HELP Line: A Call Log Management System
(Technical Report)

Advocates and Critics of Autonomous Vehicles
(STS Research Paper)

An Undergraduate Thesis Portfolio
presented to the faculty of the
School of Engineering and Applied Science
in partial fulfillment of the requirements for the degree
Bachelor of Science in Computer Science

by

Niti Paudyal

May 5, 2016

Preface

Automated software systems operate with minimal or no human intervention. The rise of automated systems raises questions about their value and has major social implications. The effects maybe detrimental, and a thorough analysis of the implications must be determined before these systems become mainstream.

Madison House's HELP Line is an anonymous telephone hotline service intended to serve the University of Virginia community. HELP line aims to provide the community a confidential and non-judgmental listener to discuss any issues a caller may be dealing with. HELP line is staffed by over 100 volunteers, and to increase HELP line effectiveness the capstone team created a call log management system that stores crucial information about callers. Although the old system provided a means to store information; it was slow, unstable and poorly implemented. The new system tracks the development of a caller to provide volunteers accurate information to be the most informed listener. The call log provides reporting on caller statistics, a cleaner user workflow, and better searching capabilities for volunteers and administrators to streamline the overall HELP line program.

Automated automobiles, known as Autonomous Vehicles or AVs, are in development. AVs will appear on public roads by the 2020's. AV advocates and critics clash as they fail to recognize the two-fold situation regarding reduction of risk and cost as participant groups have different interests and values. Before AVs become common, however, they must overcome legal and ethical obstacles by answering what kind role artificial intelligence should have in AVs. Advocates credit AVs to provide savings in safety, greenhouse gases, productivity among others but these benefits are overstated as they neglect to recognize potential for new risks or cost. Labor markets may see large disruptions as groups such as the New York Taxi Workers Alliance fight early automated systems like Uber and Lyft with AVs on the horizon. AVs resemble past technological innovations where automation threatens entire jobs sectors, and reshapes many jobs. AVs rattle the traditional insurance industry raising questions of who should bear responsibility for errors caused by AVs. These liability concerns quickly become an issue of what ethical model cars should bear. What is the future of AVs? Many of the issues are ethical, and not particular to AVs but to all artificial intelligent systems. Policy regarding a universal standard of norms for the public will have to be established before AVs become mainstream.

Advocates and Critics of Autonomous Vehicles

An STS Research Paper
presented to the faculty of the
School of Engineering and Applied Science
University of Virginia

by

Niti Paudyal

May 5, 2016

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.

Signed: ____

Approved: _______________________________________ Date _________________________
Peter Norton, Department of Engineering and Society

Introduction

Automated software systems operate with minimal or no human intervention. A common example is online language learning platforms such as Rosetta Stone that eliminate teachers by replicating their expertise through computer-assisted language learning. These substitutes can displace teachers and degrade the quality of education offered to students (Clardy, 2009). Yet automation of education has proven economical and the number of users engaged in it is growing. The trend has major social implications.

Automation has extended to the realm of automobiles: Tesla and Google (among others) are developing autonomous vehicles (AVs). Semi-autonomous vehicles are now in service. Levels of vehicle autonomy are defined by the National High Traffic and Safety Administration, from level 0 (no-automation) where the "driver is in complete and sole control," to level 4 (full self-driving automation) where "the vehicle is designed to preform all safety-critical driving functions and conditions for an entire trip" and drivers are "not expected to be available for control at anytime during the trip" (NHTSA, 2013). These companies seek to produce a completely autonomous car (level 4). This goal, however, is socially, legally, and economically disruptive. It implicates insurance and manufacturing companies and the job market for taxi and truck drivers, and raises problems of safety, privacy, and security of the artificial intelligence agent in AVs. AVs are therefore controversial. How can the disagreement between advocates and critics of autonomous vehicles be explained?

AVs will appear on public roads by the 2020's. Before AVs become common, however, they must overcome legal obstacles, which are major barriers to entry for level-4 autonomous cars. Advocates and critics disagree because they do not recognize the two-fold situation regarding reduction of risk and cost as participant groups have different interests and values.

Review of Research

Researchers have examined the inevitable disruptions AVs will bring without considering how participant groups collide while supporting their causes.

Haggemans and Thierer (2014) contend that ethical issues in high-stake situations will be worked out over time through a trial-and-error processes. Hagemann and Thierer (2014) and Fagnant and Kockelman (2013) acknowledge that AVs will disrupt labor markets.

While most AV advocates claim that AVs will eliminate human error in road accidents, and thus eliminate 90% of crashes, Schoettle and Sivak (2014) disagree, claiming that AVs may preform no better than an experienced human driver. They contend that during the transition period, when AVs and conventional cars coexist, safety might worsen. Fagnant and Kockelman (2013) claim that AVs would reduce private car ownership and thus improve environmental conditions. Schoettle and Sivak (2014), however, contend that people may spend more time in cars, negating AVs benefits.

Fagnant and Kockelman (2013) and Haggemans and Thierer (2014) contend that the responsibility for insurance and liability not yet unknown at this time and will be a matter of policy.

Policy, Ethics and Artificial Intelligence

Public perceptions of autonomous cars are mixed, and include unfavorable views (Schoettle & Sivak, 2014). The University of Michigan's Transportation Institute surveyed public opinion in the U.S., U.K., and Australia with a sample of 1,533 people, and concluded that 54 % were apprehensive about partially autonomous cars; 60% were apprehensive about fully autonomous cars. But autonomous systems in other modes of travel may raise expectations. Stephanie Brinley of HIS Automotive claims that autonomous trucks will likely resemble passenger airplanes at first, where the driver becomes “less of a driver and more of a decision maker,” and with time the public will ease into AVs.

However, experts disagree about how AVs should manage decisions in rare, high-stakes situations. Fagant and Kockelman (2014) raise such philosophical questions, such as how should AVs prioritize the safety of their own occupants relative to those of other vehicles or to pedestrians. This is commonly known as the Trolley Problem posed by philosopher Philippa Foot (Thomson, 1985). The introduction of AVs makes personal ethics a matter of public concern as those ethical issues will manifest in legal and policy choices.

Haggemans and Thierer (2014) contend that for “life-and-death” situations, there will be “thorny ethical questions” but that they will be worked out over time through trial and error. Representative Tom Petri (R-WI), chairman of the Subcommittee on Highways and Transit of the House Transportation and Infrastructure Committee, agrees, claiming that there are “many issues with driverless vehicles, and it’s impossible to anticipate all — or even most — of them,” and notes Congress should instead “maintain a flexible system that deals with real problems rather than anticipate problems that don’t exist yet” (Ali, 2014). Organizations such as Technology Liberation Front and the Competitive Enterprise Institute that espouse principles of limited government warn against the precautionary principle or the “natural fear of the unknown to translate into policies,” citing its consequences (Scribner, 2014 & Thierer, 2012). AV critics thrive on natural fear and pose a “serious threat to technological progression” as “hypothetical worst-case scenarios trump all other considerations” (Thierer, 2012). Both organizations argue that policy makers should maximize the potential for “permission less innovation”, the idea that experimentation with new technologies should generally be allowed by default, to permit ongoing experimentation to produce the best product, before public policy applies ethical standards.

The classic trolley problem can be adapted to AVs, as a utilitarian, consequentialist, deontological or duty-based problem. Bonnefon, Sharriff, and Rahwan (2015) conducted a survey through Amazon's Mechanical Turk, an online crowdsourcing tool, concluding participants were "generally comfortable with utilitarian AVs, programmed to minimize an accident's death tool." However, people were much more resistant to take the utilitarian course for the car to swerve and kill its owner when they were the driver. MIT *Technology Review* (2015) notes that if AVs are "programmed to sacrifice their owners, then more people are likely to die because ordinary cars are involved in so many more accidents" – thus a Catch-22 situation which could then affect consumer adoption. Which model will be adapted and who chooses? Will cars follow different rules or one standard? Will individuals who are used to making their own ethics judgment resist these standards? These questions remain unanswered but are critical to the development of artificial intelligent systems behind AVs.

All artificial intelligence (AI) driven systems face ethical questions. Similarly, movements against autonomous weapons has already begun, spearheaded by Apple's co-founder Steve Wozniak and Tesla's Elon Musk, with over 14,000 signatures (Autonomous Weapons, 2015). The open letter notes though AI has potential for good, it warns that autonomous machines are detached from their actions and thus their consequences. AVs face the same backlash, and a similar standard could be claimed. AI driven risk analysis lies at the core of the problem for critics. In traditional cars, humans are held responsible for decisions. Neuroscientist Antonio Damasio, and most theories claim human decisions are a mix of logic and emotions (Pontin, 2014). The question then becomes can AI replicate human emotions? Would emotions take away from the power of the AI? Alan Turing, the father of theoretical artificial intelligence, claims in his "polite convention" that if a machine behaves like a human, then it is as intelligent as the human being itself (Turing, 1950). AV advocates counter, claiming the use of computing power in AVs is in effect to do what humans cannot. Policy for AVs must answer a long debated question in the artificial intelligence community, to include or exclude emotions in AI.

Safety and Savings

Bob Joop Goos, Chairman of the International Organization of Road Accident Prevention, states human error accounts for 90% of road accidents (Abkowitz, 2014). AV advocates claim that AVs could prevent all such crashes. However, they fail to consider that AVs could introduce new risks, such as system failures, cyber attacks, and risk compensation or "the tendency of road users to make additional risks when they feel safer" (Litman, 2015).

A 2016 Morning Consult poll surveying about 2,000 people revealed that 43% of respondents thought AVs are not safe; only 32% believed they are safe (Johnson, 2016). The poll "hinted that people are unlikely to change their minds any time soon." Almost two-thirds reported they are unlikely to buy a AV car in the next 10 years. Consumers 30 and older are more resistant, citing fear in letting go of control. Manufacturers address the resistance with "semi autonomous" cars. However, such control raises concern for manufacturers. In fully autonomous cars, Google warns of "human element" risks (Shepardson, 2016). Google claims that providing human occupants "with mechanisms to control things like steering, acceleration, braking… could be detrimental to safety because the human occupants could attempt to override the (self-driving system's) decisions." Tesla Model S owner Bill Nelson claims he activated automatic steering because “as we approached the concrete wall, my instincts could not resist. I grabbed the wheel” (USA Today, 2016). Google fears risk compensation effects like this.

As a connected vehicle, AVs are prone to cyber attacks. Advocates fail to mention the safety risk associated with cyber attacks with AVs as they shift their focus to net savings for AVs. A reduction of 1,000 lives are savings but the details of those savings prove critical to the consumers. If AVs eliminate all current accidents but introduce new ones due to cyber attacks, even with net savings, critics may argue that the introduction of new types of incidents are not justified. To justify the tradeoff, a consequentialist may now require twice as many savings rather than a strict comparison on net savings (Lin, 2013). However, Lin argues that in a given year an “arbitrarily and unpredictably doomed to be victims” thus “there’s no issue with replacing some or most of them with a new set of unlucky victims.”

AV advocates contend they will be greener. The mission of SAFE’s (Securing America’s Future Energy) Autonomous Vehicle Task Force is to encourage the “widespread deployment” of AVs while serving the group’s interest in a cleaner environment. Fagnant and Kockelman (2013) claim that AVs will reduce vehicle ownership, while self-driving taxis would replace personal vehicles. However, Schoettle and Sivak (2014) contend that AVs would enhance the mobility of the blind, the disabled, underage children, and the elderly, increasing the overall travel per vehicle up to 75%. The environmental claims may be unrealistic (Litman, 2015). Though SAFE’s task force envisions AVs as a means to a cleaner environment, they may prove detrimental.

Productivity, fuel and across the board savings from AVs may be massive. Platooning, where trucks cut wind resistance is a component of autonomous trucks that can cut fuel costs by travelling in close processions (Kaye & Sterling, 2016). Costs savings in wages and fuel bills may be massive. The American Trucking Association (2015) reports the freight business alone is worth $700 billion in 2014. Gerald Waldron, managing director of Australian Road Research, says AVs provide “a huge benefit, and that’s before you start looking at the impacts on road safety” citing AV testing estimates savings of 40 percent. Given these savings, pricing of AVs also does not seem to be an issue either as Morgan Stanley (2015) reports “full autonomous capability will add only about $10,000 to the cost of a car, at today’s prices, which we expect will fall significantly by the time technology is ready to be commercialized.” Morgan Stanley notes with negligible costs, AVs overall could contribute $1.3 trillion in savings for the US economy, and over $5.6 million globally. These savings translate to 8% of the US annual GDP where the reallocation of these savings may have major implications itself. The productivity gains are estimated to be $507 billion. The savings are hard to decipher, as consumers instead of driving may devote time to other activities such as sleep, phone calls, work, etc. The possibilities for consumers are endless. AVs bring across the board savings, however, it is unknown if those savings will implicate the all groups in economy positively.

Many of the safety and fuel or environment benefits are overstated as they neglect the potential for new risks or costs. AVs mitigate risks by eliminating human error but can introduce new ones. AVs may seem to decrease emissions of greenhouse gases as car ownership declines in favor of car sharing, but AVs may increase ridership with new passengers and increased mobility. AVs may yield net savings in lives but they may be prone to new incidents due to cyber attacks. The estimated savings AVs provide is massive but fail to recognize how those savings would translate in the economy.

Disruptions to Labor Markets

Driverless cars will cause major disruptions in labor markets as they may replace the 240,000 taxi drivers or 1.6 million truck drivers in the U.S., claims Fagnant of the Eno Center for Transportation (Pulmer 2013). As AVs replace humans, disruptions to job markets will bring scrutiny to them.

Hagemann and Thierer (2014) predict that AVs will have “similar disruptive impacts” on various industries that will result in “short-term economic perturbations and employment dislocations.” They draw parallels to computers for comparison, quoting Peter Singer: “For hundreds of years, there was a highly skilled profession of men who did mathematics for hire. They were well paid, many making the equivalent of $200,000 a year. They were called ‘calculators.’” Hagemann and Thierer (2014) claim just like computers, professions have been consistently reshaped by technology, and intelligent vehicles will follow suit. AVs may bring radical disruption or complete extinction to some professions, such as cabbies, truck drivers, and truckers, but as with all technological advances they occur incrementally. With the automation of dispatching and the luxury of using unlicensed drivers, conventional taxi drivers are already feeling disruptions through ride sharing applications like Uber, and Lyft. This disruption has caused organizations like the New York Taxi Workers Alliance (2015), a union for the 50,000 taxi drivers in NYC, cry out for driver unity against ride sharing apps like Uber noting this new paradigm is a situation in which only Uber profits, claiming that Uber threatens drivers as they are “rushing to bring out the first ‘Driverless Car’ so that they don’t have to ‘lose’ the fare to the driver.” The Taxi Alliance resists technological innovations that may disrupt their work.

Though changes such as ride sharing applications are incremental, Hagemann and Thierer (2014) cautions not to “discount the possibility that a new technology will kill off a large sector;

nor can we assume away the possibility that all industries will adapt to new advances in technology." However, they also argue that this disruption could create different types of jobs, where the traditional mechanic has to gain a few more skills now to fix an AV. Automation may reshape more jobs than it destroys.

NPR reports that in 2014 trucking, delivery, and tractor drivers were the most common jobs in 27 of the 50 US states (Bui, 2015). The International Brotherhood of Teamsters or Teamsters, an American labor union of freight drivers and warehouse workers, has the most to lose from driverless cars (Who Are The Teamsters?, 2015). Unions such as Teamsters and the American Trucking Associations claim that trucks will continue to need human operators, much like passenger airplanes (Davies, 2015). Teamsters' resistance can be explained as a defense of their members' livelihoods.

Truckers like Kevin Baxter, former Australian truck driver, continually downplay concerns stating "the whole time I've been trucking, the railways have been going to replace me, but that hasn't happened" (Kaye & Sterling, 2016). The trucking industry has seen shortages worldwide; the number of drivers is expected to stall while demand is forecasted to jump to 80% by 2031. The American Trucking Association aware of AV technology discounts the magnitude of the impact AV may have on truckers (Costello & Suarez, 2015). Truckers tenaciously compare the effect of AVs to the impact older transportation systems like railroads and air transport disregarding that AV technology does not introduce a competitor rather it reduces the human operator, the truckers themselves.

Though AVs may disrupt the labor market, even destroying some sectors, the impact of technological innovation is familiar and inevitable.

Insurance and Liability

AVs raise complex liability issues for insurance and manufacturing companies. In California and Nevada, where self-driving cars are legal, insurance companies want stricter regulations (Cohen, 2013). Current regulations from National Highway Traffic Association fail to outline which party will be responsible for incidents. Consumers, manufacturers, and insurance companies remain torn.

Fagnant and Kockelman (2014) of the Eno Center for Transportation claim that even with near-perfect autonomous driving, instances where crashes are unavoidable exists. In real-time decision making, human drivers are not held responsible “to circumstances beyond their control, regardless of whether their decision was the best” while AVs are equipped with sensors, and predictive algorithms that lead to more informed decisions. This raises problems of responsibility. Consumer Reports notes that it will be answered over time, and may depend on the situation (Monticello, 2016).

Insurance companies collect 157 million in auto premiums annually where 90% of accidents are caused by human error (Abkowitz, 2014). Will AVs eliminate auto insurance companies? AVs may cause “significant reduction in insurance premiums” according to RAND Corp, as the average auto insurance in the U.S. is $1,029 for sedans (AAA, 2015). While personal auto insurance liability will decrease, RAND predicts manufacturers liability will increase (Anderson et al., 2014). KPMG’s insurance task force concludes that current safety technology for crash prevention already accounts for a 7-15% decline in claim frequency while insurance companies have yet to respond with new insurance policies (KPMG, 2015). Frank Diana of TCS Global Consulting predicts that AVS will bring massive reductions in accidents, commute time, and the overall the number of cars on the road creating a ripple effect disrupting established industries where such as 90% of insurance premiums could disappear (Diana, 2014). AVs result in lower claim volume with a $200 billion impact in personal and commercial auto insurance each year as of most of the claims are a direct function of the frequency and severity of accidents. With accidents curtailed by AVs, those premiums go away. Cutts (2014) claims that the "human element" to driving was the major monkey maker for automobile insurers and with AVs eliminating 90% of claims, AVs could see heavy resistance. However, liability could shift to manufacturers as they handle data of vehicle operation and can correct risks. This additional liability runs risk for manufacturers to bankruptcy should something fail, a clear indicator to resist AVs. Yeomans of Llyods (2014) notes that AVs could later draw parallels to product liability insurance where, unlike traditional cars, insurers are more concerned with the model of the car than the user. However, If AVs became more of a service than a good, liability would transfer to who ever owns the fleet. Many unknowns for insurance and liability remain but Cutts (2014) claims product laws for AVs will gradually evolve through a body of case law. The impact of liability, Cutts (2014) says, will depend on the situation, and information known to the AVs to the ethical model chosen will determine how it implicates the parties involved.

Conclusion

Autonomous Vehicles will be mainstream in the 2020's and many questions remain unresolved. Many of the issues are all ethical. It is not an issue particular to AVs but to all artificial intelligent systems which in the future will become norm. Policy regarding a universal standard of norms for the public will have to established, and AVs are the just the beginning of this process. Participant groups will continue to advance their causes with advocacy and litigation as more issues are brought into the judicial process, and more will rise as they foresee more disruptions cross industry sectors similar to other technological disruptions like internet provided.

References

AAA. (2015). Cost of Owning and Operating Vehicle in U.S. Increases Nearly Two Percent According to AAA's 2013 'Your Driving Costs' Study. www.newsroom.aaa.com

Abkowitz, A. (2014, September 10). Do Self-Driving Cars Spell Doom for Auto Insurers? Bloomberg. www.bloomberg.com

Ali, A. (2014, March 20). Government Hitting The Breaks On Driverless Cars. OZY.

American Trucking Association. (2015, May 11). Tracking Revenues $700 Billion for the First Time According to New Report. www.trucking.org

Autonomous Weapons: An Open Letter from AI & Robotics Researchers. (2015, July 28). Future of Life Institute. www.futureoflife.org

Bonnefon J. , Shariff, A. , Rahwan, I. (2015, October 13). Autonomous Vehicles Need Experimental Ethics: Are We Ready for Utilitarian Cars?

Bui, Q. (2015, February 5). The Most Common Job In Every State. NPR. www.npr.org

Bomey, N. (2016, March 15). Self-driving car leaders ask for national laws. USA Today. www.usatoday.com

Clardy, A. (2009, June). Distant, on-line education: Effects, principles and practices. Unpublished manuscript. ERIC.

Cohen, A. (2013, January 14). Will Self-Driving Cars Change the Rules of the Road? Time. www.ideas.time.com

Costello, B., & Suarez R. (2015, October). Truck Drive Shortage Analysis 2015. American Trucking Association. www.trucking.org

Cutts, J. (2014). Technology Trends to Watch: On the Road to Driverless Cars. Consumer Electronics Association.

Davies, A. (2015, May 15). World's First Self Driving Semi Truck Hits the Road. Wired. www.wired.com

Diana, F. (2014, Feb 5). Autonomous Vehicles: A Disruption Case Study.

Fagnant, D., & Kockelman, K. (2014, October 1). Preparing a Nation for Autonomous Vehicles Opportunities, Barriers and Policy Recommendations. Eno Center for Transportation. www.enotrans.org

Kaye, B., & Sterling, T. (2016, Jan 21). Safety, savings fuel push for driverless trucks. Reuters.

KPMG. (2015). Automobile insurance in the era of autonomous vehicles. www.kpmg.com

Lin, P. (2013, July 13). The Ethics of Saving Lives With Autonomous Cars is Far Murkier Than you Think. Wired. www.wired.com

Litman, T. (2015, 10 December). Autonomous Vehicle Implementation Predictions. Victoria Transport Policy Institute.

Monticello, M. (2016, March 31). The State of the Self-Driving Car. Consumer Reports.

Morgan Stanley. (2015, Jan 23). Autonomous Cars: The Future is Now. www.morganstanley.com

NHTSA. (2013, May 20). U.S. Department of Transportation Releases Policy on Automated Vehicle Development. www.nhtsa.gov

Johnson, F. (2016, Feb 8). Voters Aren't Ready for Driverless Cars. Morning Consult.

Pontin, J. (2015, 17 June). The Importance of Feelings. MIT *Technology Review.*

Pulmer, B. (2013, October 23). Here's what it would take for self-driving cars to catch on. Washington Post. www.washingtonpost.com

Schoettle, B., & Sivak, M. (2014). A Survey of Public Opinion About Autonomous and Self-Driving Vehicles in the U.S., the U.K., and Australia. *The University of Michigan Transportation Research Institute*.

Schoettle, B., & Sivak, M. (2015). Potential Impact of Self-Driving Vehicles on Household Vehicle Demand and Usage, Sustainable Worldwide Transportation Program. *The University of Michigan Transportation Research Institute*.

Scribner, M. (2014, April 23). Self-Driving Regulation: Pro-Market Policies Key to Automated Vehicle Innovation. Competitive Enterprise Institute.

Securing America's Future Energy. (2015, 14 October). "SAFE launches Autonomous Vehicle Task Force to Advance U.S. Economic and National Security." www.secureenergy.org

Shepardson D. (2016, Feb 10). In boost to self-driving cars, U.S. tells Google computers can qualify as drivers. Reuters.

The New York Taxi Workers Alliance (2015). Call for Driver Unity! http://www.nytwa.org

Thierer, A. (2012, Feb 4) The Precautionary Principle Meets Driverless Cars in DC. The Technology Liberation Front.

Thomson, J. (1985, May). The Trolley Problem. The Yale Law Journal, 94(2).

Turing, A. (1950). Computing Machinery and Intelligence. Mind.

Who Are The Teamsters? (2015). www.teamster.org/about

“Why Self-Driving Cars Must Be Programmed to Kill” (2015, Oct 22). MIT *Technology Review*

Yeomans, G. (2014). Autonomous Vehicles: Handing Over Control: Opportunities And Risk For Insurance. Llyod’s Exposure Management.

HELP Line: A Call Log Management System

A Capstone Report
presented to the faculty of the
School of Engineering and Applied Science
University of Virginia

by

Niti Paudyal

with

Jasmine Cha
Kevin Chan
Allen Huynh
Apurva Kasanagottu
Ben Lowman
Seamus Lowry

April 8, 2016

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.

Signed: ____

Approved: ______________________________________ Date ________________________
Aaron Bloomfield, Department of Computer Science

HELP Line: A Call Log Management System

Allen Huynh
University of Virginia
abh3hu@virginia.edu

Apurva Kasanagottu
University of Virginia
ak4zk@virginia.edu

Ben Lowman
University of Virginia
brl2xx@virgnia.edu

Jasmine Cha
University of Virginia
jbc4gy@virginia.edu

Kevin Chan
University of Virginia
khc8gx@virginia.edu

Niti Paudyal
University of Virginia
np8bx@virginia.edu

Seamus Lowry
University of Virginia
spl6yc@virginia.edu

Abstract

HELP Line is an anonymous telephone hotline service intended to serve the University of Virginia (UVa) community. The telephone hotline maintains a call logging system in which volunteers log call information to better service callers. The previous logging system was unstable and difficult to use; this led to the development of a new HELP Line system. This system improved the productivity of volunteers through additional features such as searching, reporting, training mode, and tagging callers. It also enhanced old components of the system which include notifications, correct NetBadge authentication, and comments. This system was developed using Django 1.8.4, Python 3.5, and MySQL. Future work includes increased optimization of searching and reporting, an emphasis on User Interface and User Experience, and an online chat system to complement the telephone hotline.

1. Introduction

HELP Line is an anonymous telephone hotline service intended to serve the University of Virginia (UVa) community. HELP Line is affiliated with UVas volunteer center, Madison House, and is run by Program Directors. These program directors are are students with extensive volunteer experience with HELP Line. Program Directors help train and manage 100 HELP Line volunteers, who field over 500 calls per semester and talk to callers about issues including depression, assault, and stress from school. HELP Line volunteers then record these calls in a logging system. Currently, they use a Django-based system that allows features such as adding a caller, adding a call, merging callers, creating comments, and maintaining a log. This system contains the bare requirements for logging phone calls; however, the lack of certain capabilities lead to inefficiencies and inaccurate recording. While the non-profit has a current system, it is unstable and unreliable; therefore, it is difficult to estimate how long it takes to complete its tasks. Also, the lack of search capabilities and ability to disable volunteers creates issues of duplicate entries and allows inactive volunteers access to the system. Furthermore, due to the unreliable nature of the former system, staff were often forced to record calls in Google Documents. Program Directors then transfer the calls from the document to the system, which requires additional time.

Given these limitations, HELP Line was selected as an organization to participate in UVa CS 4970/4971, the Service Learning Practicum (SLP). The SLP is a year-long computer science capstone fulfilling course that focuses on agile software development model projects. The course is split into monthly phases that consist of two iterations, which are two weeks long. SLP pairs capstone groups with non-profit organizations to build a system for the organization and allow students to work on a project that affects real users.

2. Related Work

HELP Line previously used an older, custom, online logging system. This call log fulfills most of the requirements of our system. However, their old system is difficult to use, contains many bugs, and is very slow. There is one developer in charge of maintaining the old system but the server is frequently down, and HELP Line volunteers must resort to using Google documents to store call and caller data. Program Directors then input all the missing data once the website is functioning again. The old system also lacks features such as generating reports, notifying volunteers when others comment on their calls, and providing a training mode for new volunteers. In addition, the old HELP Line contained inaccurate and corrupted data due to mishandling the deletion of both volunteers and callers.

While there are several commercially available products available, HELP Line has unique requirements that prevent them from easily using such products. These requirements include leveraging UVas authentication system, managing complex merge operations, and a custom permissions structure. While possible to imitate on a commercial solution, such features are impossible to replicate.

3. System Architecture

Our system was implemented using the Django 1.8.4 web application framework on Python 3.5. Django is database-driven and follows the Model-View-Controller (MVC) paradigm. Most of our application is implemented within the confines of this framework. User-facing HTML (view) is rendered using the Django template engine, database schema (model) is specified using special Django derived classes, and the application logic (controller) is a set of Python functions callable by specified URLs. Django's object-relational mapper (ORM) manages the underlying database implementation.

The nature of the HELP Line system is that most useful operations are either querying tables from the database or adding new rows to existing tables. Consequently, the controller logic of the HELP Line system is relatively straightforward. Only a small portion of the application code is dedicated to implementing HELP Line specific algorithms. The following is a brief summary of the core components of the application:

- `views.py`: Contains application logic. Retrieves information from the database and serves information to the client via rendering HTML templates.
- `models.py`: Python classes defining the data schema. The Django ORM uses this information to create corresponding database tables.
- `forms.py`: A collection of classes that specify how the user may change model data in the database.
- templates/*: A collection of hierarchical HTML templates that specify the structure of each page in the web application.
- `urls.py`: Mapping from URL strings to controller functions in `views.py`.

While the majority of our application was developed strictly within the Django framework, it is worth noting that a significant portion steps outside of the Django paradigm. A large amount of JavaScript was required client-side to provide the dynamic user experience necessary for some pages. All pages that incorporate JavaScript extensively leverage JQuery. In addition, many also use Asynchronous JavaScript and XML (AJAX). While Django does not preclude the use of such technologies, their use is not directly supported.

3.1 Add and Edit Callers

When adding a caller, callers must be identified by a name chosen by the volunteer, an age range of either 0-18, 18-25, 26-40, 41-60, or 60+, characteristics, a gender, their story, determine whether they are a UVa student, and optional fields for call topics. The characteristics and story fields are markdown supported and thus can include a variety of styling. All volunteers can add callers through the add callers page; however, only Program Directors can edit and delete callers. When editing callers, the instance of that caller is repopulated into a form to allow for modifications on the caller details and overrides the original.

3.2 View Callers

The view callers page is the natural entry point for exploring all logged data in the HELP line system (Figure 1). This page is dual paned: the left pane displays a selectable list of all callers, and the right pane displays all logged calls associated with the current selected caller. In addition, the list of all callers can be instantly filtered and navigated with the keyboard.

As nearly all data in the database is indexed by this page, sending all data dependencies is not feasible. Instead, only the list of caller names is transferred to the client on initial page load. When a caller is selected, an AJAX request is made to the appropriate Django view. This view returns the rendered template containing all previous calls. The rendered template is inserted into the right pane using JQuery. In addition, Django is instructed to prefetch all data from the database on initial page load, speeding up the AJAX calls.

3.3 Merge Callers

Given the anonymity of callers, duplicate callers are often unintentionally created in the system. When a Program Directory discovers a duplicate caller, the two (or more) callers are merged. Every call associated with the secondary callers is moved over to the primary

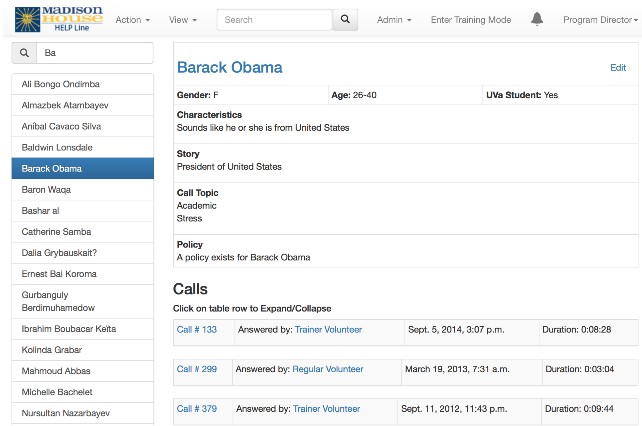

Figure 1. View callers page

caller. Additionally, the policies, characteristics, stories, and call topics of the secondary callers are appended to those of the primary caller. The primary caller retains its information about the callers name, gender, and age. The secondary callers are then deleted to finish merging. A caller cannot be merged with itself.

3.4 Add and Edit Calls

Adding calls is one of the most-used features of the HELP Lines system. Most of the work of volunteers is ingesting calls to HELP Line. Consequently, a large amount of time was spent optimizing its layout. It was discovered that volunteers would frequently browse the caller catalog (callers and their associated calls) while adding a call to more effectively identify repeat callers. To simplify this process, a dual pane layout was implemented containing both the call form and caller catalog. The left caller catalog pane can be quickly hidden with a button toggle.

The implementation of the caller catalog pane is nearly identical to that of the list callers page. The call form, however, has several notable features. First, the caller and volunteer selectors automatically update with type-ahead prediction. That is, the database is searched in real time for strings containing the typed substring. This feature is powered by AJAX calls to an endpoint established by a Django plugin. In addition, once a caller is selected, the caller policy for that caller is immediately displayed. The volunteer is given the ability to view previous alternatives suggested to the selected caller, and any number of alternatives may be added. The dynamic creation and deletion of alternative forms is accomplished using JQuery and Django Formsets.

Calls can be edited by the volunteer that took the call or Program Directors who have permission to edit and delete all calls in the system. Edit call is implemented simply by repopulating the instance of the call into a form and overriding the original. Deleting a call deletes all alternatives and comments associated with the call.

3.5 Move Calls

Given the magnitude of callers in the database, it is easy for Volunteers to associate calls with the incorrect caller. Program Directors can move calls they believe are associated with the incorrect caller to another caller. Comments are also moved to the correct caller.

In the move calls page, the Program Director first specifies the call(s) to move. The user can search for a call with the call id or the call summary, or both. The call to move field allows multiple calls to be specified. The Program Director specifies the destination caller, where the call will be moved, in the move to caller field. Since all calls are associated with a caller, a call is moved by changing the foreign key relationship to the destination caller.

3.6 Alternatives

Alternatives are recommendations of the HELP Line staff to its callers. They are associated with a call, and are classified by an alternative type and a description of why the particular type is pertinent. The database table containing these alternatives cannot be accessed in the application directly; rows of this table are only visible when viewing the associated call. Alternative Types specify the general nature of the alternative, including if it is a referral to an outside agency. The set of Alternative Types is editable by Program Directors.

3.7 Call Topics

Call topics is an optional field for a caller. A call topic is a classification of an issue faced by the caller. These call topics allow volunteers to more precisely search the system for a caller or a group of callers. A caller has a many-to-many relation with call topics. In order to implement call topics as a many-to-many field, callers and call topics are two different entities in the database. Call topics is added as a many-to-many attribute to a caller.

3.8 Policy Updates

Policy updates is a feature that informs users of new caller policies or rules. A policy update is automatically created when a user edits the policy for a caller. In addition, a policy update can be created through the Admin navigation bar by Program Directors and requires information on which caller the policy applies to, the subject of the policy, and text explaining the policy. Only Program Directors can delete policy updates. Recent policy updates appear on the homepage when users first logs in, and all policy updates can be viewed by all volunteers from the view button on the navigation bar.

3.9 Hangups

HELP Line records all calls, even when the caller hangs up. Hangups are recorded with a click of button. The system records the time that the call occurred and the volunteer who recorded it. Hangups and calls are considered different entities on the database layer. Hangups can be deleted by Program Directors.

3.10 Comments

Program directors and volunteers can comment on calls to provide feedback and improve call quality. Adding, editing, and deleting comments occurs through AJAX calls to ensure the page is not refreshed. Comments can be threaded one level deep. Instead of using Django to automatically generate forms, they were implemented manually to allow more flexibility.

3.11 Notifications

Notification alerts volunteers when someone has commented either on a call they recorded or a comment they made. Program Directors frequently comment on calls to provide feedback to volunteers on what they did well or could have done better while taking a call. This feature allows volunteers to know when a relevant comment has been made and on which call or comment it was in response to.

A bell icon was added to the navigation bar. Every time a page is loaded, the number of new notifications for a volunteer is calculated from when a volunteer last clicked the notifications button and then is displayed in red on the bell icon. When the notification button is clicked, a bootstrap popover appears and displays a table with a link to a caller page for each notification. This table contains notifications for every comment for which the call or the parent of the comment is the volunteer logged in. After the notifications are viewed, the timestamp of last viewed notification is updated to show zero new notifications.

3.12 Search and Advanced Search

In the old HELP Line system, repetitive information was common due to a lack of search capabilities. Search queries the database to retrieve data associated with the provided keywords. The general search filters through all calls, callers, and volunteers with the given keyword. All relevant calls, callers, and volunteers are returned. For advanced search, volunteers may supply call summary keywords, caller attributes, the volunteer who recorded the call, date range, and any call topics associated with a caller. Advanced search only returns calls matching all the given keywords.The user may supply all six fields or as few as one. All volunteers are able to search. Searching is not logged by the system.

3.13 Training Mode

Training mode is a feature implemented to allow the customers to train new volunteers without exposing them to sensitive data in the production system. Trainee group members can still log in and become familiar with its use. Program Directors and Trainers control the members of this group. Only Program Directors and Trainers can switch between normal mode and training mode. Volunteers enter the system in normal mode, and Trainees enter in training mode. Callers, calls, hangups, and policy updates are not transferred between the two modes.

Initially, the design called for two separate databases, but implementation was too complex as much of the data would need to be shared across databases. In the end, each of the four models that would not transfer between modes were given a flag to indicate whether they were training data. In addition, volunteers were given a flag to indicate whether or not they were training data. When a request to read or write any of the four non-transferable models is received, only those whose training flag matched the volunteers training flag are presented.

3.14 Logs

The logs page is only shown to Program Directors. A log is created for almost all actions in the system and contains information about what action occurred, the date and time of the action, and the volunteer associated with the action. A helper method is called at the end of almost every controller method to create the necessary log object. A Program Director can regularly click a button on the logs page to clear the logs and delete unnecessary log data.

3.15 Reports

Reporting was the largest new feature requested by the customer. Reports on calls, callers, and call topics can be generated by Program Directors through the admin navigation. A CSV is generated based off parameters such as year, calls, callers, or call topics. Reports on calls can be generated based on the day, hour and day of the week of when the call was taken, and reports for callers can be generated based off characteristics such as gender, age, and based off the number of UVa students. Reports for call topics can be generated based off the number of calls with a particular call topic, and the percentage of callers to call topics. A bar chart is then rendered off the CSV utilizing the Data-Driven Documents (D3) javascript framework, and a frequency tip library that indicates the value of each bar on the chart. All data on calls, callers, and call call topics can also be exported as CSV for further analysis.

3.16 User Management

Program Directors can create a user by using a form to input information: first name, last name, email, username, password, group, gender, and graduation year. The email field is important as the system can send an email to the user with the password using the SMTP information specified in the settings file.

Program Directors also have access to a page which lists all users. Due to the number of users, the page loads this data via AJAX. In addition, Program Directors can filter, edit inline, and delete the users listed on this page. Bulk operations are supported. When the Program Director submits the request to bulk edit users, an array of volunteer ids and the desired action is submitted to the server. Deleting a user also removes all content with a foreign key relationship to user such as the users calls, comments, logs, and hangups. As such, disabling a user is provided as an alternative which toggles a flag to determine access without deleting all associated content.

3.17 Netbadge Authentication

NetBadge is a service used by the University of Virginia (UVa) to authenticate on either a username/password combination or a digital certificate on the client machine. User information is sent to a remote server where it is authenticated. Then, the remote server sends back a response that puts the authenticated users computing id on the local servers REMOTE_USER variable. A computing id is a unique identifier given to a UVa student or employee for authentication purposes.

Django does not natively support this form of authentication. A link was added to the login page that redirects the user to an index.php file in the static folder. This file is protected by a .htaccess file that first redirects to NetBadge authentication routines. Once NetBadge has authenticated the user, index.php is called to place the validated computing id into the database along with a random fifty character string which is added for security purposes. Once that information is in the database, index.php returns control to a Django view, passing the random string as a URL parameter. In the Django view, the database is queried for a login token that matches the given string. Once found, it logs in the volunteer with the matching computing id.

4. Challenges and Design Decisions

4.1 Python 3 and Dropping Python 2 Support

The system originally was designed to only work on Python 2. However, after several months, it was found that the system would need to run on Python 3. One of the required changes included adding a try-except statements ensure version-specify imports did not cause crashes. In addition, the continuous integration test system was changed to use Python 3. It was eventually decided that Python 2 support would be dropped because it would allow the team to focus more on features rather than testing the system for both versions of Python. In addition, reporting on call topics uses Python 3 set functionality.

4.2 Data Transfer, Corruption, and Time Zones

After development was completed, the old data needed to be transferred to the new system. A SQL script parses the old systems tables and inserts the data correctly into the new systems tables. This system creates a one-to-one relationship between volunteers and users to leverage Djangos built-in Group and User functionality. The previous system did not, which means that appropriate user information was loaded into the auth_user table, including a copy of the hashed passwords, before the volunteer entries were created and associated. This allows volunteers to keep the same password they used for the previous site. Our system tracks more fields than the previous system and thus the transferred models will have NULL or default values for these fields (i.e. graduation_year for volunteers).

Another important data change is the permission schema. The old system used an access field assigned a number while the current system uses four Django Groups: Program Director, Trainer, Volunteer, and Trainee. Since the Trainer and Trainee permissions are new in the current system, no corresponding fields existed in the previous system. As such, members were added as either Volunteer or Program Director permissions. The old system had a table for call topics, the table did not appear populated as most of the entries were empty or unclear. As well, the client indicated that the system was not designed with this functionality, so this table was not merged over. Comments, hangups, and logs were transferred in a fairly straightforward manner.

Some corruption was discovered in calls table of the original database. Each call is associated with both a volunteer and a caller. The original system had 240 calls that were missing either their volunteer or caller foreign key id. The volunteer key was most likely lost by deleting them from the system and improperly cascading the delete, leaving no information about the original volunteer. It is unlikely that this information can be recovered. The caller key was most likely lost when two callers were incorrectly merged. Specifically, it is believed that these calls were copied and associated correctly, but the old call had a NULL primary key. Much of this corruption was solved because only 40 calls were missing their caller primary key. A dummy volunteer was created to be associated with the calls that were missing their volunteer primary key, and the 40 calls left without a caller were not transferred.

Due to issues in which Django could return or aggregate on only date components as the result of a filter, some MySQL code is specified explicitly inline. This code was used to return a table containing only the relevant date components for reporting, and, while it does limit database interoperability, no other solution was easily found. There may exist other database backends that work, but only MySQL is officially supported. Any transfer of this system risks breaking or violating one of these requirements through use of either a different Python version or different database management system. Specific instructions to transfer or install the system are given in the installation instructions.

5. Workflow

The purpose of the system is to aid volunteers while recording calls to HELP Line. There are four groups of users that will use the system: Volunteers, Program Directors, Trainers, and Trainees.

Volunteers frequently add calls. If the caller does not exist, the user fills out the add caller form. Volunteers can also look at past calls that they have been made. In the current Helpline system, the Volunteer can see their most recent calls and policy updates on the home page. In addition to checking most recent calls and policy updates, the Volunteer may want to find a specific caller. If a Volunteer cannot remember the exact name of the caller, the Volunteer can use the search bar to find keywords in the callers name or the callers story. The Volunteer can also search the callers name in the list callers page to view all of their previous calls. While viewing the previous calls, the Volunteer can edit their recorded calls and comment on any call.

Program Directors have similar workflow to Volunteers. Many of the PD actions affect the entire system. PDs have permission to add, edit, and delete alternative types, policy updates, call topics, and Volunteers. They are also able to merge two existing callers, move a call from one caller to another, view logs on the system, and generate reports.

The system also includes training mode. Training mode is a sandbox where new Volunteers, called Trainees, can practice logging calls and using the system. All Trainees automatically enter the system in training mode. PDs give existing Volunteers the permission to become a Trainer to mentor the Trainees. When a Volunteer becomes a Trainer, in addition to their existing functionality on the main system, the Trainer can enter training mode. When a Trainer is in training mode, they have limited permissions, similar to a PD. In training mode, a Trainer can add a policy update, merge callers, and move calls.

6. Results

The new HELP Line system is an extended re-implementation of the previous system. The most important metric of comparison is the relative increase in stability of the new system. The old system, even during our own development, was only intermittently functional. Despite the short time our system has been in production, it is more thoroughly tested and has so far proven to be more stable than the prior system. No customer feedback was related to system instability.

The performance improvements of our system are difficult to quantify. When loading similar pages in both systems, the new system is marginally faster, though the non-scientific nature of such tests makes conclusions difficult to extrapolate. Our system includes several features that significantly speed up the work flow of volunteers. Small, productivity increasing features abound: automatically populating call fields after adding a caller, automatically notifying the user of policy updates, instantly filtering the list of callers, more advanced search features, comment notifications, and extensive use of type-ahead field population, etc. In our testing, the current HELP Line volunteers provided positive feedback regarding these features.

When conducting usability testing, volunteers, who possessed no prior knowledge of the system, were able to more quickly execute common work flows. The customer expressed appreciation for new features added such as searching, reporting, policy updates and split screen view when adding callers. In addition, our system produces statistics and exports raw data improving Madison Houses ability to gather analytics about HELP Line. This is likely the single largest functional improvement of the system.

7. Conclusion

We successfully developed an improved re-implementation of the HELP Line call log system. Our system, in addition to offering functional improvements, was faster in practical use due to a myriad of time-saving features. The HELP Line volunteers have expressed positive feedback, and have noted a definite improvement over the previous system.

8. Future Work

Future work includes improving and expanding newly added features like search and reporting. Reporting is limited to some number of fixed parameters for analysis. In general, future work could focus on database optimizations. While developing the current system, the performance ramifications of Django queries were not considered. The main focus of this project was on the implementation of functionality as dictated by the customer thus future work should have a bigger emphasis on the user interface and user experience. Bootstrap was utilized as the HTML, CSS, and JS framework which provided faster development but led to poor design decisions as the team relied entirely on this framework to handle both UX and UI. Lastly, an online chat system could be implemented.

9. Acknowledgements

We would like to thank Madison House, HELP Line, Rachel Winters and Sruthi Poduval for their support and help in the completion of this capstone project.