Developing a Reliable and Economical Web Portal for Meals on Wheels

(Technical Paper)

New Methods in the Teaching of Computer Science

(STS Paper)

A Thesis Prospectus Submitted to the

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

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

Alexander Hicks
Fall, 2019

Technical Project Team Members
Michael Benos
Kyle Leisure
Kevin Naddoni
Maxwell Patek
Joshua Santana
Nathanael Strawser

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

INTRODUCTION

Many interested students approach Computer Science with the belief that they are not "computer people," similar to how many people approach mathematics, which is harmful to their education and should be addressed (Meerbaum-Salant, Armoni, Ben-Ari, 2013). Due to this idea, many of those students or learners do not even attempt to take an introductory level computer science course, or they may take one and struggle, causing them to affirm their preconceptions of the discipline instead of being welcomed in and helped (Robins, Rountree, & Rountree, 2010). As discussed in research by Yadav, Gretter, Hambrusch and Sands in 2016, since many users, and even teachers, potentially lack sufficient technical knowledge, it can sometimes be difficult to create technological solutions for problems affecting mostly non-technical people.

The technical report described below attempts to provide web application functionality for volunteers, but in this case, from discussion with Allie Dudley and Susan Baker in September 2019, the contacts from Meals on Wheels, it was determined that the volunteers tend to be older and less familiar with technology and computer science than most people. Due in part to ideas such as this, the redevelopment of a portal for Meals on Wheels Charlottesville that is built with accessibility in mind, is important to help provide a platform in order to provide a nonprofit the ability to streamline their process while simultaneously introducing technological solutions to problems that were previously out of scope.

TECHNICAL TOPIC

Despite America being one of the richest countries in the world, an estimated eight million of its aging citizens face the threat of hunger (World Bank, 2019; NCOA, 2015). Meals on Wheels is America's oldest and largest organization dedicated to mitigating this issue through

community chapters (MOWA, 2019). The non-profit's local chapter delivers meals to disabled or elderly people in the Charlottesville-Albemarle area who cannot cook or buy food themselves. With the help of volunteers, the organization packs, labels, and distributes meals to customers via various delivery routes. In addition, volunteers drive a few shuttle routes to deliver meals to locations outside of the Charlottesville-Albemarle area (A. Dudley, personal communication, September 27, 2019).

A small team of paid staff administers Charlottesville's Meals on Wheels chapter by managing delivery routes, maintaining current and prospective customer information, and ensuring that all daily jobs are filled by at least one volunteer (S. Bayker, personal communication, September 13, 2019). These administrative tasks can get rather complex due to a combination of daily, weekly, biweekly, monthly, and one-time volunteer shifts and customer needs. Also, most of the meals delivered are sourced via donation, which makes predicting supply difficult.

The greater U.S. Meals on Wheels organization sells professional software to help staff manage the complexity of their tasks; however, the Charlottesville office cannot afford it (A. Dudley, personal communication, October 11, 2019). Thus, staff managed volunteers, customers, and routes by hand until approximately three years ago, when a University of Virginia computer science capstone team created a web portal for them. Adopting this web portal gave Meals on Wheels' staff more time to focus on essential tasks by automating physical reports and tedious manual tasks.

A subsequent capstone team updated the web portal to its current state, but Meals on Wheels' staff still desires improvements. First, staff complained that the web application has become increasingly slow over time. After examining the current codebase, we believe this

slowness is likely due to its cluttered, unclear data storage and the use of an inexpensive hosting solution. Second, staff identified several organizational oddities within the app layout, making some tasks take longer than necessary. Finally, staff requested the addition of new features, including historical report generation and general search functionality.

It is clear that the system needs an update; however, the technical debt accumulated by the separate capstone teams developing features over a two-year period necessitates a rewrite. Our capstone team's goal, therefore, is to write a new application that satisfies Meals on Wheels' needs and has a more reasonable and maintainable backend for long-term deployment, including state-of-the-art modularity via Docker, normalized database models, and cost-effective cloud deployment via Amazon Web Services. By redesigning and modernizing from the ground up, our project should enable Meals on Wheels to operate at lower costs and function more quickly; the organization should have more time and money to help customers in need. We aim to provide a minimum viable product including features such as account creation, assigning volunteers to routes, and volunteer substitution by the end of the current year. We plan on releasing the complete, working product by May 2020.

Requirements for a Minimum Viable Product

- All Users
    - As a user, I should be able to create my own account (including custom username), so I can log in and see personalized information.
    - As a user, I should be able to request to change my password in case I forget it.
- Volunteers
    - As a volunteer, I should be able to release my route on a day, so someone else can substitute for that job.
    - As a volunteer, I should be able to pick up a released route on a particular day, so no routes go without a volunteer.
    - As a volunteer, I should be able to pick up a new route that has not been assigned to any volunteer, so I can plan my hours in advance.
- Staff
    - As staff, I should be able to create clients, so I can accommodate a growing client base.

- As staff, I should be able to generate reports, so I can prepare daily operations.
- As staff, I should be able to manually create delivery routes, so I can customize the volunteer's tasks.
- As staff, I should be able to manually delete delivery routes, so I can avoid cluttering the portal with unused routes.
- As staff, I should be able to assign volunteers to recurring routes, so I can plan delivery.
- As staff, I should be able to substitute one-time volunteers for jobs, so I can ensure that all necessary jobs are filled.
- As staff, I should be able to release volunteers from their recurring routes, so I can assign another volunteer to the recurring route.
- As staff, I should be able to one-time release volunteers from their routes, so I can allow other volunteers to substitute.
- As staff, I should be able to print reports that have been generated by any staff, so can have physical report copies.
- As staff, I should be able to see who is volunteering on a particular day, so I can stay organized and communicate as necessary.

STS TOPIC

Computer science education as a subject is remarkably new, even considering the age of computer science in general. Due to this relative youth, there is much discussion within the community about how it should be taught, especially introductory classes. Many individuals enter computer science classes with the mindset that they just have to get through it, and that they just are not “computer people” (Sherriff, 2016). According to Mau in 2003 and from discussions with Leslie Cintron in 2019, these selected reasons, among many others, have led to what is known as a persistence problem in computer science, especially targeting minorities within the field, such as women or non-white and non-asian men. As a result of this persistence problem, there is much research being done to attempt to open computer science up to more diverse populations. This research is especially important because as technology becomes a more and more important part of our lives, people will have to interact with it more and more, so providing welcoming environment upon their introduction to the field is crucial. This persistence problem is not a new or unknown issue, and several computer science professors at

UVA are working to help fix this problem according to Luther Tychonievich and Jim Cohoon (Cohoon and Tychonievich, 2019).

In order to study this topic, this STS Proposal will explore technological momentum, a prominent framework in the field. As discussed by Hughes in 1994, technological momentum distilled down is the idea that "social development shapes and is shaped by technology," and as such fits this discussion of computer science education cleanly. Computer science education did not exist until about fifty years ago, and as such, it has not had the history of other branches of education such as physics or English. Technological momentum as a concept is time dependent, which is another reason it is especially relevant for this topic of study due to the rapidly changing nature of computer science in general. Many other disciplines use textbooks as sources of knowledge for classes and readings, but because computer science changes so fast, textbooks can be outdated as soon as the year they are printed (Sherriff, 2016). This difference, as well as other factors necessitate some changes in how computer science is taught, but some of that may need to change. Technological momentum is a great framework, but it is not without its critics. Since one of the main ideas of technological momentum is combining technological determinism and the social construction of technology, it shares many of its flaws with those two theories (Hughes, 1994). Some specific ideas to point out are the development of automobiles, which falls under technological momentum because as automobiles have grown and developed, there was a large push towards more economically friendly and greener vehicles, but this can have the unintended consequence of raising prices and forcing more expensive and more environmentally dangerous vehicles on the poor in areas were automobiles are required to work or even just a standard of living.

This research is important because as the world transitions into a more and more technological place, it is important that everybody is able to enjoy the benefits of this new world (Meerbaum-Salant et al., 2013). To this end, computer science education must grow and adapt in order to broaden its appeal to a greater spectrum of people than it is currently serving well.

RESEARCH QUESTION AND METHODS

Computer science is one of the fastest growing engineering disciplines in the world, and it has applications on vast quantities of problems (Kay et al., 2000). However, its application to some problems can be restricted by both the user's feelings about computer science, as well as their knowledge of it. As such, a relevant research question is:

> How are introductory computer science classes, specifically how CS1 (CS 111x) and CS2 (CS 2110), are taught at the University of Virginia, and how can they be improved, specifically by exploring their similarity to introductory foreign language classes?

In order to explore this question, two primary research methods will be used, documentary research methods and interviews (Seabrook, 2019, slide 5 and 11). Due to the targeted nature of this paper of University of Virginia computer science, one of the main sources of information will be UVA computer science professors because they are, in some cases, the only ones with the logic behind some of the decisions made in teaching CS1 at UVA. Interviews will also be conducted with introductory level foreign language instructors in order to begin studying whether methods used in foreign language instruction are applicable to computer science. The interview questions will be written and edited by the end of the current year in order to schedule and complete interviews in January 2020. Additionally, documentary research methods will be used to distill current state of the art research on the topic at other universities

into something that can be analyzed and studied. One important source is the journal, Computer Science Education. This journal explores many of the topics and factors relevant to teaching computer science, including how CS1 and CS2 are taught.

CONCLUSION

The technical project consists of one main deliverable, a web application that Charlottesville Meals on Wheels organization will be able to use for both volunteer and customer management, allowing the volunteers there to focus on more important issues. The expectation for this project will be a fully working portal by the end of the Spring 2020 semester that will help streamline the process for the Meals on Wheels managers and allow more of the volunteers to access the portal. The expectation of the STS Research is a paper that explores possibilities to improve introductory computer science education here at UVA including the application of foreign language teaching methods to computer science, which in turn could help in restructuring the way these CS1 and CS2 classes are taught, opening computer science at UVA up to a more diverse population.

REFERENCES

Ala-Mutka, Kirsti M (2005). A Survey of Automated Assessment Approaches for Programming Assignments, Computer Science Education, 15:2, 83-102, DOI: 10.1080/08993400500150747

Allman, E. (2012). Managing technical debt. *Communications of the ACM*, *55*(5), 50. https://doi.org/10.1145/2160718.2160733

Goldstein, I. (2019). What! No GUI?. *Information Systems Education Journal,* 17(1) pp 40-48. http://isedj.org/2019-17/ ISSN: 1545-679X. (A preliminary version appears in The Proceedings of EDSIGCON 2018)

Grover, Shuchi, Pea, Roy & Cooper, Stephen (2015). Designing for deeper learning in a blended computer science course for middle school students, Computer Science Education, 25:2, 199-237, DOI: 10.1080/08993408.2015.1033142

Hughes, Thomas (1994). Technological Momentum https://collab.its.virginia.edu/access/content/group/484deb3f-d8f1-405a-8b1c-5541ca7dd540/Readings/Hughes%20-%20Technological%20Momentum.pdf

Kay, Judy, Barg, Michael, Fekete, Alan, Greening, Tony, Hollands, Owen, Kingston, Jeffrey H. & Crawford, Kate (2000). Problem-Based Learning for Foundation Computer Science Courses, Computer Science Education, 10:2, 109-128, DOI: 10.1076/0899-3408(200008)10:2;1-C;FT109

Kong, Siu-Cheung & Wang, Yi-Qing (2019). Positive youth development from a "3Cs" programming perspective: a multi-study investigation in the university, Computer Science Education, DOI: 10.1080/08993408.2019.1599646

Mannila, Linda, Peltomäki, Mia & Salakoski, Tapio (2006). What about a simple language? Analyzing the difficulties in learning to program, Computer Science Education, 16:3, 211-227, DOI: 10.1080/08993400600912384

Mau, W. C. (2003). Factors that influence persistence in science and engineering career aspirations. The Career Development Quarterly, 51(3), 234-243.

Meerbaum-Salant, Orni, Armoni & Mordechai, Michal (Moti) Ben-Ari (2013). Learning computer science concepts with Scratch, Computer Science Education, 23:3, 239-264, DOI: 10.1080/08993408.2013.832022

Microsoft (2017). Description of the database normalization basics. https://support.microsoft.com/en-us/help/283878/description-of-the-database-normalization-basics.

MOWA. (2019). Meals on Wheels America. National Office. https://www.mealsonwheelsamerica.org/learn-more/national

NCOA. (2015, June 4). National Council On Aging. Facts About Senior Hunger. https://www.ncoa.org/news/resources-for-reporters/get-the-facts/senior-hunger-facts/

Robins, Anthony , Rountree, Janet & Rountree, Nathan (2003). Learning and Teaching Programming: A Review and Discussion, Computer Science Education, 13:2, 137-172, DOI: 10.1076/csed.13.2.137.14200

Seabrook, Bryn (2019). Methods Workshop [PowerPoint Slides]. Retrieved from https://collab.its.virginia.edu/access/content/group/484deb3f-d8f1-405a-8b1c-5541ca7dd540/Presentations/Methods%20Workshop.pptx

Sherriff, Mark (2016). Lecture on Introduction to Programming. Personal Collection of M. Sherriff, University of Virginia, Charlottesville VA.

Sunggingwati, D., & Haviluddin. (2019). Learning Style Preferred by English and Computer Students in Indonesia Context. *International Journal of Emerging Technologies in Learning*, *14*(10), 46–61. https://doi.org/10.3991/ijet.v14i10.9997

Weintrop, David (2019). Block-based programming in computer science education. *Commun. ACM* 62, 8 (July 2019), 22-25. DOI: https://doi-org.proxy01.its.virginia.edu/10.1145/3341221

Williams, Laurie, Wiebe, Eric, Yang, Kai, Ferzli, Miriam & Miller, Carol (2002). In Support of Pair Programming in the Introductory Computer Science Course, Computer Science Education, 12:3, 197-212, DOI: 10.1076/csed.12.3.197.8618

World Bank (2019). GDP per capita. https://data.worldbank.org/indicator/ny.gdp.pcap.cd

Yadav, Aman, Gretter, Sarah, Hambrusch, Susanne & Sands, Phil (2016). Expanding computer science education in schools: understanding teacher experiences and challenges, Computer Science Education, 26:4, 235-254, DOI: 10.1080/08993408.2016.1257418