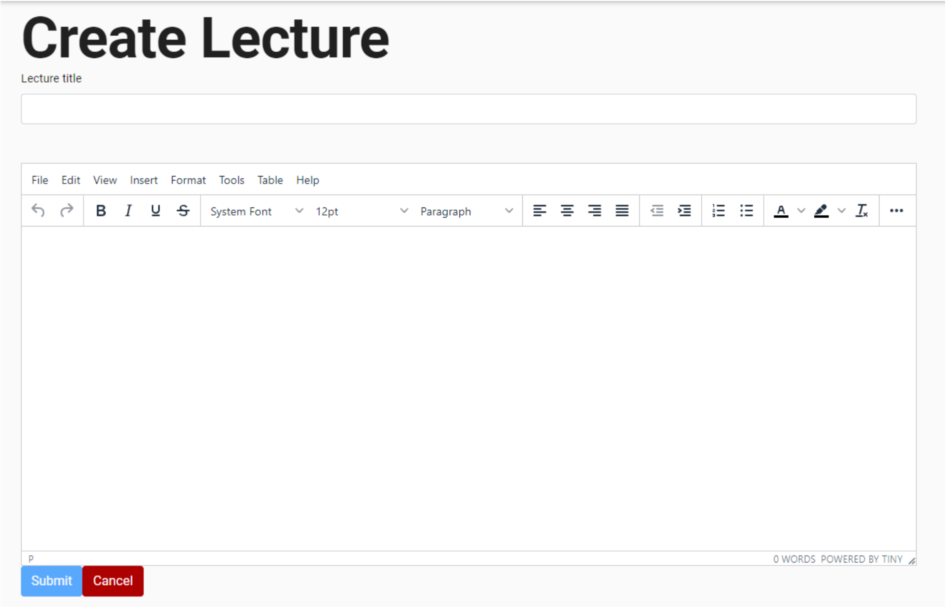


Image	Profile	First Name	Last Name	Username	Grade	Role	Remove
👤	VIEW	Tony	Stark	ironman	N/A	Instructor	
👤	VIEW	Mark	Grayson	invincible	10th	Student	REMOVE
👤	VIEW	Izuku	Midoriya	oneforall	9th	Student	REMOVE

Figure 9.8: Course roster for viewing users assigned to a course.

Teachers and students can view the lecture's content on the course lecture landing page. Teachers can create/edit the lecture content on a separate page, shown in Figure 9.9. The Create/Edit lecture page allows users to add or edit the lecture title and the content with a rich text editor, provided by TinyMCE [104]. The editor has keyboard shortcuts that allow users to use it without using their mouse. The blind teacher from the participatory design study found the feature beneficial for accessibility.

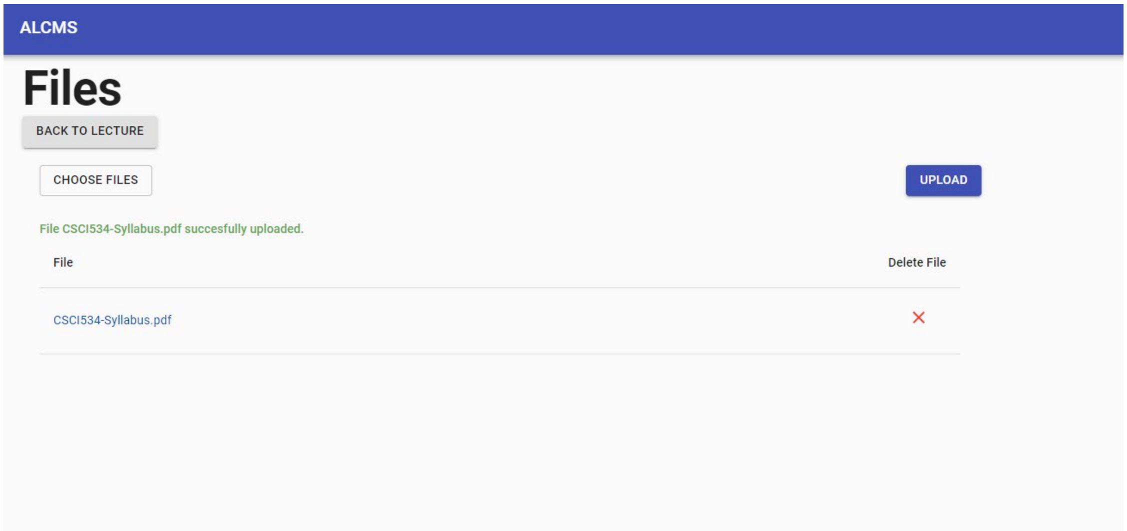


Create Lecture

Lecture title

Figure 9.9: Create lecture page with a rich text editor.

In addition to adding textual content to a lecture, teachers can upload and attach files for students to access. After creating the lecture, teachers can enter the files page and upload their attachments (as shown in Figure 9.10). Students can also enter that page to retrieve the files (but not see the controls for uploading files).



Files

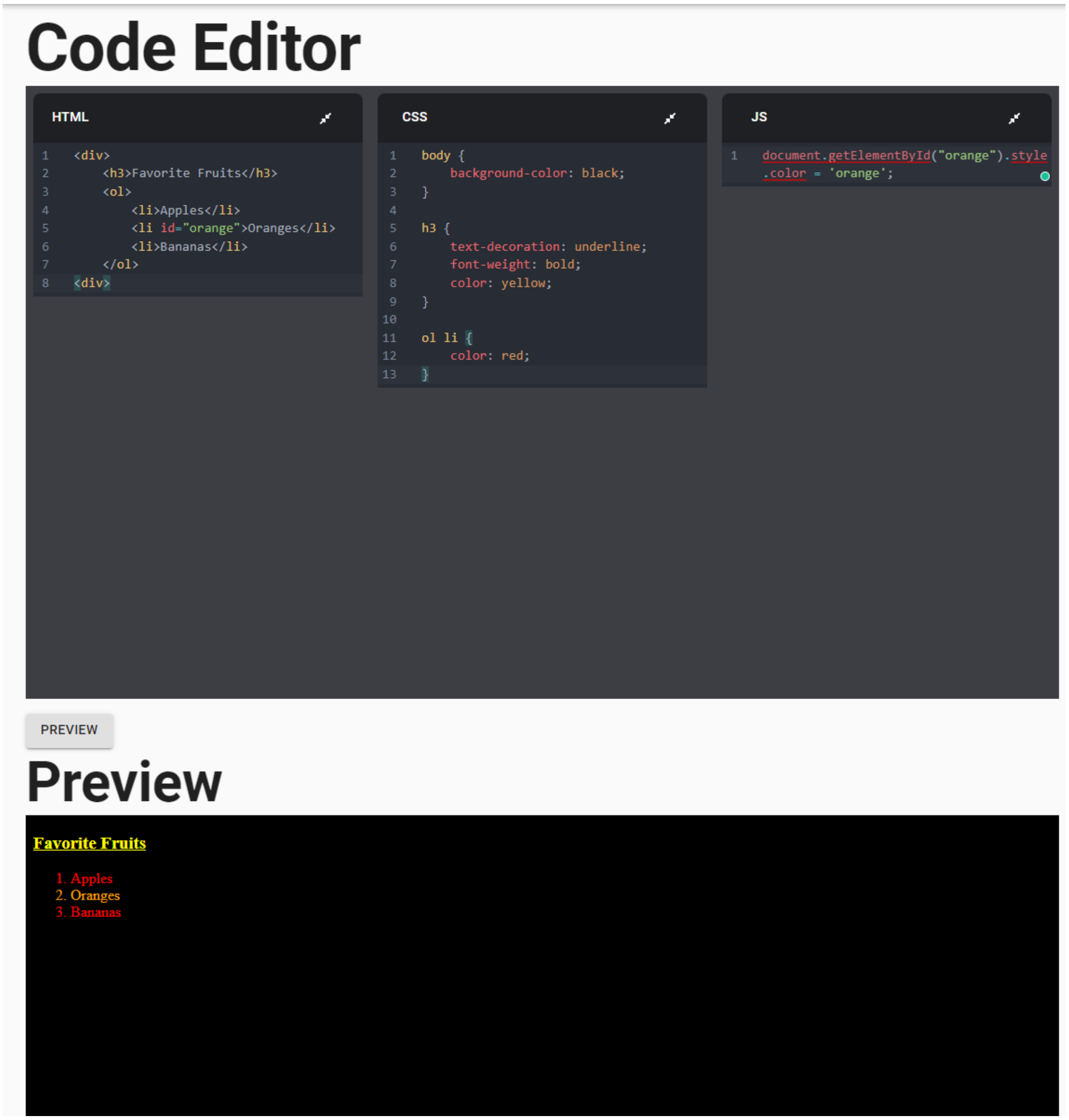
File CSCI534-Syllabus.pdf succesfully uploaded.

File	Delete File
CSCI534-Syllabus.pdf	✕

Figure 9.10: File upload page for lectures.

Code Editor

Students and teachers can use the code editor to practice basic coding concepts in web development. The editor only supports HTML, CSS, and JavaScript, three industry-standard web design and development languages. The editors were created using CodeMirror [105], an extensible code editor library that supports accessibility and touchscreen support. Each editor is a CodeMirror instance for enabling code-specific support such as syntax highlighting. In addition, the JavaScript editor has lint support for highlighting basic syntax errors. Users are not required to add code to all three editors. If a user wants to see what the resulting webpage would look like, they can click on the "Preview" button to the page in the preview pane (see Figure 9.11 for an example). As a caveat, the editor does not allow users to download code into their respective files (e.g., download .html, .css, or .js files).



Code Editor

HTML

```
<div>
    <h3>Favorite Fruits</h3>
    <ol>
        <li>Apples</li>
        <li id="orange">Oranges</li>
        <li>Bananas</li>
    </ol>
<div>
```

CSS

```
body {
    background-color: black;
}

h3 {
    text-decoration: underline;
    font-weight: bold;
    color: yellow;
}

ol li {
    color: red;
}
```

JS

```
document.getElementById("orange").style.color = 'orange';
```

Preview

Favorite Fruits

1. Apples
2. Oranges
3. Bananas

Figure 9.11: Code editor for practicing web development with a preview pane.

Conclusion

In this chapter, I detail the architecture, implementation, and functionality of the ALCMS prototype designed with input from the findings of Chapter 8. While most of the desired features were implemented, specific features, such as the live chat or messaging, were not addressed due to constraints in resources and time. In the next chapter, I discuss a study evaluating the usability and accessibility of the prototype LMS from the point of view of high school students interested in computer science, both sighted and visually impaired. The study's findings will provide insight into how students perceived the viability of the platform designed from their peers' feedback.

CHAPTER TEN

THE EVALUATION OF THE ACCESSIBLE LEARNING CONTENT MANAGEMENT SYSTEM

Introduction

Chapter eight discussed the participatory design of the accessible learning content management system (ALCMS) with K-12 teachers and blind or visually impaired (BVI) students. The teachers and students served as codesigners, offering their perspective and input on the system's critical design and interaction features. Chapter nine details the architecture and implementation of the ALCMS and the prototype's features, which addresses one or more of the user needs from chapter eight. In this chapter, I detail the evaluation of the ALCMS with high school students learning computer science. The study's goal is to assess whether students, both sighted and visually impaired, perceive the prototype system as a usable and accessible medium for accessing learning content from their courses.

Research Questions and Hypotheses

This study is guided by the research question mentioned in chapter six:

R3: In what ways does an online learning platform improve the usability and accessibility in online learning programming for students with visual impairments?

To answer the question, this study will answer the following research questions:

R1: To what extent do students perceive the usability of a learning content management system designed by their peers?

R2: To what extent would students adopt e-learning after using the learning management system designed by their peers?

Based on the research questions, the following hypotheses were developed:

H_{01} After using the ALCMS, students will not find the platform to be of “good” or better usability and accessibility.

H_{A1} After using the ALCMS, students will find the platform to be of “good” or better usability and accessibility.

H_{02} After using the ALCMS, students will not express an interest in using e-learning after using the online learning platform.

H_{A2} After using the ALCMS, students will express an interest in using e-learning after using the online learning platform.

Table 10.1 describes the relationship among the research questions, hypotheses, and measures.

Table 10.1: Relationship among research questions, hypotheses, and measures

Research Question	**Hypotheses**	**Measures**
R1: To what extent do students perceive the usability of a learning content management system designed by their peers?	H_{01}, H_{A1}	Single ease questions for tasks, system usability scale questionnaire
R2: To what extent would students adopt the learning content management system designed by their peers?	H_{02}, H_{A2}	Technology acceptance model questionnaire

Method

Participant Recruitment

Eligibility for participation included being a high school student (grades 9-12), with some experience learning computer science (learning to write code or theory). I recruited both sighted as well as BVI students for this study. I reached out to K-12 teachers through open forums online and listservs for teachers to recruit students at their institutions. I informed teachers of the study's purpose, the students' involvement, the risks and benefits, and other important information. Teachers who agreed to help recruit students were provided the study materials to review and to have them reviewed by their school's leadership (e.g., superintendent). As the population of interest are minors (under the age of 18), school leadership must review a researcher's study materials (even when approved by the researcher's institutional review board) to determine if the research is safe for their students and faculty. Once the institution's leadership approved the study, teachers began identifying students interested in participating and began sending the parent informed consent documents to their parents (it is required to receive parent permission for minor participants). After receiving permission from the parents, the students were provided with an assent document (an informed consent document for minors), informing them of the research's purpose and their role.

Participant Demographics

I recruited ten participants for the study, with seven identified as blind or visually impaired (one blind and six low vision). All of the visually impaired students have had their impairment since birth. Four (40%) of the participants are in the 11th grade, three

(30%) are in the 10th grade, 2 (20%) in the 9th grade, and one in the 12th grade. Most (six) participants were male, one female, one identified as non-binary or third gender, and two preferred not to say. Most students have less than a year of coding experience prior to the study; one student has been learning for three years. All but one student have used an online learning platform prior to the study. Table 10.2 provides detailed descriptions of the participant demographics. The literature on usability studies shows an inconsistency regarding the "ideal" sample size. Early research has shown that approximately 3-5 participants are valid for usability studies. However, more recent literature shows that a higher sample size increases the percentage of discovering problems in a system [106]. A rule of thumb mentioned by Hwang and Salvendy indicates a 10 +/- 2, depending on the type of evaluation to be conducted [107]. Macefield indicated that for quantitative usability studies, where statistical significance is sought, a group size between 8-25, with 10-12 participants as a baseline, is valid [108]. Based on calculations provided from a usability sample size calculator [109], the recommended size for this study is 13.

Table 10.2: Student demographics

Variables		**Mean/Frequency and Percentages**
Academic Classification	12th	1 (10%)
	11th	4 (40%)
	10th	3 (30%)
	9th	2 (20%)
Gender	Male	6 (60%)
	Female	1 (10%)
	Non-binary/third gender	1 (10%)
	Prefer not to say	2 (20%)
Race/Ethnicity	Asian	1 (10%)
	Black/African American	3 (30%)
	Caucasian/White	4 (40%)

	LatinX or Hispanic	1 (10%)
	Middle Eastern/North African	1 (10%)
Visual Impairment	Yes	7 (70%)
	No	3 (30%)
Used Prior LMS	Yes	9 (90%)
	No	1 (10%)

Procedure

Students who received parental permission were provided a study packet. The packet contained the child informed assent document, an instructions document containing the tasks to complete (see Table 10.3), their assigned participant ID, the links to the learning platform and evaluation survey, and a folder containing code snippets for one of the tasks. Students had to complete all the tasks using the ALCMS and complete the corresponding evaluation survey. Students had to input their participant ID to match the responses to them but also to maintain confidentiality. The evaluation survey consists of three components to assess the system's usability. Single ease questions (SEQs) are a straightforward metric to measure the perceived ease or difficulty of performing a task (see Appendix J). Users rate the task on a Likert scale of 1 (very hard) to 7 (very easy). Participants rated each task and filled out a short answer textbox with the rationale behind their rating. The next section of the survey consisted of the system usability scale (SUS) questionnaire, which measures user perceptions of system usability [110], [111]. The scale consists of ten statements on a scale of 1 to 5. For scoring, the responses are aggregated to equal a score up to 100. The last section consisted of the technology acceptance model (TAM) questionnaire, which measures the understanding behind technology acceptance and use [112]. The TAM helps explain why a user accepts or rejects information technology and posits that the user's behavioral intentions influence acceptance/rejection, attitude, perceived usefulness, and ease of the system. The TAM model developed in Park et al. 2009 was adopted for this research. The model used adds to the original TAM (perceived ease of use & perceived usefulness, attitude, behavior intention) with additional factors such as self-efficacy, subjective norm, and system accessibility. To this day, this instrument has not been used with minor participants or within a K-12 educational context. Post-evaluation interviews were conducted with the BVI students a few days after the study to learn more about their experience using the ALCMS and how it compares to existing LMSs they may have used before.

Table 10.3: Tasks

Task Identification	Task
Task 1	Log in to the system
Task 2	Navigate to and enter profile
Task 3	Change profile picture
Task 4	Navigate and enter course
Task 5	View course roster
Task 6	Navigate and find course module
Task 7	Navigate and find course lecture
Task 8	Read and summarize lecture
Task 9	Download a file from the lecture
Task 10	Use code editor to create a webpage

Results

Task Difficulty

Table 10.4 shows the means for each task, first for all students, then for sighted and visually impaired students. Observations from the data show that students scored most of the tasks between six and seven, indicating they found the tasks easy to very easy to do. Tasks three, nine, and ten were scored below six, with three and nine rated the lowest at 4.1. The BVI students scored changing profiles pictures (Task3) higher than sighted students. However, the sighted students scored downloading and reading a lecture file (Task9) higher than BVI students. In contrast, both groups scored similarly using the code editor (Task10).

Examining the rationale behind the ratings for task three, some BVI students indicated they had previous experience with uploading profile pictures. In contrast, others who did not have prior experience gave it a low score because they found it challenging. Other BVI students had technical problems with the task, such as the system not accepting their photos and finding workarounds to complete the task. These technical challenges were echoed by the sighted students who scored the task low. For task nine, the rationale behind the low scores from BVI students was that they found the PDF file inaccessible and had trouble reading it. Most of the students did not have trouble downloading the file. In contrast, sighted students did not encounter any issues with the task. For Task ten, both groups found it reasonably straightforward to use the code editor to make a webpage. One student misinterpreted the instructions and used the wrong file to insert code into the editor; hence, he scored low on this task.

Table 10.4: Means of difficulty of tasks performed (1 = Very hard, 7 = Very easy)

Task	**Mean (All)**	**Mean (Sighted)**	**Mean (Visually Impaired)**
Task 1	6.6	6	6.86
Task 2	6.9	7	6.86
Task 3	4.1	3.67	4.29
Task 4	6.5	6	6.71
Task 5	6.2	6	6.29
Task 6	6.4	6	6.57
Task 7	6.5	6.33	6.57
Task 8	6	5.67	6.14
Task 9	4.1	6.67	3
Task 10	5.5	5.33	5.57

System Usability Scale

Table 10.5 shows the total SUS score for each participant. The mean score was 62.8; the average SUS score, in general, is 68 [113]. The score indicates marginal usability, but there are areas for further improvement. The mean score for sighted students was 76.3, while the mean score for BVI students was 57. The findings suggest that sighted students found the ALCMS more usable than BVI students.

Table 10.5: SUS scores of students

Student	Vision Status	SUS Score
S1	Sighted	83
S2	Visually impaired	80
S4	Sighted	58
S5	Sighted	88
S8	Visually impaired	43
S9	Visually impaired	45
S10	Visually impaired	43
S12	Visually impaired	48
S13	Visually impaired	70
S15	Visually impaired	70
Mean (SE)		62.8(5.56)

Perceived Acceptance of E-Learning

Table 10.6 shows all students' means for each construct and then broken down by sighted and visually impaired students. Observing the mean scores, all students rated each construct between 3 and 4.2, indicating neither agree nor disagree nor somewhat agree. Students were generally positive in perceived ease of use, their attitudes toward e-learning, and the self-efficacy in e-learning while being indifferent to their perceived usefulness and system accessibility. BVI students were more positive in perceived usefulness and attitudes toward e-learning than sighted students, but there was no statistical significance.

Table 10.6: Mean scores of technology acceptance model constructs (1 = Strongly disagree, 5 = Strongly agree)

TAM Construct	**Mean (All)**	**Mean (Sighted)**	**Mean (Visually Impaired)**
Perceived ease of use (PE)	4.16	4.2	4.14
Perceived usefulness (PU)	3.59	2.87	3.9
Attitude (AT)	4	3.9	4.1
e-learning self-efficacy (SE)	4.2	4.33	4.14
System accessibility (SA)	3.3	3.6	3.14

Reflections on Using the ALCMS

I conducted interviews with the blind and visually impaired students shortly after they completed the evaluation of the ALCMS to gather more in-depth information regarding their experience using it. Using qualitative analysis, I solicited common trends from their responses. Students were asked about their initial thoughts before using the system, how their opinions changed after using it, what they liked or disliked about it, areas of improvement, and how the system compared to existing learning management systems they used.

Most students were curious about the prospect of the ALCMS. S8 believed it would be similar to CodeHS [114], an LMS he was familiar with using. S12 was initially concerned about navigating the platform but found it to be a more effortless experience than initially believed:

> *"First, I thought that it will be a hard to manage to like navigate, but when I just went into it and I logged in to the system, I found is easier than what I thought." – S12*

Asking about the positives of the ALCMS, all the students mentioned how easy it was to use. S2 mentioned how it was "intuitive" and "snappy," that the bigger font makes it someone like him, who has low vision, to see the content better. S12, how is blind, found it to be accessible with the Voiceover screen reader:

> *"And it got like everything is accessible voiceover on the homepage and in login the profile thing like" – S12*

S13 liked the built-in code editor as it allowed her to try out programming in languages she had not tried, as she is relatively new.

> *"Is kind of fun to look at different ways, or how to code different ones." – S13*

When asked about things they did not like when using the ALCMS, there were only a few challenges students came across. The biggest challenges were changing the profile picture and the code editor:

> *"One was like I had a issue with like trying to make the profile picture, it was very difficult for me to do and I had to get some help trying to do that" – S15*

Students mentioned problems uploading their photos to the ALCMS, indicating a technical issue with the system. Some students mentioned that the code was not working with the code editor unless placed in a particular order. S13 mentioned that the print was too small to read the content comfortably.

When I asked students about areas of the ALCMS that could be improved, the only suggestion was making the font size customizable so people with varying levels of vision can adjust it for their comfort level:

> *"… customizing the size of the font as far as like changing some people like changing the color to where they can be focused on reading stuff." – S10*

Most of the students have used another LMS prior to the study. I asked their experiences with them compared to using the ALCMS. S2 used Project STEM [115], which supports STEM curricula for K-12, including computer science, and he felt the user interface for ALMCS was superior:

> *"Well okay see I use project stem which I have to be honest, I don't really like so compared to that your ui was a lot more like it was a lot more user friendly… the way they organize their assignments, and the actual modules for instruction, if you want to go to a text you have to go back to the homepage, which is quite annoying and that's the main challenge." – S2*

Compared to CodeHS, most students found the ALCMS to be more accessible and easier to use. S9 felt it was "a whole lot much easier" to use. S12 felt CodeHS and Code.org were not accessible and believes the ALCMS is a more accessible alternative. S13 and S15 felt that CodeHS was slightly better than the ALCMS but for different reasons. S13 was more interested in the learning content and would have liked to see more activities and exercises included in the ALCMS. S15 felt navigation was slightly more straightforward than the ALCMS. To note, students have been using CodeHS longer than the ALCMS; therefore, they may have more familiarity using CodeHS's LMS.

I asked students if, in its current state, they would recommend the ALCMS to other blind or visually impaired students. All students expressed that they would recommend the system to other students because of its accessibility:

> *"Yes, yes, definitely um it's very it's very, even if, like if you are like i'm very severely visually impaired and I found it to be pretty easy to read everything and like find what I needed to find so if someone was as visually impaired as me and they needed to learn about computer science I would definitely recommend your program" – S2*
>
> *"Yes I recommend if anybody want to learn online…" – S12*

Discussion

Perceptions of Usability

The study sought to answer two research questions about user perceptions of the ALCMS. The first question focused on the usability of the ALCMS from the perspectives of students:

R1: To what extent do students perceive the usability of a learning content management system designed by their peers?

The task analysis showed that students were able to complete the tasks and were generally positive about the ease of the task. Only tasks three, nine, and ten posed some measure of difficulty. Findings from the SUS measure showed the students, on average, found the system to be of marginal (between "ok" and "good") usability. Despite the mixed scores from the SUS measure, students generally had a positive experience with the system; therefore, I reject the null hypothesis (H_{01}) and accept the alternative hypothesis (H_{A1}) that students find the platform to be of "good" or better usability and accessibility. The results show that blind or visually impaired students can perform most of the same tasks as well as the sighted students using the ALCMS with the design features provided from the co-design workshops. Despite the challenges, BVI students found the code editor usable and completed the assigned task. The BVI students found the editor useful for practicing while learning to code. The simplicity of the editor ties back to my exploratory research of how BVI persons prefer editors with a simplistic user interface as they are more straightforward to navigate with screen readers. The task of downloading and reading the file from the lecture revealed a continued pattern of an area of concern for BVI persons. The low score for the task was due primarily to the inaccessibility of the PDF file that they had to summarize. No BVI student mentioned having difficulty downloading the file itself; hence, from a technical perspective, it was not a design flaw of the ALCMS but rather the file used for the lecture.

Perceptions of Accepting E-Learning

The second question focused on students' perception of using e-learning:

R2: To what extent would students adopt e-learning after using the learning management system designed by their peers?

From the analysis of the TAM constructs, the students were generally positive with using e-learning technology, their self-efficacy with e-learning, and their attitudes. For perceived usefulness and accessibility, students were indifferent. BVI students rated higher usefulness than sighted students, aligning with my exploratory work and the participatory design study in chapter eight. BVI students prefer online instruction over in-classroom. Hence, I reject the null hypothesis (H_{02}) and accept the alternative that students are likely to express a desire to accept e-learning. The results suggest that both groups are equally capable of and willing to use e-learning.

Limitations

Several limitations impacted the study and its findings. The most significant is the low sample size (N = 10). The study was designed to be a usability pilot to assess the functionality and features of the prototype, for which the sample size was acceptable. However, this means that the results are not generalizable for all users who may use such a system. The primary challenge in recruitment was the availability of blind or visually impaired high school students learning computer science. As research suggests, compared to sighted high school students, there is a gross disproportion of BVI students actively engaged in computer science education (CSEd). Finding and earning buy-in with teachers for sighted and visually impaired students was another challenge. Teachers are often predisposed with teaching their courses and may not express an interest to be involved in the research as it may be seen as “additional work.” Given the protocols in place due to the COVID-19 pandemic, recruitment and interaction took place virtually. This makes engagement and procuring trust from participants a challenge. Students, especially minors, may be less trusting of someone they cannot engage in person.

Another limitation is the facilitation of the activities for the study. Teachers of student participants aided in distributing and instructing students on completing the task. However, this may present a possible disconnect in communications of expectations of the study. One instance was when one student did not use the correct file containing the code snippets for the code editor task. Also, teachers often had to remind students to complete the tasks and evaluations. There were some drop-offs of participants due to the perceived complexity of the tasks, even without having done at least one task. In reflection, a study of this nature would ideally be conducted in a lab-based environment or at a school within a classroom to ensure proper execution.

Conclusion

This study assessed the viability of a prototype learning management system for sighted and visually impaired students. Students completed a series of assigned tasks using the system and evaluated it with a survey measuring ease of use, overall system usability, and perception of accepting e-learning. Overall, students were generally positive with the system, found the tasks to be easy to complete, and were likely to use e-learning in the future. The analysis also showed that both sighted and visually impaired students were similar in their perceptions of usability, ease of use, and willingness to accept e-learning, further demonstrating how the ALCMS can be designed to create a learning environment usable by both user groups. In the next chapter, I summarize my dissertation research, its broader impact, and areas of future research.

CHAPTER ELEVEN

CONCLUSION

Introduction

My dissertation was the result of informative research that revealed technological and educational barriers that negatively impact the learning experiences of blind and visually impaired (BVI) persons in computing education (CEd). The most critical challenges encountered included existing learning materials that are not designed to be accessible for users of assistive technologies such as screen readers or magnification; therefore, the onus falls on the instructors to provide the necessary modifications. A second challenge is the code development tools such as integrated development environments (IDE) that, while are sophisticated software, have complex user interfaces and do not work well with screen readers. Such barriers create a gap in learning progression between sighted and BVI students and potentially dissuade BVI students from pursuing a career in the tech industry. However, in all three studies, I discovered how e-learning could be leveraged to improve accessibility in computing education. It offers advantages for BVI students to fully engage in the learning. To this end, this research aimed to understand and discover how e-learning can be designed to be a more accessible learning experience for BVI students. A survey study revealed how sighted, and visually impaired students perceived themselves learning in online environments;

particularly, BVI students were indifferent to their ability to take online courses. This led to a participatory design study with BVI students and K-12 teachers to learn how learning management systems (LMS) should be designed to be accessible for assistive technologies and better facilitate the teaching and learning processes in online spaces. This led to the development of the accessible learning content management system (ALCMS), which was evaluated by sighted and visually impaired high school students to assess its usability and accessibility. Findings revealed that students liked the system and found it straightforward to accomplish common tasks done on existing learning platforms. Furthermore, post-study interviews showed that BVI students felt the ALCMS was more accessible than previous LMS they have used in the past.

Contributions of The Dissertation

My dissertation has arguably contributed one of the first studies exploring accessibility in online learning technologies to support blind or visually impaired K-12 computing education. Further, it is arguably among the first studies to examine how user-centered design can be leveraged to improve accessibility in online learning environments for people with disabilities. My dissertation provided evidence in improving the science of engaging BVI persons in technology design through non-visual design methodologies such as storytelling. Some of the major contributions from this dissertation are summarize in Table 11.1.

Table 11.1 Major contributions of dissertation

Contribution	Contributing Chapters
Showing how accessibility barriers in K-12 education persists in professional developer settings	3,4
Advantages of visual impairment in programming and learning	3,4
Identifying preferences/advantages/disadvantages in online learning for BVI learners	3,4,7,8
Identified barriers/challenges for K-12 of BVI students in teaching CS	4,8
Design and interaction preferences for developing accessible learning management systems	8
Using storytelling as design activity for blind or visually impaired co-designers	8
Designing an accessible learning management system	9
Using the Technology Acceptance Model within a K-12 context	10
Demonstrating perceptions of usability and accessibility of a learning management system using user-centered design	10

The study in Chapter Three describes how professional programmers with visual impairments manage their disability in the workplace and how they see their line of work could be improved. The interviews suggest that they often use technologies that are not accessible, and they look for alternatives to continue to contribute to their work. Specific tasks or activities such as pair programming and designing graphical user interfaces (GUIs) were deemed not accessible, and, hence they do not engage in them. Programmers desire some assistance with using developer tools such as manuals or tutorials spoken from the perspective of a developer with a visual impairment. I also asked about their learning experiences during their journey towards their career. The findings align with existing research. Most of their classroom experiences were fraught with inaccessible learning content and editors, inconsistent accommodations from teachers, and having to look for electronic resources to supplement their learning. A unique finding was the perceived advantages of having a visual impairment. Advantages included having greater focus, memorizing the codebase, and performing specific tasks better than sighted programmers. The contributions of this student were accepted and published in the proceedings of the *ACM International Conference on Software Maintenance and Evolution* in 2020.

The study in Chapter Four details interviews with K-12 teachers with experience teaching computer science to BVI students. The goal is to learn more about the challenges teachers face in providing instruction to students and what they perceive as the most significant barriers their students face in computing education. The findings show that teachers find most of the computer science (CS) curricula do not support blind or visually impaired students and, therefore, often modify them or create new materials for their students. Teachers often provide a variety of formats of their materials specific to blind or low-vision students. An additional challenge was the lack of institutional support (i.e., software licenses, new hardware, and devices), limiting what they could teach. Like the study in Chapter Three, teachers mention that visually impaired students developed better mental models of the codebase and possessed greater attention to detail when reading code. Lastly, teachers wanted additional opportunities for professional development to train them on how to teach computer science, especially to visually impaired students. The contributions of this work were accepted and published in the proceedings of the *ACM Technical Symposium on Computer Science Education* in 2021.

In Chapter Five, I conducted a diary study to collect data about the teaching and learning experiences in a K-12 CS course with a teacher and two BVI students. Teachers and students submitted answers to a question prompt as diary entries after every lecture, providing information regarding the lecture, the materials, the languages and tools used, and the perceived difficulty of the assignments. From the students' diaries, they could complete all the assigned work and expressed little challenges to their learning over the four weeks of the study. Students felt the teacher's instruction proved helpful, and the course's online nature was accessible. The teacher's entries revealed they believed the students comprehended the topics with few challenges and could complete the assignments. As in the previous studies, the course was taught online and provided an accessible learning experience for the students. The contributions were accepted and published in the proceedings of the *Annual Meeting of the Human Factors and Ergonomics Society* in 2021.

The study discussed in Chapter 7 was designed to explore perceptions of readiness for online learning of sighted and visually impaired students and discover perceived advantages, disadvantages, and challenges in online learning unique to BVI persons. Students took an online survey asking questions pertaining to competencies for online learning, which they rated across perceived importance and confidence. The survey also consisted of open-ended questions asking students what they believe are the advantages and disadvantages of online learning and, for BVI students, specific challenges they face in online learning environments. The findings show that sighted students are the most confident in using online learning technologies such as LMS. However, BVI students did not feel more confident than or view any competency as more important than others. While students find online learning convenient and provide flexibility in their schedules, they admit it is difficult to make friends and connect with their classmates online. When watching online lectures, they are more tempted to be distracted by environmental factors. For BVI students, inaccessible course content is a significant barrier, even in online courses and feeling isolated from their class. This research contributes to an understanding of the perceived readiness for taking online courses for students with visual impairments and will be submitted for review in a venue focused on e-learning.

In Chapter Eight, the study was designed around using participatory design (PD) with two goals in mind: 1) learn more about teachers and students' past experiences using learning management systems and 2) uncover the design and interaction preferences of BVI students and teachers to make an accessible and usable LMS for computing education. Because my participants have a visual impairment, I used a variant of PD called co-constructing stories that uses storytelling to solicit design considerations from the co-designers as they think about their past experiences and how they would imagine such a system would function. This was the first study to use co-constructing stories with BVI participants to inform the design of an online learning environment. The study serves as evidence for the HCI community for how we may reconceptualize participatory design and other design methodologies to support the participation of people with disabilities. The contributions of the study will be submitted for review to a venue focused on accessibility and computing.

The usability study in Chapter Ten is the first study to evaluate an online learning environment designed from the input of BVI end-users, particularly in computing education. In particular, it is the first study to explore the accessibility of online learning environments in computing education with K-12 students. The findings show that sighted and visually impaired students could complete the assigned tasks and perceived them as easy to very easy. The BVI students found the code editor a welcomed addition and found it reasonably accessible when completing the coding tasks. The biggest praise was the simplicity and minimalist design of the user interface, making navigation trivial, something that plagues existing systems. While some usability challenges were encountered, students were generally positive overall and saw the ALCMS be a more accessible alternative to existing platforms such as CodeHS and Code.Org. Also, students felt e-learning is something they could thrive in after using the ALCMS. This study provides insights into how a user-centered design approach can lead to a more usable and accessible learning experience in online environments for sighted and visually impaired students. The study's findings will be submitted for review to a venue in either human-computer interaction or accessible computing.

Future Work

My dissertation has laid the foundation for exploring equity, inclusion, and accessibility in online learning. While my dissertation demonstrated how society could reimagine learning platforms to be accessible for persons with visual impairment, other areas lack sufficient research to make online learning environments more accessible and usable for marginalized populations. A logical next step would be to explore barriers faced by learners with other disabilities (i.e., hearing, cognitive, motor) and what solutions can adequately address them. One approach would be exploring machine learning techniques to develop adaptive interactions with learning platforms. The learning platform can learn from user interactions, learner attributes, and preferences for user interface display. As remote learning increases in prevalence and provides more opportunities for mentoring, education, and community building, it is important to understand additional challenges other than technology that may limit or deny full participation of persons from low socioeconomic status (SES) and people with disabilities. Additional research is required to explore what societal, environmental, and human factors contribute to the widening digital divide of users and differential access to online learning opportunities.

Final Remarks

My dissertation documents my journey towards exploring ways to improve accessibility in K-12 computing education for blind or visually impaired learners. Through my research, I have demonstrated how a human-centered approach to design can positively impact the user experience of using online learning environments for persons with visual impairments. The COVID-19 pandemic caused academia, industry, government, and other venture to shift to remote learning/work. This model has shown to be effective for maintaining productivity while providing convenience and flexibility. As such a model continues to be used, especially in education, society must ensure that all learners have equitable access to resources to succeed in online education. My dissertation has provided the foundation for how society may provide a better learning experience in online environments, especially for people with disabilities.

APPENDICES

Appendix A

Study 1 Interview Questions

Work Environment

What company do you work for?

Do you work alone or as part of a team?

If part of a team, what are the ways that you work well with your team?

If part of a team, what are some of the challenges you face when working with your team?

What issues do you face at work?

Assistive Technologies

What assistive technologies, if any, do you use to perform your job?

> In what ways do you use these technologies to help you perform your job?
>
> What challenges, if any, do you face when using these technologies?

Language and Tools

What language(s) have you found to be the easiest to use given your visual impairment?

What software tool(s) have you found to be the easiest to use given your visual impairment?

What language(s) have you found to be the most challenging to use given your visual impairment?

What software tool(s) have you found to be the most challenging to use given your visual impairment?

Development Style and Challenges

What programming tasks do you face the biggest challenges?

Explain these challenges

What workarounds do you use to address some or all of these challenges?

What programming tasks do you find fairly easy to do?

Do you think your disability gives you an advantage over sighted developers in some way?

Do you think there are some tasks you do better than other sighted developers?

Education

What made you go into programming?

Do you go to school to learn to program? If yes, what kind of school?

Describe your experience learning how to code

What language(s) were you learning?

What software tool(s) were you learning?

What aspects of learning to code seemed easier to you?

What were some of the challenges you faced when learning to code?

What tools and strategies did you use to learn how to code?

Areas of Improvement

What could be improved to make programming tools more accessible and usable for people with visual disabilities?

In what ways could learning to code be made easier for people with visual disabilities (software, educational curriculum, etc)?

Appendix B

Study 2 Interview Questions

How many years of teaching experience?

How many years of experience teaching students with visual impairments?

Where do you currently teach? Is it a mainstream school or a school for the blind or visually impaired?

Have you taught at either a mainstream/school for the visually blind?

What grade(s) have you taught?

What course(s) have you taught?

What programming language(s) have you taught in your classes? Why did you choose those languages?

What tool(s) are you using/have you taught for your classes? Why did you choose those tools?

What topic(s) do you teach for your courses?

Do you notice if the blind or visually impaired set up their physical environments differently from the sighted students?

What assistive technologies do the students use when working on an assignment?

Are there any accommodations you provide to students with visual impairments?

How do you present the lecture material to students with visual impairments?

How are the assignments presented to students with visual impairments?

What topics from your courses have students with visual impairments perform well in or have little difficulty in learning? Why?

What topics from your courses have students with visual impairments struggled with or have difficulty in learning? Why?

What programming tasks have you noticed students with visual impairments doing well with?

What programming tasks have you noticed students with visual impairments struggling with?

From your experience working with students with visual impairments, what are some things they are able to do without assistance?

What challenges have you noticed the students faced as they use the software tools during the lesson?

What workarounds/strategies have you noticed students using to overcome these challenges?

What assistance do you offer to those students who face these challenges?

Are there things that students with visual impairments do better than sighted students?

What are your biggest concerns with these students learning to code?

What are some ways we can improve software tools like editors and IDEs to improve their accessibility for students with visual impairments?

What are resources that teachers need to better help teach students with visual impairments how to code?

Is there anything else that you would like to add or share?

Appendix C

Study 3 Student Diary Study Questions

PID:__________________________

Date:__________________________

Name of course:_____________________________

What programming languages did you use today? (type N/A if no languages were used)

What programming tools did you use today? (type N/A if no tools were used)

What did you do today?

Did you have a lecture today? Yes No

If yes, what was today's lecture about?

What did you learn from today's lecture?

What did you find easy to understand from today's lecture? Why?

What did you find challenging to understand from today's lecture? Why?

Was there any way today's lecture could have been improved? If yes, how?

Did you have an assignment to do today? Yes No

What is the goal of the assignment?

What parts of the assignment was easy for you to do? Why?

What parts of the assignment was challenging for you to do? Why?

Did your visual impairment affect your ability to work on your assignment in any way? If yes, how?

Was there any way the assignment could have been improved? If yes, how?

On a scale of 1 to 5, 1 being poor and 5 being excellent, please rate the following, if applicable.

The format the lecture materials were presented was: 1 2 3 4 5

The format the assignment was presented was: 1 2 3 4 5

On a scale of 1 to 5, 1 being challenging and 5 being easy, please rate the following, if applicable.

I found using the programming language for today's lecture/assignment to be:
1 2 3 4 5

I found using the programming tools for today's lecture/assignment to be:
1 2 3 4 5

Overall, I found the assignment to be:
1 2 3 4 5

Overall, I found today's lecture to be:
1 2 3 4 5

Appendix D

Study 3 Teacher Diary Study Questions

Date:____________________________

Course:__________________________

What programming languages did you use today? (type N/A if no languages were used)

What programming tools did you use today? (type N/A if no tools were used)

Describe the purpose of today's lecture?

How were the lecture materials presented to the students?

Were any in-class assignments given to the students during the lecture? If so, what were they?

What aspects of the lecture did the student(s) find easy? Why?

What aspects of the lecture did the student(s) find challenging? Why?

What aspects of the in-class assignments did the student(s) find easy? Why?

What aspects of the in-class assignments did the student(s) find challenging? Why?

Did you assign any homework assignments this past week? Yes No

What aspects of the homework assignment did the student(s) find easy? Why?

What aspects of the homework assignment did the student(s) find challenging? Why?

On a scale of 1 to 5, 1 being easy and 5 being challenging, please rate the following:

The student(s) found the in-class assignment(s) to be: 1 2 3 4 5

The student(s) found the homework assignment to be:1 2 3 4 5

The student(s) found today's learning materials to be:1 2 3 4 5

The student(s) found the concepts of today's lecture to be:
1 2 3 4 5

Appendix E

Readiness for Online Learning Survey

Demographics

What is your age? ____________

What is you gender?

[ ] Man

[ ] Woman

[ ] Non-binary

[ ] Prefer not to say

[ ] Prefer to self-describe: ____________

What is your ethnicity?

[ ] Asian

[ ] American Indian or Alaska native

[ ] Black/African American

[ ] Hispanic or LatinX

[ ] Middle Eastern or North African

[ ] Native Hawaiian or Other Pacific Islander

[ ] White

[ ] Mixed Race

As of the completion of this survey, what is your current academic standing?

[ ] Freshman

[ ] Sophomore

[ ] Junior

[ ] Senior

[ ] Postbac

[ ] Grad Certificate

[ ] Masters

[ ] Doctoral

[ ] Postdoc

Have you ever taken a fully online course?

[ ] Yes

[ ] No

How many fully online courses have you taken, including now?

____

What is/was your field study?

____

Do you identify as someone with a visual impairment?

[ ] Yes

[ ] No

START - Questions based on answering ‘Yes’ to being visually impaired

Low vision is characterized as having a visual acuity of between 20/70 and 20/400 in the better-seeing eye with conventional corrections such as glasses or contact lenses or a visual field of 20 degrees or less. Blindness is characterized as having a visual acuity of 20/400 or worse in the better-seeing eye or a visual field of 10 degrees or less. Based on these definitions, how would you define your degree of impairment?

[ ] Low vision
[ ] Blindness

What assistive technologies do you use when taking online courses (select all that apply)?

[ ] Screen reader
[ ] Magnification
[ ] Braille display
[ ] Other: ______________
[ ] I do not use any assistive technologies

END

<u>Open-ended questions</u>

What advantage(s), if any, do you believe online learning offer you? Text box

What disadvantage(s), if any, have you come across in online learning? Text box

If any, what are the biggest challenges you have faced in your online learning?

> **How have these challenges impacted your learning experience?**
> Text area

What synchronous communication tools have you used for your online courses (e.g., Zoom, Microsoft Teams, Google Hangout, etc.)?

What asynchronous communication tools have you used for your online courses (e.g., discussion boards, emails, etc.)?

What learning management systems (e.g., Moodle, Canvas, Blackboard) have you used in your online courses?

Have you experienced any technology-related challenges in your online course(s) (e.g., Zoom, Adobe Connect, Moodle, Canvas)?

[ ] Yes
[ ] No

> **Please describe these challenges**
> Text area

START – Questions related to respondents who answered ‘Yes’ to being visually impaired

In what ways, if any, has your visual impairment impacted your online learning experience?

Did you request any accommodations for any of your online courses?

[ ] Yes
[ ] No

> **Were your accommodations met? Why or why not?**
> Text area

END

Student Readiness in Online Learning Scale

Importance

All statements measured on 5-point Likert scale: 1 = not important at all, 2 = unimportant, 3 = neither important or unimportant, 4 = somewhat important, 5 = very important

Online Student Attributes

Rate how important these competencies are for you in your online learning.

- Set goals with deadlines.
- Be self-disciplined with studies.
- Learn from a variety of formats (lectures, videos, podcasts, online discussion/conferencing).
- Be capable of following instructions in various formats (written, video, audio, etc.)
- Utilize additional resources to answer course-related questions (course content, assignments, etc.)

Time Management

Rate how important these competencies are for you in your online learning.

- Devote hours per week regularly for the online class.
- Stay on task and avoid distractions while studying.
- Utilize course schedule for due dates.
- Complete course activities/assignments on time.
- Meeting multiple deadlines for course activities

Communication

Rate how important these competencies are for you in your online learning.

- Use asynchronous technologies (discussion boards, email, etc.)
- Use synchronous technologies (Webex, Collaborate, Adobe Connect, Zoom, etc.) to communicate.
- Ask the instructor for help via email, discussion board, or chat.
- Ask classmates for support (accessing the course, clarification on a topic).
- Discuss feedback received (assignments, quizzes, discussion, etc.) with the instructor.

Technical Competence

Rate how important these competencies are for you in your online learning.

- Complete basic computer operations (e.g., creating and editing documents, managing files, and folders).
- Navigate through the course in Learning Management Systems (e.g., Moodle, Canvas, Blackboard, etc.).
- Participate in course activities (discussions, quizzes, assignments synchronous sessions).
- Access the online grade book for feedback on performance.

- Access online help desk/tech support for assistance.

Confidence
All statements measured on 5-point Likert scale: 1 = very unconfident, 2 = somewhat unconfident, 3 = neither confident or unconfident, 4 = somewhat confident, 5 = very confident

Online Student Attributes
Rate your confidence in your ability to accomplish the following competencies in your online learning.

- Set goals with deadlines.
- Be self-disciplined with studies.
- Learn from a variety of formats (lectures, videos, podcasts, online discussion/conferencing).
- Be capable of following instructions in various formats (written, video, audio, etc.)
- Utilize additional resources to answer course-related questions (course content, assignments, etc.)

Time Management
Rate your confidence in your ability to accomplish the following competencies in your online learning.

- Devote hours per week regularly for the online class.
- Stay on task and avoid distractions while studying.
- Utilize course schedule for due dates.
- Complete course activities/assignments on time.
- Meeting multiple deadlines for course activities

Communication
Rate your confidence in your ability to accomplish the following competencies in your online learning.

- Use asynchronous technologies (discussion boards, email, etc.)
- Use synchronous technologies (Webex, Collaborate, Adobe Connect, Zoom, etc.) to communicate.
- Ask the instructor for help via email, discussion board, or chat.
- Ask classmates for support (accessing the course, clarification on a topic).
- Discuss feedback received (assignments, quizzes, discussion, etc.) with the instructor.

Technical Competence
Rate your confidence in your ability to accomplish the following competencies in your online learning.

- Complete basic computer operations (e.g., creating and editing documents, managing files, and folders).
- Navigate through the course in Learning Management Systems (e.g., Moodle, Canvas, Blackboard, etc.).
- Participate in course activities (discussions, quizzes, assignments synchronous sessions).
- Access the online grade book for feedback on performance.
- Access online help desk/tech support for assistance.

Appendix F

Participatory Design Student Contextual Story

Introduction

For today’s session, I will present two stories. The first story will introduce the context in which the imagined platform will be used. The imagined platform will be mentioned but will not be described in this first story. I will ask you some reflection questions regarding the story upon conclusion. I will then have you think about potential future actions taken by the main character during this story.

I will then read the second story that introduces and describes the imagined platform within the context. Again, I will ask you some questions regarding the story when finished. You will then be asked to develop some alternative designs and use cases for this platform by putting yourself in the main character’s perspective.

Story

This is the story of a student, blind since birth, attending a school for students with visual impairments. The student is proficient in using the VoiceOver screen reader on their Macbook as well as their iPhone. The student hears about an upcoming course on programming focused on web development, where HTML, CSS, and JavaScript will be taught. Excited about the prospect of learning to code, the student enrolls. Due to social distancing policies for the upcoming year, the course will be conducted in a distance-learning format. The student, who has never taken an online course before, expresses some concerns about distance learning. Some concerns include the design of the materials, the accessibility of the electronic tools, the platform used to practice coding, among other things.

Questions:

1. What aspect(s) of this story relates to your experience?
2. What concerns or considerations would be a factor given what you learned from the story?
3. Put yourself in the role of the student. What are the next steps you would take?

Appendix G

Participatory Design Teacher Contextual Story

Introduction

For today's session, I will present two stories. The first story will introduce the context in which the imagined platform will be used. The imagined platform will be mentioned but will not be described in this first story. I will ask you some reflection questions regarding the story upon conclusion. I will then have you think about potential future actions that the main character may take during this story.

I will then read the second story that introduces and describes the imagined platform within the context. Again, I will ask you some questions regarding the story when finished. You will then be asked to develop some alternative designs and use cases for this platform by putting yourself in the main character's perspective.

Story

This is a story of a teacher who works as a STEM instructor at a school serving students with visual impairments. The teacher has a bachelor's degree in mathematics and did a minor in computer science. For the upcoming semester, the teacher is assigned to teach a course in programming. The teacher decides to focus on front-end web programming using HTML, CSS, and JavaScript. Due to new social-distancing policies for the upcoming school year, the course will be distance learning. Students will need to access the lectures and materials electronically. Further, the students will need access to a web-based code editor to practice the coding lessons throughout the course. There are six students in the class with a wide range of visual acuity and proficiency in using a computer and their assistive technology. Three students have low vision and mostly use magnification. The other three students are legally blind, two of whom use a screen-reader. One uses a screen-reader and a refreshable braille display. The teacher decides it is best to an online learning platform to host the lessons, lecture materials. In addition, the teacher needs to find a web-based code editor accessible for assistive technologies such as screen readers. The challenge the teacher faces is the printed materials created for an in-person course. Now, the teacher must convert the materials from physical to electronic form to make them available on the platform.

Questions:

1. What aspect(s) of this story relates to your experience?
2. How do you feel this teacher may begin preparing their printed materials for their course's new distance-learning format?
3. What concerns or considerations would be a factor given what you learned from the story?
4. Put yourself in the role of the teacher. What are the next steps you would take?

Appendix H

Participatory Design Student Conceptual Story

Story

At the start of the course, the student is granted access to a learning content platform for the web development course. Using VoiceOver, the student navigates to the input fields for entering their username and password, then moves to and select Login to be taken to the platform's homepage. Once there, the student, starting at the website title, begins to navigate the top menu, hearing the button options as they move through the page until they reach the course list pane, where the course they land on is Web Development Foundations. The student navigates the course information, hearing the name of the course, semester and year, and the name of the instructor before arriving at a button the reads "Enter Course." The student selects the button and is taken to the course homepage. The student moves through the course page until they hear the screen reader speak out, "file, course syllabus." They click on the link, which downloads the PDF file of the course syllabus. They switch from the web browser to the PDF file to read over the content. The student switches back to the browser to continue navigating the learning platform. The student comes across a button that says, "Module 1 – History of the Web." The student clicks on the button to be taken to the module homepage. On the page, they navigate to the button reading "Lecture 1: What is the Web?" Clicking on the link and navigating to the page, they move through the lecture content. Going through the content, the student notices the screen reader coming across images but no description, making it nearly impossible for them to understand what the image represents. The student becomes frustrated and leaves the lecture page. The student finds another button for a lecture entitled "Lecture: Overview of Web Technologies." The student clicks the button and follows the lecture, navigating through the content. Towards the end of the lecture, a link reads "Exercise: Using HTML." Curious, the student follows the link to a page where it introduces an activity for creating an HTML page. Navigating through the content, the student comes across the header of a code editor window that reads "HTML." There is a built-in code editor where the student can add the HTML code provided in the activity instructions to create an HTML page. The student copies the HTML code provided in the instruction, navigates back to the editor window, and pastes the content. The instruction reads to click on the button named "Run," found on the editor's right-hand side, and can view the resulting HTML page on the review pane underneath the code editor. The student navigates to the Run button and selects it, then moves to the preview pane, the screen reader speaking out the only piece of text that appears on the web page "Hello World." The student, intrigued at the thought of using an online editor as opposed to a desktop-based integrated development environment, feels more confident and optimistic about the course for the semester.

Questions:

1. What are your thoughts about this story?
2. What did you like/dislike about the story?
3. What aspect(s) of this story relates to your experience?

4. Put yourself in the role of the student. Thinking aloud, how would this story happen from your perspective?

Appendix I

Participatory Design Teacher Conceptual Story

Story
As the teacher prepares their distance-learning course, they find a learning platform with the necessary functionality for the class. The platform is free to use for courses for up to 25 people. The teacher creates an account, providing an email address, a secured password, a username, and a profile picture. The teacher is taken to the home page, which is empty given that no courses have been created. There is a big green button with a plus symbol in the middle with a label reading "Add a course." The teacher selects the button and is taken to the Course Creation page. The teacher provides a name for the course, the season, and the year of the course. The teacher then selects the Create Course button to complete course creation. They are then taken to the Add Students page, where they can create student accounts. The teacher presses the plus sign to add a row for a new student slot, providing their name, grade level, email address, and password. After all the students have been entered, the teacher selects the Add button to finish the student roster. They are taken to the course home page, displaying the course's name, semester of creation, and instructor's name. There are three buttons on the left side of the page; the one on top reads, "Add module," the button in the middle says, "Add lecture," and the bottom button reads "Roster." The teacher selects the Add Module button, which takes them to the Module Creation page. The teacher provides a title for the module and selects "Create." Within the module home page, the teacher selects the option for creating a lecture. The teacher finds they can write out their content on the lecture creation page using a built-in Rich Text editor. Using the editor, the teacher translates their printed materials onto the editor, adding images with alt-text along the way. After adding the content, the teacher selects the Save button to save their work in the lecture. The teacher finds an option to upload files on the same page. The teacher clicks on the Upload file button, and a prompt to upload files appears. The teacher selects some PDF files on their computer and selects the Upload button. After all the files have loaded, the teacher views the finished lecture, with the text and images added and the name of the uploaded files on the lecture pane's right-hand side with links to the actual file. Additionally, there is a button on the left to bring up the built-in code editor. They click on the button, and the web editor pops up from the bottom. The editor features three separate editors, one for each web language (HTML, CSS, JavaScript) and a larger pane for previewing the web page created from the code. After viewing the student-centered lecture page, the teacher selects the Return to Teacher View button to go back to the lecture page's teacher side. Feeling satisfied with the content, the teacher selects the course home page and adds another lecture.

Questions:

1. What are your thoughts about this story?
2. What did you like/dislike about the story?
3. What aspect(s) of this story relates to your experience?

4. Put yourself in the role of the teacher. Thinking aloud, how would this story happen from your perspective?

Appendix J

Student LMS Evaluation Survey

Participant ID: __________________
Date: _____________________

Task Assessment

For each of the following tasks, please rate the difficulty of the task from 1 (Very difficult) to 7 (Very easy). Please provide detailed reasoning behind your rating.

Log into system
1 2 3 4 5 6 7
What contributed to your rating?

Navigate to and enter your profile page
1 2 3 4 5 6 7
What contributed to your rating?

Change your profile picture
1 2 3 4 5 6 7
What contributed to your rating?

Navigate to and enter the course
1 2 3 4 5 6 7
What was the name of the course?

What contributed to your rating?

Navigate to the course roster and view the profile information of the course roster
1 2 3 4 5 6 7
Provide two names of the students in the course and the name of the instructor

What contributed to your rating?

Navigate to and enter a course module
1 2 3 4 5 6 7
What was the title of the module?

What contributed to your rating?

Navigate to and enter a course lecture
1 2 3 4 5 6 7
What was the title of the lecture?

What contributed to your rating?

Read through the lecture

1 2 3 4 5 6 7

Provide a summary of the lecture and a short description of the images.

What contributed to your rating?

System Usability Scale

All items scored from 1 (Strongly disagree) to 5 (Strongly agree)

1. I think that I would like to use this system frequently.
2. I found the system complex.
3. I thought the system was easy to use.
4. I think that I would need help to be able to use this system.
5. I found the various features in this system were well integrated.
6. I thought there was too much inconsistency in this system.
7. I would imagine that most people would learn to use this system very quickly.
8. I found the system very clumsy to use.
9. I felt very confident using the system.
10. I needed to learn a lot of things before I could get going with this system.

Technology Acceptance Model Survey

All items scored from 1 (Strongly disagree) to 7 (Strongly agree)

I. Perceived ease of use (PE)
   a. I find e-learning easy to use.
   b. Learning how to use an e-learning system is easy for me.
   c. It is easy to become skillful at using an e-learning system.

II. Perceived usefulness (PU)
   a. E-learning would improve my learning performance.
   b. E-learning would increase academic productivity.
   c. E-learning could make it easier to study course content.

III. Attitude (AT)
   a. Studying through e-learning is a good idea.
   b. Studying through e-learning is a wise idea.

c. I am positive toward e-learning.

IV. E-learning self-efficacy (SE)

a. I feel confident finding information in the e-learning system.

b. I have the necessary skills for using an e-learning system.

V. System accessibility (SA)

a. I have no difficulty accessing and using an e-learning system in the university.

Demographics

Current grade level:

- 9th
- 10th
- 11th
- 12th

Gender:

- Man
- Woman
- Non-binary/third gender
- Prefer not to say
- Prefer to self-describe

Race/Ethnic Group

- Asian
- Black/African-American
- Caucasian
- LatinX or Hispanic
- Middle Eastern/North African
- Native American/Alaskan Native
- Native Hawaiian or Other Pacific Islander
- Mixed Race

Do you identify as having a visual impairment (blindness or low vision)?

- Yes
- No

[For users who answered 'Yes' to the previous question]

Would you say you have low vision or blindness?

- Low vision
- Blindness

How long have you identified as having a visually impairment?

____

How long have you been learning to write code? (enter 0 if you have never learned to code prior to this year)

____

REFERENCES

[1] Bureau of Labor Statistics, “Computer and Information Technology Occupations : Occupational Outlook Handbook: : U.S. Bureau of Labor Statistics,” Aug. 16, 2019. https://www.bls.gov/ooh/computer-and-information-technology/home.htm (accessed Aug. 16, 2019).

[2] ACT, “Six-Figure Tech Salaries: Creating the Next Developer Workforce.” http://www.arcgis.com/apps/MapJournal/index.html?appid=b1c59eaadfd945a68a59724a59dbf7b1 (accessed Jan. 14, 2020).

[3] Code.org, “Learn computer science. Change the world.,” *Code.org*. https://code.org/ (accessed Jan. 14, 2020).

[4] A. D. Rayome, “CIO Jury: 83% of CIOs struggle to find tech talent,” *TechRepublic*, 2017. https://www.techrepublic.com/article/cio-jury-83-of-cios-struggle-to-find-tech-talent/ (accessed Jan. 14, 2020).

[5] S. Harrison, “Five Years of Tech Diversity Reports—and Little Progress,” *Wired*, 2019. Accessed: Jan. 14, 2020. [Online]. Available: https://www.wired.com/story/five-years-tech-diversity-reports-little-progress/

[6] CSforALL, “Home Page,” *CSforALL*. https://www.csforall.org/ (accessed Jan. 14, 2020).

[7] National Federation of the Blind, “Blindness Statistics,” *Blindness Statistics | Federation of the Blind*, 2019. https://nfb.org/resources/blindness-statistics

[8] J. L. Cmar and M. McDonnall, “Blind people have increased opportunities, but employers’ perceptions are still a barrier,” *The Conversation*, Oct. 14, 2019. http://theconversation.com/blind-people-have-increased-opportunities-but-employers-perceptions-are-still-a-barrier-124977 (accessed Jan. 29, 2021).

[9] E. C. Bell and N. M. Mino, “Employment Outcomes for Blind and Visually Impaired Adults,” *J. Blind. Innov. Res.*, vol. 5, no. 2, 2015, Accessed: Jan. 29, 2021. [Online]. Available: https://www.nfb.org/images/nfb/publications/jbir/jbir15/jbir050202.html

[10] Freedom Scientific, “JAWS,” *JAWS – Freedom Scientific*. https://www.freedomscientific.com/products/software/jaws/ (accessed Aug. 27, 2019).

[11] NV Access, “About NVDA,” *NV Access*, Jul. 05, 2017. https://www.nvaccess.org/about-nvda/ (accessed Aug. 27, 2019).

[12] “Vision Accessibility - Mac,” *Apple*. https://www.apple.com/accessibility/mac/vision/ (accessed Aug. 27, 2019).

[13] K. Albusays and S. Ludi, “Eliciting Programming Challenges Faced by Developers with Visual Impairments: Exploratory Study,” in *Proceedings of the 9th International Workshop on Cooperative and Human Aspects of Software Engineering*, New York, NY, USA, 2016, pp. 82–85. doi: 10.1145/2897586.2897616.

[14] C. M. Baker, C. L. Bennett, and R. E. Ladner, “Educational Experiences of Blind Programmers,” in *Proceedings of the 50th ACM Technical Symposium on*

Computer Science Education, New York, NY, USA, 2019, pp. 759–765. doi: 10.1145/3287324.3287410.

[15] A. Mandal, “What is visual impairment?,” *News-Medical.net*, Jun. 27, 2012. https://www.news-medical.net/health/What-is-visual-impairment.aspx (accessed Feb. 03, 2020).

[16] Centers for Disease Control and Prevention, “Common Eye Disorders | Basics | VHI | CDC,” Feb. 20, 2019. https://www.cdc.gov/visionhealth/basics/ced/index.html (accessed Feb. 03, 2020).

[17] World Health Organization, “Vision impairment and blindness,” *Blindness and vision impairment*, Oct. 08, 2019. https://www.who.int/news-room/fact-sheets/detail/blindness-and-visual-impairment (accessed Feb. 03, 2020).

[18] MedlinePlus, “Refractive Errors.” https://medlineplus.gov/refractiveerrors.html (accessed Feb. 03, 2020).

[19] K. Boyd, “What Is Macular Degeneration?,” *American Academy of Ophthalmology*, May 10, 2019. https://www.aao.org/eye-health/diseases/amd-macular-degeneration (accessed Feb. 03, 2020).

[20] MedlinePlus, “Cataract.” https://medlineplus.gov/cataract.html (accessed Feb. 03, 2020).

[21] K. Boyd, “What Is Glaucoma?,” *American Academy of Ophthalmology*, Aug. 28, 2019. https://www.aao.org/eye-health/diseases/what-is-glaucoma (accessed Feb. 03, 2020).

[22] “Disability Statistics.” http://www.disabilitystatistics.org/reports/acs.cfm?statistic=2 (accessed Aug. 16, 2019).

[23] US Department of Education, Center for Education Statistics, “Baccalaureate and Beyond Longitudinal Study (B&B),” 2016. https://nces.ed.gov/surveys/b&b/ (accessed Apr. 26, 2020).

[24] National Science Foundation, “Doctorate Recipients from U.S. Universities 2018 | NSF - National Science Foundation,” 2018. https://ncses.nsf.gov/pubs/nsf20301/data-tables/ (accessed Apr. 26, 2020).

[25] Code.org, CTSA, and ECEP Alliance, “2020 State of Computer Science Education: Illuminating Disparities,” 2020. Accessed: Oct. 15, 2020. [Online]. Available: https://advocacy.code.org/2020_state_of_cs.pdf

[26] S. Mealin and E. Murphy-Hill, “An exploratory study of blind software developers,” in *2012 IEEE Symposium on Visual Languages and Human-Centric Computing (VL/HCC)*, Sep. 2012, pp. 71–74. doi: 10.1109/VLHCC.2012.6344485.

[27] K. Albusays, S. Ludi, and M. Huenerfauth, “Interviews and Observation of Blind Software Developers at Work to Understand Code Navigation Challenges,” in *Proceedings of the 19th International ACM SIGACCESS Conference on Computers and Accessibility*, New York, NY, USA, 2017, pp. 91–100. doi: 10.1145/3132525.3132550.

[28] A. Armaly, P. Rodeghero, and C. McMillan, “A Comparison of Program Comprehension Strategies by Blind and Sighted Programmers,” *IEEE Trans.*

Softw. Eng., vol. 44, no. 8, pp. 712–724, Aug. 2018, doi: 10.1109/TSE.2017.2729548.

[29] P. Rodeghero, C. McMillan, P. W. McBurney, N. Bosch, and S. D'Mello, "Improving automated source code summarization via an eye-tracking study of programmers," in *Proceedings of the 36th International Conference on Software Engineering*, Hyderabad, India, May 2014, pp. 390–401. doi: 10.1145/2568225.2568247.

[30] W. Schweikhardt, "A Programming Environment for Blind APL-programmers," in *Proceedings of the International Conference on APL*, New York, NY, USA, 1982, pp. 325–331. doi: 10.1145/800071.802260.

[31] R. M. Siegfried, D. Diakoniarakis, and K. G. Franqueiro, "Making Visual Programming Accessible to the Blind," *CPSN*, pp. 47–53, 2005.

[32] M. Konecki, "A new approach towards visual programming for the blinds," in *2012 Proceedings of the 35th International Convention MIPRO*, May 2012, pp. 935–940.

[33] M. Konecki, "GUIDL as an aiding technology in programming education of visually impaired," *J. Comput.*, vol. 9, no. 12, p. 6, Dec. 2014.

[34] J. Sánchez and F. Aguayo, "Blind Learners Programming Through Audio," in *CHI '05 Extended Abstracts on Human Factors in Computing Systems*, New York, NY, USA, 2005, pp. 1769–1772. doi: 10.1145/1056808.1057018.

[35] A. C. Smith, J. M. Francioni, and S. D. Matzek, "A Java programming tool for students with visual disabilities," in *Proceedings of the fourth international ACM conference on Assistive technologies*, Arlington, Virginia, USA, Nov. 2000, pp. 142–148. doi: 10.1145/354324.354356.

[36] A. M. Stefik, C. Hundhausen, and D. Smith, "On the Design of an Educational Infrastructure for the Blind and Visually Impaired in Computer Science," in *Proceedings of the 42Nd ACM Technical Symposium on Computer Science Education*, New York, NY, USA, 2011, pp. 571–576. doi: 10.1145/1953163.1953323.

[37] A. M. Stefik, C. Hundhausen, and R. Patterson, "An empirical investigation into the design of auditory cues to enhance computer program comprehension," *Int. J. Hum.-Comput. Stud.*, vol. 69, no. 12, pp. 820–838, Jul. 2011, doi: 10.1016/j.ijhcs.2011.07.002.

[38] A. C. Smith, J. S. Cook, J. M. Francioni, A. Hossain, M. Anwar, and M. F. Rahman, "Nonvisual tool for navigating hierarchical structures," *ACM SIGACCESS Access. Comput.*, no. 77–78, pp. 133–139, Sep. 2003, doi: 10.1145/1029014.1028654.

[39] C. M. Baker, L. R. Milne, and R. E. Ladner, "StructJumper: A Tool to Help Blind Programmers Navigate and Understand the Structure of Code," in *Proceedings of the 33rd Annual ACM Conference on Human Factors in Computing Systems*, New York, NY, USA, 2015, pp. 3043–3052. doi: 10.1145/2702123.2702589.

[40] A. Armaly, P. Rodeghero, and C. McMillan, "AudioHighlight: Code Skimming for Blind Programmers," in *2018 IEEE International Conference on Software*

Maintenance and Evolution (ICSME), Sep. 2018, pp. 206–216. doi: 10.1109/ICSME.2018.00030.

[41] V. Potluri, P. Vaithilingam, S. Iyengar, Y. Vidya, M. Swaminathan, and G. Srinivasa, “CodeTalk: Improving Programming Environment Accessibility for Visually Impaired Developers,” in *Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems*, New York, NY, USA, 2018, p. 618:1-618:11. doi: 10.1145/3173574.3174192.

[42] A. Stefik and S. Siebert, “An Empirical Investigation into Programming Language Syntax.” Association for Computing Machinery, Nov. 01, 2013. Accessed: Apr. 09, 2020. [Online]. Available: http://doi.org/10.1145/2534973

[43] “Scratch - Imagine, Program, Share.” https://scratch.mit.edu/ (accessed Apr. 09, 2020).

[44] “MIT App Inventor.” https://appinventor.mit.edu/ (accessed Apr. 09, 2020).

[45] L. B. Caraco, S. Deibel, Y. Ma, and L. R. Milne, “Making the Blockly Library Accessible via Touchscreen,” in *The 21st International ACM SIGACCESS Conference on Computers and Accessibility*, Pittsburgh, PA, USA, Oct. 2019, pp. 648–650. doi: 10.1145/3308561.3354589.

[46] L. R. Milne, C. M. Baker, and R. E. Ladner, “Blocks4All Demonstration: A Blocks-Based Programming Environment for Blind Children,” in *Proceedings of the 19th International ACM SIGACCESS Conference on Computers and Accessibility*, New York, NY, USA, 2017, pp. 313–314. doi: 10.1145/3132525.3134774.

[47] V. Koushik and C. Lewis, “An Accessible Blocks Language: Work in Progress,” in *Proceedings of the 18th International ACM SIGACCESS Conference on Computers and Accessibility*, Reno, Nevada, USA, Oct. 2016, pp. 317–318. doi: 10.1145/2982142.2982150.

[48] V. Koushik, D. Guinness, and S. K. Kane, “StoryBlocks: A Tangible Programming Game To Create Accessible Audio Stories,” in *Proceedings of the 2019 CHI Conference on Human Factors in Computing Systems*, New York, NY, USA, 2019, p. 492:1-492:12. doi: 10.1145/3290605.3300722.

[49] “Code Jumper.” https://codejumper.com/ (accessed Apr. 09, 2020).

[50] M. Wollerton, “Children who are blind learn how to code with Code Jumper,” *CNET*, 2019. https://www.cnet.com/news/children-who-are-blind-learn-how-to-code-with-code-jumper/ (accessed Oct. 08, 2019).

[51] M. S. Horn and R. J. K. Jacob, “Designing tangible programming languages for classroom use,” in *Proceedings of the 1st international conference on Tangible and embedded interaction*, Baton Rouge, Louisiana, Feb. 2007, pp. 159–162. doi: 10.1145/1226969.1227003.

[52] P. Doustdar, “The Tools of a Blind Programmer,” *Parham Doustdar’s Blog*, 2016. https://www.parhamdoustdar.com/2016/04/03/tools-of-blind-programmer/ (accessed Apr. 26, 2020).

[53] J. P. Bigham, M. B. Aller, J. T. Brudvik, J. O. Leung, L. A. Yazzolino, and R. E. Ladner, “Inspiring blind high school students to pursue computer science with instant messaging chatbots,” in *Proceedings of the 39th SIGCSE technical*

symposium on Computer science education, Portland, OR, USA, Mar. 2008, pp. 449–453. doi: 10.1145/1352135.1352287.

[54] S. K. Kane and J. P. Bigham, “Tracking @stemxcomet: teaching programming to blind students via 3D printing, crisis management, and twitter,” in *Proceedings of the 45th ACM technical symposium on Computer science education*, Atlanta, Georgia, USA, Mar. 2014, pp. 247–252. doi: 10.1145/2538862.2538975.

[55] S. Ludi, “Robotics Programming Tools for Blind Students,” Jun. 2014, Accessed: Apr. 26, 2020. [Online]. Available: http://dspace.calstate.edu/handle/10211.3/121968

[56] “Home | AccessComputing.” https://www.washington.edu/accesscomputing/ (accessed Apr. 26, 2020).

[57] K. Albusays and S. Ludi, “Eliciting Programming Challenges Faced by Developers with Visual Impairments: Exploratory Study,” in *2016 IEEE/ACM Cooperative and Human Aspects of Software Engineering (CHASE)*, May 2016, pp. 82–85. doi: 10.1109/CHASE.2016.025.

[58] A. Armaly, P. Rodeghero, and C. McMillan, “A Comparison of Program Comprehension Strategies by Blind and Sighted Programmers,” in *Proceedings of the 40th International Conference on Software Engineering*, New York, NY, USA, 2018, pp. 788–788. doi: 10.1145/3180155.3182544.

[59] MAXQDA, “MAXQDA: Qualitative Data Analysis Software | Windows & Mac,” *MAXQDA - The Art of Data Analysis*. https://www.maxqda.com/ (accessed Jul. 18, 2019).

[60] Codeacademy, “Learn to Code - for Free,” *Codecademy*. https://www.codecademy.com/ (accessed Dec. 30, 2020).

[61] C. M. Baker, L. R. Milne, and R. E. Ladner, “Understanding the Impact of TVIs on Technology Use and Selection by Children with Visual Impairments,” in *Proceedings of the 2019 CHI Conference on Human Factors in Computing Systems*, New York, NY, USA, May 2019, pp. 1–13. doi: 10.1145/3290605.3300654.

[62] “CodeHS - Teach Coding and Computer Science at Your School | CodeHS.” https://codehs.com/ (accessed Aug. 28, 2020).

[63] “The Quorum Programming Language.” https://quorumlanguage.com/ (accessed Apr. 09, 2020).

[64] Sphero, “STEM Kits & Robotics for Kids | Inspire STEM Education with Sphero,” *Sphero*. https://sphero.com/ (accessed Aug. 28, 2020).

[65] C. M. Baker, L. R. Milne, and R. E. Ladner, “Understanding the Impact of TVIs on Technology Use and Selection by Children with Visual Impairments,” in *Proceedings of the 2019 CHI Conference on Human Factors in Computing Systems*, New York, NY, USA, May 2019, pp. 1–13. doi: 10.1145/3290605.3300654.

[66] K. Baxter, C. Courage, and K. Caine, *Understanding your users: A practical guide to user research methods*. Morgan Kaufmann, 2015.

[67] S. Carter and J. Mankoff, “When Participants Do the Capturing: The Role of Media in Diary Studies,” p. 10, 2005.

[68] "Code.org Computer Science Principles using Quorum Programming Language." https://quorumlanguage.com/lessons/code.html (accessed Aug. 28, 2020).

[69] A. Stefik, R. E. Ladner, W. Allee, and S. Mealin, "Computer Science Principles for Teachers of Blind and Visually Impaired Students," in *Proceedings of the 50th ACM Technical Symposium on Computer Science Education*, New York, NY, USA, Feb. 2019, pp. 766–772. doi: 10.1145/3287324.3287453.

[70] K. Albusays, "Exploring Auditory Cues to Locate Code Errors and Enhance Navigation for Developers Who Are Blind," *SIGACCESS Access Comput*, no. 120, pp. 11–15, Jan. 2018, doi: 10.1145/3178412.3178414.

[71] L. R. Milne and R. E. Ladner, "Blocks4All: Overcoming Accessibility Barriers to Blocks Programming for Children with Visual Impairments," in *Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems*, Montreal QC, Canada, Apr. 2018, pp. 1–10. doi: 10.1145/3173574.3173643.

[72] M. Rouse, "What is a Learning Management System (LMS) and What is it Used For?," *SearchCIO*. https://searchcio.techtarget.com/definition/learning-management-system (accessed Dec. 26, 2020).

[73] K. Shriner, "Total US Degree-Granting Enrollments, 2000-2018," *Center for Distance Education Research*, May 11, 2020. https://distanceeducationresearch.org/total-us-degree-granting-enrollments-2000-2018/ (accessed Oct. 25, 2021).

[74] D. Lederman, "Professors' Slow, Steady Acceptance of Online Learning: A Survey," *Insider Higher Ed*, 2019. https://www.insidehighered.com/news/survey/professors-slow-steady-acceptance-online-learning-survey (accessed Oct. 25, 2021).

[75] H. Kentnor, "Distance Education and the Evolution of Online Learning in the United States," *Curric. Teach. Dialogue*, vol. 17, no. 1 & 2, p. 18, 2015.

[76] M. Cantrall, "Distance Education Timeline," *Sutori*, Jun. 29, 2017. https://www.sutori.com/en/story/distance-education-timeline--kDnyERHUuxfrz78nqjc9Cbew (accessed Oct. 02, 2021).

[77] University of Pheonix, "Online College, Affordable & Always On," *University of Phoenix*. https://www.phoenix.edu/ (accessed Oct. 25, 2021).

[78] A. D. Dumford and A. L. Miller, "Online learning in higher education: exploring advantages and disadvantages for engagement," *J. Comput. High. Educ.*, vol. 30, no. 3, pp. 452–465, Dec. 2018, doi: 10.1007/s12528-018-9179-z.

[79] P. Gautuam, "Advantages And Disadvantages Of Online Learning," *eLearning Industry*, Oct. 10, 2020. https://elearningindustry.com/advantages-and-disadvantages-online-learning (accessed Sep. 21, 2021).

[80] K. Mukhtar, K. Javed, M. Arooj, and A. Sethi, "Advantages, Limitations and Recommendations for online learning during COVID-19 pandemic era," *Pak. J. Med. Sci.*, vol. 36, no. COVID19-S4, pp. S27–S31, May 2020, doi: 10.12669/pjms.36.COVID19-S4.2785.

[81] E. W. Huff, K. Boateng, M. Moster, P. Rodeghero, and J. Brinkley, "Exploring the Perspectives of Teachers of the Visually Impaired Regarding Accessible K12 Computing Education," in *Proceedings of the 52nd ACM Technical Symposium on*

Computer Science Education, New York, NY, USA, Mar. 2021, pp. 156–162. doi: 10.1145/3408877.3432418.

[82] G. Anderson, “Remote learning shift leaves students with disabilities behind,” 2020. https://www.insidehighered.com/news/2020/04/06/remote-learning-shift-leaves-students-disabilities-behind (accessed Oct. 02, 2021).

[83] Y. Kotera, V. Cockerill, P. Green, L. Hutchinson, P. Shaw, and N. Bowskill, “Towards another kind of borderlessness: online students with disabilities,” *Distance Educ.*, vol. 40, no. 2, pp. 170–186, Apr. 2019, doi: 10.1080/01587919.2019.1600369.

[84] K. Vredenburg, J.-Y. Mao, P. W. Smith, and T. Carey, “A survey of user-centered design practice,” in *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, Minneapolis, Minnesota, USA, Apr. 2002, pp. 471–478. doi: 10.1145/503376.503460.

[85] F. Martin, B. Stamper, and C. Flowers, “Examining Student Perception of Readiness for Online Learning: Importance and Confidence,” *Online Learn.*, vol. 24, no. 2, pp. 38–58, Jun. 2020.

[86] A. Pascual, M. Ribera, and T. Granollers, “Empathic communication of accessibility barriers in web 2.0 editing,” in *Proceedings of the 12th Web for All Conference*, Florence, Italy, May 2015, pp. 1–8. doi: 10.1145/2745555.2746642.

[87] K. D. Könings, T. Seidel, and J. J. G. van Merriënboer, “Participatory design of learning environments: integrating perspectives of students, teachers, and designers,” *Instr. Sci.*, vol. 42, no. 1, pp. 1–9, Jan. 2014, doi: 10.1007/s11251-013-9305-2.

[88] K. D. Könings, S. Brand-Gruwel, and J. J. G. van Merriënboer, “An approach to participatory instructional design in secondary education: an exploratory study,” *Educ. Res.*, vol. 52, no. 1, pp. 45–59, Mar. 2010, doi: 10.1080/00131881003588204.

[89] J. Christensen, “Telling stories: Exploring research storytelling as a meaningful approach to knowledge mobilization with Indigenous research collaborators and diverse audiences in community-based participatory research,” *Can. Geogr. Géographe Can.*, vol. 56, no. 2, pp. 231–242, 2012, doi: 10.1111/j.1541-0064.2012.00417.x.

[90] A. Ekelin, P. Elovaara, C. Mortberg, J. Simonsen, and ACM Digital Library, “Exploring Digital Storytelling as a Method for Participatory Design,” presented at the CPSR/ACM, 2008. Accessed: May 20, 2020. [Online]. Available: http://dl.acm.org/citation.cfm?id=1795234

[91] A. Gubrium, “Digital Storytelling: An Emergent Method for Health Promotion Research and Practice,” *Health Promot. Pract.*, vol. 10, no. 2, pp. 186–191, Apr. 2009, doi: 10.1177/1524839909332600.

[92] A. Kankainen, K. Vaajakallio, V. Kantola, and T. Mattelmäki, “Storytelling Group – a co-design method for service design,” *Behav. Inf. Technol.*, vol. 31, no. 3, pp. 221–230, Mar. 2012, doi: 10.1080/0144929X.2011.563794.

[93] C. Andrews, “Accessible Participatory Design: Engaging and Including Visually Impaired Participants,” in *Inclusive Designing*, Cham, 2014, pp. 201–210. doi: 10.1007/978-3-319-05095-9_18.

[94] D. O. Buskermolen and J. Terken, “Co-constructing stories: a participatory design technique to elicit in-depth user feedback and suggestions about design concepts,” in *Proceedings of the 12th Participatory Design Conference: Exploratory Papers, Workshop Descriptions, Industry Cases - Volume 2*, Roskilde, Denmark, Aug. 2012, pp. 33–36. doi: 10.1145/2348144.2348156.

[95] Centers for Disease Control and Prevention, “Family Educational Rights and Privacy Act (FERPA) | CDC,” Feb. 21, 2019. https://www.cdc.gov/phlp/publications/topic/ferpa.html (accessed Feb. 21, 2022).

[96] TypeScript, “Typed JavaScript at Any Scale.” https://www.typescriptlang.org/ (accessed Feb. 10, 2021).

[97] “React – A JavaScript library for building user interfaces.” https://reactjs.org/ (accessed Feb. 10, 2021).

[98] “Material-UI: A popular React UI framework.” https://material-ui.com/ (accessed Feb. 10, 2021).

[99] D. Cho, “React Component Library Comparison - Coding to Learn and Create - Fluid Project Wiki,” Oct. 15, 2019. https://wiki.fluidproject.org/display/C2LC/React+Component+Library+Comparison (accessed Feb. 10, 2021).

[100] D. Kay, “The state of accessible web UI frameworks,” *Darek Kay*, Dec. 19, 2019. https://darekkay.com/blog/accessible-ui-frameworks/ (accessed Feb. 10, 2021).

[101] Microsoft, “What is ASP.NET Core? A cross-platform web-development framework,” *Microsoft*. https://dotnet.microsoft.com/learn/aspnet/what-is-aspnet-core (accessed Feb. 10, 2021).

[102] Salesforce, “Heroku.” https://www.heroku.com/ (accessed Feb. 26, 2022).

[103] W. W. A. Initiative (WAI), “WAI-ARIA Overview,” *Web Accessibility Initiative (WAI)*. https://www.w3.org/WAI/standards-guidelines/aria/ (accessed Dec. 26, 2020).

[104] Tiny, “The Most Advanced WYSIWYG HTML Editor | Trusted Rich Text Editor,” *TinyMCE*. https://www.tiny.cloud/ (accessed Feb. 26, 2022).

[105] “CodeMirror.” http://codemirror.net/6/ (accessed Feb. 26, 2022).

[106] L. Faulkner, “Beyond the five-user assumption: Benefits of increased sample sizes in usability testing,” *Behav. Res. Methods Instrum. Comput.*, vol. 35, no. 3, pp. 379–383, Aug. 2003, doi: 10.3758/BF03195514.

[107] W. Hwang and G. Salvendy, “Number of people required for usability evaluation: the 10±2 rule,” *Commun. ACM*, vol. 53, no. 5, pp. 130–133, May 2010, doi: 10.1145/1735223.1735255.

[108] R. Macefield, “How To Specify the Participant Group Size for Usability Studies: A Practitioner’s GuideJUS,” *J. Usability Stud.*, vol. 5, no. 1, pp. 34–45, Nov. 2009.

[109] Blink UX, “Usability Sample Size Calculator.” https://assets.blinkux.com/usability-sample-size/ (accessed Mar. 02, 2022).

[110] J. Jacobsen, “Make UX measurable and strengthen your company’s UX culture.” TestingTime.

[111] S. Meyer, “The 7 most important UX KPIs and how to measure them,” *TestingTime*, Jan. 29, 2019. https://www.testingtime.com/en/blog/important-ux-kpis/ (accessed Dec. 28, 2020).

[112] S. Y. Park, “An Analysis of the Technology Acceptance Model in Understanding University Students’ Behavioral Intention to Use e-Learning,” *Educ. Technol. Soc.*, vol. 12, no. 3, pp. 150–162, 2009.

[113] A. Smyk, “The System Usability Scale & How it’s Used in UX | Adobe XD Ideas,” *Ideas*, Mar. 17, 2020. https://xd.adobe.com/ideas/process/user-testing/sus-system-usability-scale-ux/ (accessed Mar. 04, 2022).

[114] “CodeHS - Teach Coding and Computer Science at Your School | CodeHS.” https://codehs.com/ (accessed Aug. 28, 2020).

[115] “Project STEM.” https://projectstem.org/ (accessed Mar. 06, 2022).