Social CheatSheet: An Interactive Community-Curated Information Overlay for Web Applications

LATON VERMETTE, Simon Fraser University
SHRUTI DEMBLA, University of Waterloo
APRIL Y. WANG, Simon Fraser University
JOANNA MCGRENERE, University of British Columbia
PARMIT K. CHILANA, Simon Fraser University

Users can often find it difficult to sift through dense help pages, tutorials, Q&A sites, blogs, and wikis to locate useful task-specific instructions for feature-rich applications. We present Social CheatSheet, an interactive information overlay that can appear atop any existing web application and retrieve relevant step-by-step instructions and tutorials curated by other users. Based on results of our formative study, the system offers several features for users to search, browse, filter, and bookmark community-generated help content and to ask questions and clarifications. Furthermore, Social CheatSheet includes embedded curation features for users to generate their own annotated notes and tutorials that can be kept private or shared with the user community. A weeklong deployment study with 15 users showed that users found Social CheatSheet to be useful and they were able to easily both add their own curated content and locate content generated by other users. The majority of users wanted to keep using the system beyond the deployment. We discuss the potential of Social CheatSheet as an application-independent platform driven by community curation efforts to lower the barriers in finding relevant help and instructions.

CCS Concepts: • **Human-centered computing → Social content sharing**; • **Human-centered computing → Collaborative content creation**; • *Human-centered computing → Social recommendation*

KEYWORDS

Community-curated help; social help; software learning; software help;

ACM Reference format:

Laton Vermette, Shruti Dembla, April Y. Wang, Joanna McGrenere, and Parmit K. Chilana. 2017. Social CheatSheet: An Interactive Community-Curated Information Overlay for Web Applications. *Proc. ACM Hum.-Comput. Interact.,* 1, CSCW, Article 102 (November 2017), 19 pages. https://doi.org/10.1145/3134737

1 INTRODUCTION

Using an unfamiliar feature-rich application can often be daunting for users. Although the web offers thousands of help resources, locating the most relevant and useful instructions and examples pertinent to a task can be frustrating and time-consuming. Instead of sifting through verbose documentation, lengthy text and video tutorials, blogs, wikis and other help pages, many users instead prefer social forms of help, such

Author emails: Laton Vermette (lvermett@sfu.ca), Shruti Dembla (s2dembla@uwaterloo.ca), April Y. Wang (ayw7@sfu.ca), Joanna McGrenere (joanna@cs.ubc.ca), Parmit K. Chilana (pchilana@cs.sfu.ca)

2573-0142/2017/11-ART102 $15.00 Copyright is held by the owner/author(s). Publication rights licensed to ACM.
https://doi.org/10.1145/3134737

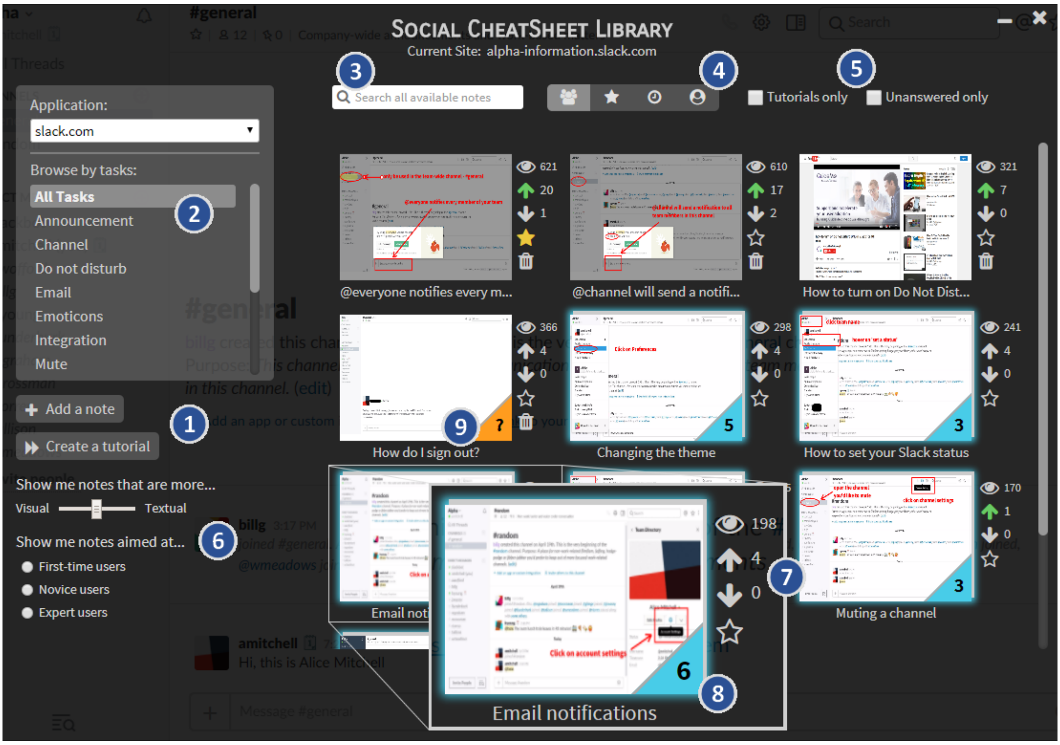


Fig. 1. Social CheatSheet's expanded "grid" view shows all relevant curated notes and tutorials. Users can (1) easily add their own curated notes or tutorials, as well as filter content by (2) existing task tags or (3) natural language search queries. They can (4) sort the display by popularity, date, owner, and bookmarks, (5) limit the display to only tutorials or unanswered questions, and (6) further filter the content by changing their learning style preference and expertise level. Each note or tutorial shows (7) the number of views and votes, and users can easily identify (8) tutorials by a blue triangle and (9) unanswered questions by an orange triangle.

as asking a question over email [37], posting questions on their social network [33], or seeking help from software Q&A forums [30,42]. Through these social mediums, users can obtain direct task-relevant *curated* instructions from other users or experts, such as a paraphrased explanation, an annotated screenshot, or a snippet of content extracted from an existing help resource (*e.g.*, a highlighted tutorial step, a relevant video frame).

Despite the benefits of curated help, the actual curation process can be cumbersome for the help provider, as multiple tools are required to capture the screen, annotate, save, and share instructions. Furthermore, these help snippets can be difficult to re-discover once they are buried in a private inbox or messaging service or on a Q&A forum [22]. Recent developments in commercial and research applications for note-taking and personal information management (*e.g.*, *EverNote* [14], *CheatSheet* [46]) offer the possibility to streamline the process of taking screen-captures, adding annotations, and saving help content for personal reuse. Tools such as CheatSheet [46] further provide mechanisms for users to easily retrieve their own annotated "cheat sheets" as an overlay atop the relevant application. What if these personal cheat sheets and notes could be augmented with content from other users learning or troubleshooting the same application? How would users want to contribute and locate relevant content? In what contexts would this approach be most helpful?

In this paper, we investigate the concept of social curation of software help content by developing a novel platform, *Social CheatSheet*, that overlays relevant community-curated instructions and multi-step tutorials atop any web application (Fig. 1) and offers an easy curation interface for adding and editing content. Although Social CheatSheet takes inspiration from several existing approaches in human-computer interaction (HCI) and computer supported cooperative work (CSCW), such as community-based help systems [7,27,28,31] and social annotation techniques [21,47], our approach is unique in that 1) it allows users to easily generate and aggregate task-focused curated instructions and multi-step tutorials using a combination of their own annotated screenshots and snippets of web-based help resources; and, 2) it makes it possible to leverage the expertise of the user community within the application to quickly find an answer to, "*What is the best instruction or tutorial for me to learn task X in this application?*" Importantly, users can retrieve and edit other users' curated help on any web application by installing a browser extension, bypassing the potential barrier of relying on application owners to integrate the community help system [6].

To inform the design of our system, we first developed a minimal prototype using the original CheatSheet [46] idea and conducted a formative study with 10 users to assess concept usefulness and generate the design requirements for a complete system. From there, we designed new community-based features, step-by-step tutorials, privacy features, personalization options, and natural-language search capabilities in Social CheatSheet to facilitate community-based curation and in-application help retrieval.

We evaluated Social CheatSheet by carrying out a weeklong deployment with 15 users to assess in context: its usefulness and usability, how well Social CheatSheet supports the design requirements we derived, and the likelihood for adoption. The majority of users were favorable towards the system, including wanting to keep using the system beyond the deployment. Despite a small community of users and limited help content, two-thirds of users revealed that they discovered something new during the deployment period just by browsing other users' curated content. While a third of the users said they would likely not add their own instructions and tutorials, given the ease of using the curation features, over half of the users were neutral and not opposed to adding content if the community were to grow in the future.

Our main contribution is in the design of a novel easily-accessible platform that can serve as a modern community-generated minimal manual [5] for learning unfamiliar tasks within a web application. Additionally, we contribute initial insights into how such a community-based help system can be used to add and retrieve content in the context of using a real-world application. While CSCW has a history of designing for knowledge and expertise sharing, our paper offers lessons for designing and evaluating what Ackerman et al. [2] call a *collaboratively constructed information space*. In this case, Social CheatSheet serves as a knowledge repository that leverages collaborative features to make the knowledge relevant and easier to discover. While our primary contribution is in the software help and learning domain, our work offers broader insights into how a personal curation system can be used to invent a new collaborative information space that serves both personal and community information needs and that users find easy to use and useful for contributing and retrieving content.

2 RELATED WORK

Our research builds upon several innovations in HCI and CSCW, but also differs from them in significant ways, as described below.

2.1 Community-based Software Q&A

Social help in the form of community-based Q&A has been increasing over the years, with well-established communities for both open source and commercial software in the last decade [28,30,41,42]. Our in-application help approach is most similar to systems such as *LemonAid* [7] and *IP-QAT* [31] that focus on delivering community-driven Q&A-style help based on application context. Like these systems, our motivation was also to reduce the back and forth between different applications and web-help resources that users often experience and to allow users to leverage each other's experiences and expertise.

However, unlike these Q&A tools, Social CheatSheet's primary focus is on helping users locate curated task-based visual instructions and step-by-step tutorials available on the web or created by the user community. Furthermore, Social CheatSheet users can customize the retrieved content by filtering by task, expertise, and learning style preferences and have access to a personal space for bookmarking useful content or adding their own private curated content. Finally, since Social CheatSheet is a browser extension that anyone can install, it does not have to be adopted and integrated by application owners, which is known to be a potential barrier [6] for embedded community help systems.

2.2 Community-based Software Tutorials

Another class of relevant research has explored how the user community can be involved in enhancing tutorial content to help users make better use of unfamiliar features and reach new skill levels when learning software. For example, the *CADament* [29] system leverages the idea of gamification to engage users in a multi-player tutorial system and promote collaborative learning experiences for a CAD application. Despite the benefits of software tutorials, users often have difficulty locating relevant content within tutorials and it has been shown that users can benefit from community-based annotations that highlight key steps [23] or comments that tightly integrate with different parts of the tutorial [4]. Users can also benefit from seeing alternative and/or additional demonstrations and explanations within the videos [27].

Although Social CheatSheet builds on the concept of community-authored tutorials, it does not generate video or documentation-style tutorials (based on feedback from the formative study); rather, it offers a unique tutorial authoring feature to segment a task as a sequence of individual annotated step-by-step screenshots and allows users to attach external help snippets (*e.g.*, a video frame, FAQ, or forum answer) to enhance the content. These tutorials are tightly integrated within an application and allow users to focus and easily navigate between individual curated tutorial steps without having to consult external resources. Furthermore, *any* future user can add, extract, or edit any part of the tutorial to improve the instructions for her own use or for sharing with the user community and can personalize the retrieval of tutorials by specifying different tasks, level of expertise, and learning preferences.

2.3 Community-based Software Help Recommendations

Several HCI systems have also investigated the idea of recommending help based on past actions of other users. For example, the *HelpMeOut* [20] system suggests solutions to help debug error messages based on what other programmers have done in the past. Similarly, *CommunityCommands* [32] contributes a collaborative filtering algorithm that can make new command suggestions based on actions of other users in a CAD application. More recently, the *DiscoverySpace* system [16] explores the idea of suggesting task-level actions to Photoshop users by harvesting data from user communities.

While Social CheatSheet captures basic user actions (*e.g.*, clicking, scrolling, typing) for generating tutorial steps, it does not detect individual commands or application-specific features. Furthermore, none of these interactions are stored or used later to affect the retrieval algorithm. To retrieve relevant content, Social CheatSheet instead relies on task tags and explicit contributions from the user community (e.g., upvotes, downvotes, views, and other metadata). Additionally, when someone views a note or a tutorial step, Social CheatSheet can suggest notes with related or alternative explanations on the same topic (or the next steps if the note is part of a tutorial). These recommendations are based on the visual and textual similarity of the content and the related metadata, not analytics collected about user activity.

2.4 Augmenting Web Page Content and Functionality

Although not in the domain of software help, many HCI systems have explored how users can augment web page content and functionality and even share these augmentations with other users. Some of these systems (*e.g.*, [21,47]) add rich social text annotations, such as comments and mark-up, as a layer overtop different elements of a web page. However, these systems largely employ page-based annotation tied to the content of the page (*e.g.*, sentences and paragraphs). In contrast, Social CheatSheet uses a task-based annotation approach: instead of adding annotations on specific page elements, it inserts a library of annotated snippets of screen shots tied to different application subtasks (generated by users or curated from existing web-based content).

In addition to page-level annotations, some systems have explored ways of collecting and summarizing web-based information related to a topic using automatic approaches and templates (*e.g.*, [13]) or using socially annotated schemas (*e.g.*, [24]). Social CheatSheet has a similar goal of consolidating and organizing content from a variety of existing web help resources, but these resources need not follow similar templates or schemas to be incorporated into user-generated tutorials. Instead, Social CheatSheet is driven more by a deliberate community effort to curate and improve a shared, growing collection of software help tutorials than by its ability to automatically capture web-based content.

Other ways of augmenting web pages include systems that allow users to share detailed browsing activity with friends and even sync browsing sessions (*e.g.*, *PlayByPlay* [49]) or adapt the structure of specific pages based on community redesigns (*e.g., CrowdAdapt* [36]) and accessibility needs [44]. However, Social CheatSheet aims not to interact with its underlying web page functionality or adapt the interface, beyond inserting its unobtrusive, initially-hidden layer of help resources. The main goal is to make it easier for users to discover curated help content for the application currently being used.

2.5 Re-finding Software Help

Lastly, an emerging class of systems has focused on helping users re-find previously used content. For example, *InterTwine* [15] supports task-based re-finding by tying together actions in a software application and browser histories and search queries. The CheatSheet [46] system allows users to take visual screenshots and save snippets of help content in a personal memory aid atop a web application. Studies of InterTwine [15] and CheatSheet [46] both indicated that a natural extension would be to explore community-based re-finding of help content. But, how a community-based re-finding and sharing platform could be designed and whether users would even find it useful in the context of their learning tasks was not investigated—this is a key focus of Social CheatSheet. In particular, Social CheatSheet extends the in-application memory aid concept [46] by connecting potentially many users of the same applications through a shared community-curated "cheat sheet," which they all draw from and contribute to atop an application.

In summary, Social CheatSheet builds upon and significantly extends prior HCI and CSCW innovations by offering a unique user-centered in-application social platform to help a user automatically discover, *"What is the best instruction or tutorial for me to learn task X on an application?"*

3 FORMATIVE STUDY

The goal of our formative study was twofold: (1) to preliminarily examine the usefulness of overlaid help content created by other users, and (2) to derive design requirements for our social curation platform. To address these goals, we first developed a minimal prototype using ideas from CheatSheet [46] such that users would see other users' cheat sheets and notes (instead of their own content) atop a given web application.

3.1 Methodology

We carried out an observational study and follow-up interviews with 10 adult participants (4M/6F) between the ages of 19-64. We recruited participants through university mailing lists, word of mouth, and flyers placed on bulletin boards on a university campus. We used the web-based *YouTube Video Editor* as our test site and asked participants to imagine that they were returning from a vacation and wanted to put together a short compilation of their best vacation video clips and pictures prior to catching their flight home. Only one of our participants had used the video editor before (but had not used the specific features that we tested). We provided participants with a written outline of the study task and asked them to attempt as many of the steps as they could in a 30-minute time slot. We asked them to use our prototype to find help when they needed it.

Three members of our research team used CheatSheet [46] to create various annotated notes on using the video editor. All of these notes were made visible to each study participant's overlaid cheat sheet as if they were coming from a larger community of contributors. We ensured that every component of the study task was explained in one or more of these notes and, for completeness of the help content, included additional notes on other major features of YouTube.

We opened the video editor's *Dashboard* page in the active browser tab at the beginning of the study and logged into a test account. We provided an example video displaying the "model outcome" with the features described in the scenario in a separate browser tab, which the participants watched before beginning the task and could refer back to at any time.

We asked participants to think-aloud as they performed the task, while one researcher observed and noted where participants encountered breakdowns or sought help. We also made screen recordings to corroborate our observations. We followed-up with a semi-structured interview to ask participants about their experience of using our prototype for finding help, how it compared to other help mediums, and their perceptions of seeing content from other users. We analyzed the data from our observations and interviews using an inductive analysis approach [43] to answer our two key research questions.

3.2 Key Findings

3.2.1 Usefulness of In-Application Curated Help

Since all of the participants were new to using the YouTube video editor features that we tested, we observed that they struggled in locating the task-relevant functionality and consulted help from our prototype every few minutes. In terms of common breakdowns that trigger need for help [39], our participants' struggles appeared to be mostly procedural (*"how do I upload my video?")*, navigational *("where is my video in the timeline view?")*, and interpretive (*"why did my annotations disappear?").*

To find help from our prototype, participants clicked on the pre-populated task tags about 5.9 times on average and issued 4.6 queries on average using our built-in search feature. Although our participants were using our prototype for the first time to learn an unfamiliar application, it was encouraging to see that, overall, they found it helpful to view other users' curated cheat sheets overlaid within the application: "*the best thing about it [prototype] was the overlay so I didn't have to interrupt my train of thought to go on Google and search for something most relevant." (P06)*

Other benefits cited by users were being able to view help visually in screenshots (versus sifting through FAQs or written instructions), and being able to see key instructions annotated based on what other people found useful (versus watching a long video or reading through documentation).

Despite the overall positive response, only 4 of the 10 participants fully completed the task in the given time limit. Our findings pointed to several limitations with our prototype in helping users search for relevant curated content, assess the relevance of retrieved content, and apply the help content to the current task.

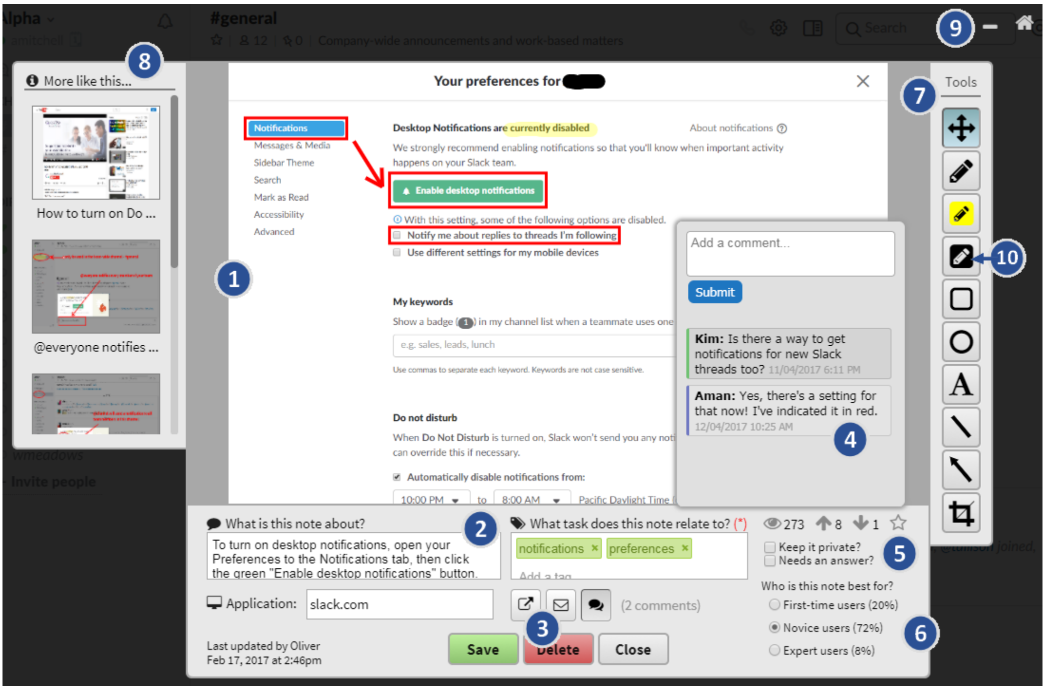
3.2.2 Design Requirements

Using data from our observations and feedback from participants, we synthesized key limitations and requirements for improving the design of in-application curated help.

***Community-oriented features*:** Users who struggled to determine the relevance of a note, often indicated that they would want to know if other users found it useful: "*I feel like if someone is looking for an answer, I think looking at number of views or popularity would help them make a decision on which one to look at first" (P08).* Users also wanted to see more features for participating and learning from the community, such as receiving recommendations for related content (*e.g.*, alternative explanations), adding comments, and requesting clarifications.

***Step-by-step tutorials*:** Participants stressed that viewing individual notes was not always helpful when a task required multiple steps. The "big picture" was often lacking or participants were unsure where to go next after completing a step. Several users made requests for viewing help content as step-by-step tutorials: "*At least for me, [what I need is] the 'first thing you need to do'. Maybe one paragraph that says 'this is the process you use to create a video'... That's actually a useful summary, and now I can actually go in and figure those things out. My feeling is if there's a 'do this first, then read this', it might be more helpful" (P05).*

***Personalization options*:** Participants indicated that the retrieved notes were often either "*too general*" or targeted towards more experienced users. Since most of our participants were new to the test application, they wanted to see something relevant to first-time users. Similarly, some users also wanted a quick way to see notes that were more visual while others wanted to see more textual explanations. Participants noted that this was a limitation of most modern help systems that usually use a "*one-size fits all*" approach for providing help.



More like this...
How to turn on Do ...
@everyone notifies ...

Your preferences for [redacted]

Notifications
Messages & Media
Sidebar Theme
Search
Mark as Read
Accessibility
Advanced

Desktop Notifications are currently disabled

About notifications

We strongly recommend enabling notifications so that you'll know when important activity happens on your Slack team.

Enable desktop notifications

With this setting, some of the following options are disabled.

□ Notify me about replies to threads I'm following

□ Use different settings for my mobile devices

My keywords

Show a badge (1) in my channel list when a teammate uses one

e.g. sales, leads, lunch

Use commas to separate each keyword. Keywords are not case sensitive.

Do not disturb

When Do Not Disturb is turned on, Slack won't send you any noti
can override this if necessary.

☑ Automatically disable notifications from:

10:00 PM to 8:00 AM Pacific Daylight Time

Add a comment...

Submit

Kim: Is there a way to get notifications for new Slack threads too? 11/04/2017 6:11 PM

Aman: Yes, there's a setting for that now! I've indicated it in red. 12/04/2017 10:25 AM

What is this note about?

To turn on desktop notifications, open your Preferences to the Notifications tab, then click the green "Enable desktop notifications" button.

What task does this note relate to? (*)

notifications × preferences ×

Add a tag

Application: slack.com

(2 comments)

273 8 1

□ Keep it private?
□ Needs an answer?

Who is this note best for?

○ First-time users (20%)
● Novice users (72%)
○ Expert users (8%)

Last updated by Oliver
Feb 17, 2017 at 2:46pm

Fig. 2. The curation interface lets users (1) capture screenshots of the application or external help resources, (2) attach metadata such as textual explanations and task tags, (3) share notes via email, (4) participate in community discussions on each note, (5) mark notes as private or in need of an answer, (6) specify a suitable expertise level, (7) annotate the screenshot with visual markers, (8) view recommended similar notes or next steps (if available), and (9) minimize the current view (without exiting). The censor tool (10) can be used to permanently black out any sensitive information.

***Privacy controls*:** Participants also expressed privacy concerns—they did not want to see someone's private information (*e.g.*, account numbers) by mistake if it appeared as part of the help content: *"Since they [a user] made this so everyone sees it... Doesn't it mean I might see someone's account numbers or passwords in [the screenshot], that they put in by accident?" (P04).* This was also a concern about participating in the curation platform in the future: users wanted to be able to mark things private when necessary or hide their content on a screenshot.

***Flexible natural-language search options*:** A key problem for all users was using our prototype's rudimentary search feature—in fact, every participant tried to search by keyword at least once and found the current search mechanism to be too restrictive. Most participants noted that as the curated content grows, a more flexible natural language search engine would be necessary to locate relevant content.

In summary, we learned that users were positive about the usefulness of help curated by other users in the context of learning a new application, but that some key interaction capabilities would be needed in our social platform to help people add and locate task-relevant instructions. These are captured in our design requirements.

4 USER INTERFACE DESIGN OF SOCIAL CHEATSHEET

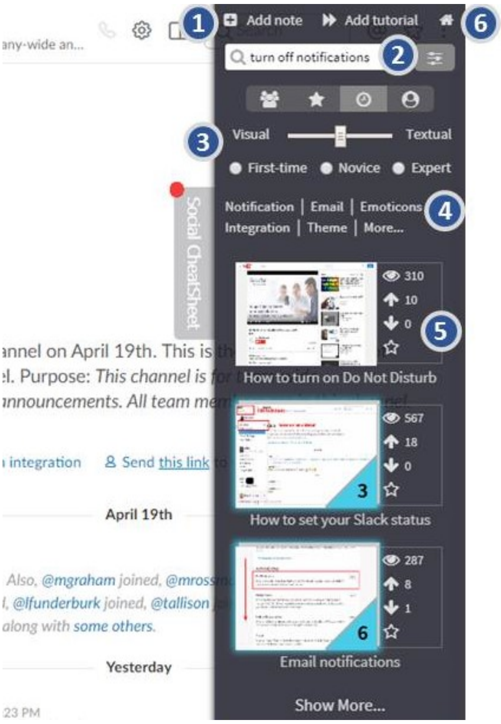
Based on our formative study, we designed *Social CheatSheet*, a novel social platform that allows many users to curate, collaborate, and discover relevant instructions and tutorials atop any web application. Our platform is built upon an extension of the personal note-taking application CheatSheet [46] and includes several new features for community participation and more targeted searching, browsing, and personalization options.

4.1 Adding Curated Content

Social CheatSheet offers a number of ways for users to add their own curated content. By default, any note or tutorial added to Social CheatSheet is public (unless explicitly marked as private). Any content can be edited by any user in a wiki-style approach, and the system displays the username of the contributor in the editor.

***Adding a single note or instruction*:** Building on the original CheatSheet [46] concept, when a user clicks “Add a Note” (Fig. 1.1 or 3.1), a screenshot of the current page is captured, and an overlay containing a curation interface in the editor view appears overtop of the page (Fig. 2.1). This interface displays the screenshot in an editor that can be annotated with a number of standard editing tools (*e.g.*, highlighting, cropping, and drawing boxes/arrows) (Fig. 2.7). Below this is a bar with several curation tools, such as adding a description to the note and tagging it as relevant to a specific application or task (Fig. 2.2) and the intended expertise of the content (Fig. 2.6). The Application field is automatically populated with the current domain name. When the note is first saved, the additional metadata attached are the creation timestamp, the current user, and the current URL.

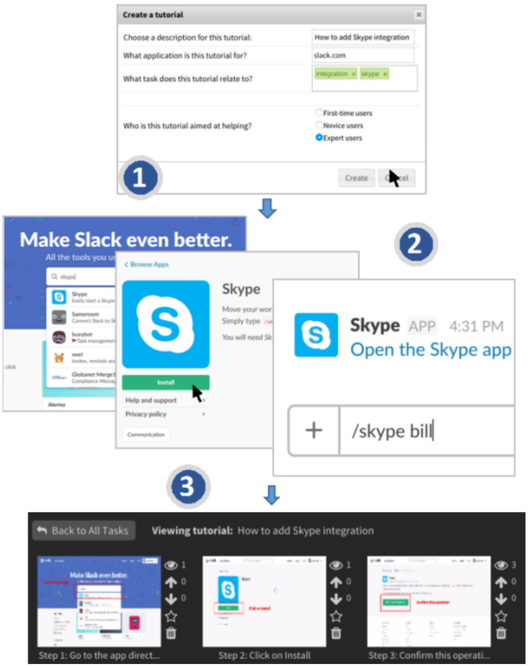
***Adding multi-step tutorials*:** To address the identified need for better organization and flow of help content during longer step-by-step tasks, Social CheatSheet offers a tutorial-generation feature. The *Create a*



annel on April 19th. This is th
l. Purpose: This channel is for
announcements. All team mem

Also, @mgraham joined, @mross
l, @lfunderburk joined, @tallison jo
along with some others.

Fig. 3. The Social CheatSheet drawer appears on the right side when a page is loaded. In the drawer, a user can (1) quickly add a new note or tutorial, (2) search by keywords, (3) sort by visual content or expertise, (4) filter content by a specific task, (5) see # of views and votes and bookmark; (6) open the full grid view.



Create a tutorial

Choose a description for this tutorial: How to add Skype integration

What application is this tutorial for? slack.com

What task does this tutorial relate to? integration × skype ×

Who is this tutorial aimed at helping? ○ First-time users ○ Novice users ● Expert users

Create Cancel

/skype bill

Viewing tutorial: How to add Skype integration

Step 1: Go to the app direct...

Step 2: Click on Install

Step 3: Confirm this operati...

Fig. 4. When creating a tutorial, users (1) first specify a title, task tags, and audience. The application field defaults to the current domain. (2) Next, they perform the task as they normally would and screenshots are automatically captured in the background (including on other visited sites). (3) Captured individual steps are displayed in the grid and automatically numbered sequentially and can be edited and deleted as needed.

Tutorial (Fig. 1.1 or 3.1) feature is a special case of adding and grouping multiple notes together—a sequence of unedited notes to be captured automatically while a task is performed, building on auto-templating first introduced in CheatSheet [46]. Users can additionally attach helpful tips and snippets from other websites during the tutorial creation process.

As an example, consider Bob, a manager, who has set up his team named Alpha on *Slack.com*, an instant messaging platform for team communication and collaboration. He wants to quickly explain how to integrate Skype with Slack to his team. He clicks on the "Add a Tutorial" feature that brings up a dialog box (Fig. 4.1) where Bob can specify the relevant metadata. Using his own annotated notes and extracted snippets from the official Slack documentation, Bob can easily create a multi-step tutorial that will be included in the Social CheatSheet library for Slack. Bob can also use the Share by Email feature (Fig. 2.3) to send this tutorial to team members who are currently not using Social CheatSheet (they will receive a web-link to the tutorial).

Asking questions or seeking clarifications: Based on the formative study's finding that more community-based features were needed, Social CheatSheet includes several ways for users to interact with each other to seek and provide further help. When viewing an existing note or tutorial, users can add comments (Fig. 2.4) to ask for clarifications or additional answers. They can also add a note as a question by marking it as needing an answer (Fig. 2.5) and optionally specifying an email address to subscribe for answers or follow-up comments. Doing so will display the note with an orange triangle (Fig 1.9) and a question mark, alerting other users that answers or clarifications are needed for this note.

Preserving privacy of curated content: Although the goal of Social CheatSheet is enable sharing of help content, our formative study highlighted the need to ensure user privacy is maintained throughout this process. Our system allows a user to mark a note as private (Fig. 2.5) and they can also use the censoring tool (Fig. 2.10) to hide private names or other confidential information.

4.2 Finding Community-Curated Content

Once users have the Social CheatSheet extension installed, the in-application collapsible *drawer* (Fig. 3) automatically retrieves the public notes and multi-step tutorials for that website. To ease the discovery of existing content and facilitate filtering, the drawer lists most popular tasks (Fig. 3.4) within the current application, sorted in descending order by the number of notes and tutorials tagged with the task.

For example, consider Alice who just joined Bob's team Alpha on Slack.com and needs to quickly figure out its key features so she can keep up with her team. When she visits her Slack team page, Social CheatSheet immediately shows the 5 most popular notes on Slack in the drawer along with the top tasks (*e.g.*, notifications, email, themes, etc.). She clicks on the notifications tag and sees the most popular notes and tutorials related to that topic. She would be interested later in learning about muting notifications, so she bookmarks the tutorial by clicking on the star symbol (Fig. 3.5).

When Alice clicks on "Show More", the drawer expands into the *grid view* and shows a full list of tasks and content for the current application (Fig. 1), appearing as an overlay across the page. When a user clicks on a task tag, the system filters down notes and tutorials to only show those tagged with the selected task.

Personalized sorting and filtering options: Following the formative study finding that more personalized retrieval of help content is needed, both the drawer and grid views offer features to sort and filter content based on personal preferences. By default, the most popular content is displayed first (based on views and votes). This can be changed to show the most recent notes, personal bookmarks, or only notes created by the user (Fig. 1.4), or to filter the content by type such as tutorials or unanswered questions (Fig. 1.5).

There are additional options to sort by expertise or by preference for visual or textual explanations (Fig. 1.6), both of which reorder the retrieved notes and tutorials such that items appropriate for the user's expertise level and preferred learning style are emphasized nearer to the top of the results. For example, since Alice is new to Slack, she clicks on the "First-time User" radio button to see only notes and tutorials appropriate for new users. Since Alice feels that she learns best with more visuals and fewer text descriptions, she moves the learning style control (Fig. 1.6) towards showing more visual curated content.

***Searching for content*:** Social CheatSheet places search bars in both the drawer (Fig. 3.2) and the full grid view (Fig. 1.3), allowing the list of notes to be filtered by a search query. The search algorithm returns a ranked list of notes and tutorials to be shown, by matching the search query against notes' description text, task tags, application name, or textual annotations drawn over the image.

***Viewing content as notes or tutorials*:** When a user clicks on a note from the grid or the drawer, an expanded version of the screenshot opens up in the editor interface (Fig. 2) along with annotations, descriptions, tags, other users' comments (Fig. 2.4), and other metadata (Fig. 2.2). If the origin of the note was a *YouTube* video, users can immediately play the video within the editor. The system also makes recommendations for other similar notes along the left side-bar (Fig. 2.8). Users can vote on the level of expertise that is commensurate with the note and also whether or not the note was helpful (Fig. 2.6). Clicking the "Share by Email" button (Fig. 2.3) allows users to quickly copy or email a link to a web page that shows the note or tutorial, to facilitate helping anyone who doesn't have Social CheatSheet installed. Users can also click on the minimize button (Fig. 2.9) to push the overlay almost fully down on the screen to easily go back-and-forth between their application task and instructions on the note.

When a user clicks on a tutorial (marked by a blue triangle, Fig. 1.8), all of the steps (individual notes) within the tutorial open up in the grid view (Fig. 1). Clicking on a particular step opens up the editor (as described above), but instead of showing the recommended similar notes, the left side bar shows a quick short-cut to other steps in the tutorial.

5 SYSTEM DESIGN AND IMPLEMENTATION

To enable Social CheatSheet to function across any web application, we built a Google Chrome extension that automatically inserts its interface and content into each web page the user visits. The notes and related metadata are all stored in a database on a third-party server, with the use of a browser extension allowing Social CheatSheet data to be transferred to and from this server without violating same-origin policies on websites.

5.1 Ranked Retrieval Approach for Showing Content

When a request for notes and tutorials is made to the Social CheatSheet server, it includes the type of retrieval (Most Popular, Most Recent, Favorites, or My Notes), the application name, and the current page title. Optionally, it may also include a task tag or search query, a learning-style preference, and an expertise level.

For a Most Popular retrieval request, the corpus of notes and tutorials for the given application is first optionally filtered to those having the specified task tag or matching a keyword in the search query, if applicable. Each note is then assigned a popularity score equal to ($upvotes - downvotes/2 + \text{sqrt}(views)/10$). Finally, any notes (or tutorials containing notes) that were created on a Web page with the same title as the user's current one are always ranked strictly higher than any without this property. The results are then returned in decreasing order of score.

For Most Recent retrieval requests, the same filtering by application, task, and search query takes place, but then the notes are simply returned in reverse chronological order. A Favorites request returns all of the user's favorited notes and tutorials, and a My Notes request returns all of the user's own created notes and tutorials, regardless of application, in the order they were favorited or created, respectively.

5.2 Approach for Recommending Similar Notes

The similarity of any two notes is calculated from the perceptual hashes (dHash [26]) of their images and pairwise TF-IDF scores of their text and task tags. A note's text, for this purpose, is the concatenation of its description and any textual annotations, separated by spaces. The image similarity was given less ranking weight because in some of our testing we found that it was less reliable for obtaining relevant results than user-curated text and tag similarity. Whenever a user requests to view a note, the 10 notes with the highest similarity are returned as recommended "similar notes".

5.3 Approach for Filtering by Learning Preferences and Expertise

A note's dominant learning style is labeled as either Textual or Visual, calculated by a decision tree when it is saved. Longer descriptions, longer text annotations, more highlighter annotations, and higher visible text fraction branch toward a Textual label, while greater numbers of other annotations (*e.g.*, arrows, rectangles) branch toward a Visual label. A note's expertise level is voted upon by the community according to the audience they deem most appropriate given the note's content and presentation (Fig. 2.6). If a user selects a learning style (Fig 1.6) or specifies an expertise level at retrieval time, each note's score in the sorting algorithm is scaled according to how closely its own calculated learning style and/or user-voted expertise level match the specified ones. Scores for tutorials are modified the same way, with their learning style and expertise level taken as the average of those values over all their individual steps.

6 FIELD DEPLOYMENT OF SOCIAL CHEATSHEET

How to best evaluate Social CheatSheet at this stage required consideration. We recognized that if we were to do an unconstrained “in the wild” field evaluation (such as [1,8]), it would take some time for independent help communities to organically emerge, and this process could be negatively impacted if there were significant usability issues (for which we had not yet tested). On the other hand, conducting a lab study would not be as compelling as we desired, given the artificial setting. We settled on a hybrid approach, similar in nature to [45] whereby we conducted a field deployment using a task-based approach where we encouraged users to explore the key functionality for curating and consuming content in the context of their own work. Our goal with the task-based field deployment was to investigate the following research questions:

1. How do users perceive the usability and usefulness of Social CheatSheet?
2. How well does Social CheatSheet support our design requirements identified in the formative study?
3. What factors would affect the long-term usage of Social CheatSheet, beyond the deployment period?

6.1 Methodology

We asked participants to use Social CheatSheet both as curators and consumers on the *Canvas* web application, our university’s learning management system. Canvas is a feature-rich application used by instructors and TAs to customize and deliver course content and by students to download lectures, submit assignments and view grades, among other more advanced tasks.

6.1.1 Participants

We recruited 15 study participants (7F, 8M) who were all between the ages of 19-34 and currently studying at a large university (6 Bachelors, 6 Masters, 3 PhD) in various majors. We posted flyers on university bulletin boards, and university mailing lists to recruit participants who were currently using Canvas either in a course as a student or as a teaching assistant (TA). Most of the participants were frequent users of Canvas: 7 used it daily, and 5 used it once a week and the rest used it less than once a month. None of the participants knew each other and we gave them anonymous user IDs during the deployment.

6.1.2 Week-long Deployment Protocol

Our deployment was carried out in three stages, yielding different data points throughout the study.

Pre-deployment setup: Study participants first attended (individually) a brief 20-30 minute setup session, during which they filled out a demographic questionnaire, and set up the extension on their laptops for the deployment period. We gave brief demonstrations to participants on how to access Social CheatSheet in the browser, an overview of the features available to them, a handout listing the example tasks and a brief summary of Social CheatSheet features just described to them. Prior to leaving, participants were remunerated with a $20 gift card and were asked to sign up for a post-deployment interview time slot.

Deployment period: We asked participants to use Social CheatSheet on their own time, as they saw fit, with their interactions in Social CheatSheet being logged to our server. We provided them with 6 example tasks that differed slightly depending on whether the participant was a TA or a student (due to different capabilities in Canvas based on user role). To test our curation features, we asked participants to assume that they were explaining the tasks to new students or new TAs. For example, we asked students to create content about forming project groups and submitting assignments, and TAs to create content about grading assignments and creating discussion threads. For assessing retrieval, we asked participants to seek help from Social CheatSheet to try to learn advanced features, such as using the real-time video streaming and the calendar scheduler.

As part of the bootstrapping process, three researchers pre-populated Social CheatSheet with a dozen notes and tutorials based on Canvas FAQs available on our university website (these did not overlap with the tasks provided to participants).

We sent daily emails to remind participants to try Social CheatSheet, and to fill out an open-ended online feedback form to identify any usability issues they encountered and/or any thoughts on using the tool. The activity logs saved detailed time-stamped information about user activity within Social CheatSheet (based on clicks on any interactive feature and opening or closing of the overlay). No other personally identifiable information or page URLs were logged.

Usage Sessions	
Avg # of usage sessions per user	9.1
Avg length of session (min) per user	9.5
Avg daily time spent (min) per user	12.7
Avg # of online surveys per user	2.9
New Content Added during Deployment	
Standalone notes created	20
Tutorials created	80
Avg # of notes and/or tutorials per user	6.7
Features Used during Deployment	
Questions asked	7
Notes marked private	6
Shares by email	4
Expertise level changes	14
Learning style slider changes	35
Upvotes and downvotes	62
Search queries issued	222
Task tags clicked	30
Censor tool uses	239
Note views	136
Tutorial views	156

Table 1: Summary of Social CheatSheet usage during the deployment

Post-deployment interviews: Study participants returned for a follow-up interview and filled out an exit questionnaire about their experience with our system. Our semi-structured interviews with participants lasted about 30 minutes and probed into user perceptions of Social CheatSheet based on their usage over the past week (often by showing the researchers some of the notes or tutorials they had created), how they felt about themselves or others using it in the future, and how it differed from ways they might normally give or receive help. Finally, the researchers helped participants uninstall the extension, and provided them with an additional $30 gift card for completing the deployment and the interview.

6.1.3 Data Analysis

As described above, we used multiple sources to collect data about participants' use and perceptions of Social CheatSheet via pre/post-deployment questionnaires, usage logs, online forms, and interviews. We used an inductive analysis approach [43] to look for recurrent themes and triangulated results from these data sources.

6.2 Results

The overall usage of Social CheatSheet and use of key features is summarized in Table 1. Below, we present insights from our main findings in relation to our research questions.

6.2.1 Usability and Usefulness of Social CheatSheet

The post-deployment questionnaires indicated that the majority of participants (11/15) found Social CheatSheet easy to use (Fig. 5). Participants highlighted several reasons for this, such as the visual nature of the curated content: *"I guess the main advantage of this is that I can have something visual that replicates the actual screen, or the interface that they're going to be using… and also the ability to really highlight stuff or, you know, write comments on the screen itself, so that's something beneficial." (P02)*

All 15 participants mentioned that they found it useful to see other people's curated notes and tutorials. Interestingly, some participants who considered themselves to be frequent expert users of Canvas revealed that they still learned something new by browsing the content curated by other users: *"I saw people use Canvas in ways I'd never used it before, and that was interesting...and to see how people responded to the same task, like they did it in different ways, it was good... I was also able to learn a lot from just checking through the feedback from other people." (P04)*

Users who would otherwise turn to external tools and applications for saving and sending help to others found Social CheatSheet easy to use for streamlining both the curation process and the means of sharing: *"I usually use the photo editor...the one for painting on the computer... I use screenshots with some social media apps... But with this [Social CheatSheet], I don't have to switch between other applications: I just use Social CheatSheet and it takes the screenshot automatically... it's easy." (P13)*

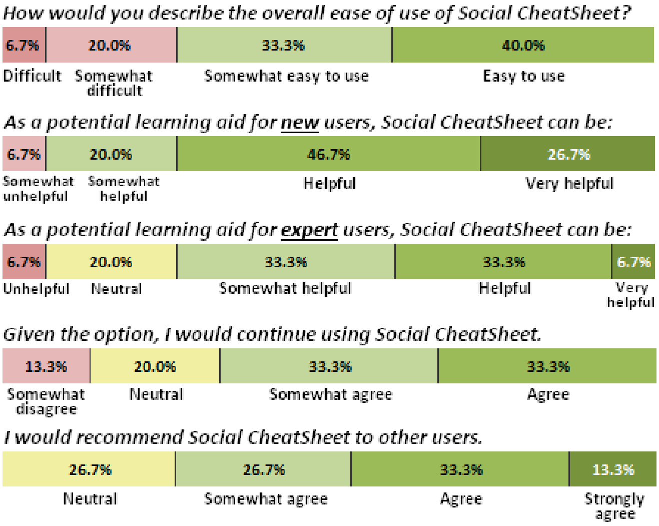


Fig. 5. Post-Deployment questionnaire results, all on a 7-point Likert scale. Scale positions with zero responses are not shown.

In addition to being easy-to-use, participants overwhelmingly indicated that Social CheatSheet was useful in finding relevant curated content compared to other forms of static help, such as documentation created by application owners: *"Maybe sometimes they [in-app help creators] didn't think through a lot of usage scenarios, but in Social CheatSheet since many persons [can contribute], it's possible to find the information you hope to find." (P01)*

As shown in Fig. 5, almost every participant (14/15) felt that new users to an application would find Social CheatSheet helpful. The common reason was that new users often face a "blank-slate problem" and do not even know where to look for help or tips. The visual annotations and step-by-step nature of the content could be particularly beneficial for these users: *"I would say Social CheatSheet is more beneficial to new users and novice users more than the expert users... It does take a lot of screenshots, and it's very detailed in that way. So that kind of level of detail would benefit novice users more than an expert user who may prefer less numbers of slides [notes] with more content condensed into each particular slide". (P02)*

Although less pervasive than for novices, a strong majority (11/15) still indicated that Social CheatSheet would be helpful for experts (Fig. 5). These participants felt that experts could use Social CheatSheet to share their knowledge with others, even if they would not necessarily find help for themselves: *"If I am an expert... I'm not going to use it for solving my questions. I may be willing to solve others' questions if I have time, if it's easy for me to use the tutorial thing." (P11)*

As expected, given that this was the first evaluation of Social CheatSheet, participants did identify some usability challenges and bugs. We collected 44 such issues through the online survey during the deployment. For example, participants wanted to have finer-grain control or real-time feedback on the steps being recorded in tutorials. They also provided suggestions for including additional features, such as voice-over, options for arranging and mass-editing tutorial steps, and more customization of tutorial annotation tools.

6.2.2 Social CheatSheet's Support for Design Requirements

We next present key findings from our deployment in relation to our five design requirements.

Community-oriented features: Overall, we found that users benefited from having community-oriented features to easily browse and locate useful content. For example, even expert users of Canvas indicated that often they forget certain features over time, and do not even know about advanced/hidden "power-user" features. Social CheatSheet was well-suited for discovering such features with community-generated tips and reminders: *"I was a TA [before], and I forgot some of the basic issues in Canvas…so, for the people who are experts for a period of time and then they don't use [an app] for a long time, [Social CheatSheet] can be a great tool to refresh their mind." (P01)*

It was interesting to see that many participants tried different community-oriented features on their own (even though they were not explicitly mentioned in the provided tasks). For example, two participants tried using the commenting features: one tried answering another participant's question on a note that was marked as needing an answer; another participant added a clarification on her own tutorial step. The email feature was used four times to send notes to other people who were not Social CheatSheet users and participants found it intuitive and useful for sharing content with their parents or friends. Several more indicated that the email feature can be useful in the future, such as to help older adults who may not want to install special extensions: *"Once [we] make tutorials with many steps, then, for example elderly people and some people who are not familiar…can just follow one step by step, and it's pretty clear to explain what should I do on this website." (P10)*

Additionally, over half of the participants indicated that the ability to ask questions to the community was useful and easy to use, with 6 unique participants actually creating at least one note marked as needing an answer (these were answered by the researchers during the deployment).

Since the participants in the study were anonymous, and none indicated that they knew any of the other participants, it is encouraging to see some evidence of direct social interactions during the deployment.

***Step-by-step tutorials*:** As shown in Table 1, tutorials were more popular than standalone notes in terms of usage and perceived utility: all but one participant marking the tutorial feature to be "useful" or "very useful". This was substantially higher than standalone notes, for which only 11/15 participants said the same, and which one participant did not use at all. Consistent with our formative study, we discovered that participants liked breaking apart the task into different steps rather than showing everything on a single image: *"I really like the tutorial function...because you can see all the [notes] together and see the context of using some function" (P03)*

***Personalization options*:** We observed that just over half of the participants (8/15) tried changing the value of the learning-style slider at some point during the deployment, and among these participants, all but one used it multiple times. Similarity, 7/15 participants tried filtering notes by expertise level, with all but one of these setting it to a value of "Novice", and the remaining one to "Expert". We were pleased to see that users were interested in trying these features without any prompting. While these personalization features were currently manually controlled by users, some users were curious to see how the system could learn their preferences over time and show relevant content accordingly.

***Preserving privacy*:** Although not explicitly stated in any task, four participants marked notes as private. This may not be indicative of realistic usage, since a primary motivation for that feature is to keep a reference for oneself in the future, and this use case is particularly ill-suited to a short-term deployment. While screenshots of Canvas were likely to contain private information such as student names or grades, most users (11/15) preferred using our censor tool to "black out" these sections of the screenshots instead (rather than making the note private). Several participants indicated that they would prefer to see the censor streamlined such that they could hide same parts of multiple similar tutorial steps, or specify areas to censor upfront when creating a tutorial.

***Flexible natural-language search options*:** While it is not a research contribution, our improvements to Social CheatSheet's search engine did improve user experience. Participants in the field deployment appeared to use search queries as their primary means of navigating the help resources within Social CheatSheet, issuing 222 freeform search queries in total (7.4x more than their use of task tags). (This differs substantially from the usage patterns during the formative lab study, where participants used 1.3x more task tags than search queries on average due to limitations of the original search algorithm).

6.2.3 Potential for Long-Term Usage and Adoption of Social CheatSheet

One of our most encouraging findings was that two-thirds (10/15) of our participants spontaneously used Social CheatSheet beyond the Canvas application. Our logs showed that they added notes or tutorials on 12 other sites, such as our university registration site, a teaching evaluation tool, Amazon, and Google Sites, among others. This non-Canvas activity comprised 25% of the all notes created and 19% of all tutorials, which was surprising given that none of our example tasks directly asked participants to use these sites. We were also encouraged to see the majority of participants (10/15) agreeing that they would continue to use Social CheatSheet after the deployment, if they had the option to do so (Fig. 5). Furthermore, 11/15 indicated that they would recommend Social CheatSheet to others (Fig. 5).

In our interviews, participants enumerated several reasons why they would keep using Social CheatSheet—application independence and ease of use were cited as key reasons: *"I think the biggest advantage is it's a plugin in the website, to any website. Whenever you want to see how to use the website you can just click on CheatSheet and then do some records" (P03).* Another participant explained that Social CheatSheet could help her avoid the social cost of calling someone and re-learning a complex process (*e.g.*, for installing software): *"I think it [Social CheatSheet] would be useful for downloading and setting up software that requires following some instructions, because sometimes when you've done it before and you want to do it a second time on a different system, you can't remember and have to go through the whole process again" (P04).* This use case also reflects an infrequent activity that the user knows she will need help with in the future, and can be viewed as a collaboration with her future self.

Another participant talked about using Social CheatSheet for coding-related use cases (as a curator and consumer): "…if I were to try to explain a code snippet to someone...then I can just create a tutorial using this [Social CheatSheet] and send an email to them. So, I guess it just has to do with how it's relevant, the tutorial I'm making is, to my own needs." (P02)

Some participants mentioned that they would only create content if they had a specific need to do so, such as a family member asking for help, but not simply to share their software knowledge unasked. Overall, most participants indicated that they would be more likely to use Social CheatSheet as a content consumer than as a content creator, in part because of the additional work required to curate high-quality content. One participant who was not a frequent user of Canvas said that he did not see himself contributing content to Social CheatSheet about Canvas, but that for other applications that he was more familiar with and needed to use advanced features, he would add notes: *"The thing is...I don't use [Canvas] that frequently, so I don't make notes…but if it's some application [e.g., Google Search] that I'm using every day, I might add notes." (P15)*

While participants were overall positive about accessing curated notes and tutorials, some felt that they did not find anything helpful due to the limited content currently available, but that the system could be more useful in the future: *"I found myself using it to try to find answers to the things I didn't know how to do in Canvas, but unfortunately there were no answers there yet. But I thought it was really handy how you can search things quickly and… if I [were] to do it next week then it would be probably more useful." (P08)*

Another person pointed out that even though he found relevant content and found the visuals to be helpful, the metadata and description quality could be improved: *"I searched through the tutorials to see if I could find if anyone else had submitted a tutorial for [a] scenario. I found some…but they were very sloppy, and I couldn't get any information from them." (P07)*

Other participants were less concerned about the content quality in the long-term because they felt that the community could help maintain content with up-votes and down-votes and it was just a matter of adding moderation features and quality checks that are already common on forums and social applications.

7 DISCUSSION

Our design and evaluation of Social CheatSheet demonstrates how users could participate and benefit from community-curated help, particularly multi-step tutorials, atop feature-rich web applications. More broadly, our work offers insights into a how to design a useful collaboratively constructed information space [2] that not only leverages the benefits of the community, but also offers users a private space for their own content and ability to personalize retrieval. Although our evaluation was a short-term small-scale deployment, it was encouraging to see that users could still find useful curated content, benefit from trying out the system on other applications, and be willing to keep using the system beyond the deployment. Nonetheless, our findings also highlight some limitations that future work can address to improve different aspects of social curation in the context of learning software, as we outline below.

7.1 Incentivizing Social Curation and Community-Building

Our findings showed that a third of the users would likely not add their own curated notes and tutorials. This is not surprising given other findings on online communities where most users are consumers rather than contributors [3]. However, given the ease of using the curation features, over half of the users were neutral and not opposed to adding content if the community were to grow in the future. Also, since many users already curate content for helping family and friends or even take notes for themselves to save helpful instructions [46], it may just be a matter of further incentivizing users to participate and adjust to the Social CheatSheet medium for curation. Furthermore, given the long-term growth of technical help communities such as *Stack Overflow* [30] and software discussion forums [28], we have some confidence that Social CheatSheet can eventually grow as a community portal.

In future work, our platform can be used to investigate different aspects of user participation in this new medium via controlled field experiments. For example, to what extent does the ease of visual curation entice more people to participate than a traditional text-based curation interface? How can we employ techniques such as gamification [10,11] that have worked well in other domains to improve in-application social curation? How would participation rates compare across different domain-specific communities (*e.g.*, users of spreadsheets vs. 3D modelling vs. health applications)?

As user contributions grow and more content gets added, naturally a concern may be if the user experience would be compromised in some way. Future work could investigate how well the retrieval algorithms and the usability of the system scale with active user communities and growth in curated content.

7.2 Maintaining Quality and Availability of Curated Help

Despite the overall positive feedback about retrieving help, users did raise some concerns about the long-term availability of relevant high-quality instructions and related metadata. As we explore mechanisms for incentivizing users to participate in the social curation process, we believe it is important to also design mechanisms to ensure proper maintenance of community-curated content. In particular, given the rapid evolution of web applications, it is important to consider how the content and metadata could be maintained to preserve relevance. In our current model, the out-of-date information will get down-voted by the community, but future work could explore how to automatically detect web page changes and compute a “freshness” score of the related help content.

In our field study, users retrieved content added by other anonymous users and they did not express any major concern about not knowing a contributor’s identity or reputation. However, as has been shown in previous work, content quality and user perceptions in the long run can be affected by contributions made by experts vs. non-experts [34]. Future work could explicitly investigate how users perceive curated content added by anonymous users (as they did in our study) vs. users revealing their identities and additional information about their expertise.

Even though Social CheatSheet is a community-based help system that does not depend on application owners, there is opportunity for technical app support staff to help their users by maintaining the availability and quality of the help content. Support staff already contribute content on *Twitter*, *Facebook*, and product forums [48], often by providing “verified” answers—similarly, they could use Social CheatSheet to broadcast important instructions, changes, or tips to users that would appear within a user interface.

7.3 Personalizing and Recommending Targeted Curated Help

Although users found it easy to locate and filter relevant curated help, there may be an opportunity to further push the boundaries of our social curation platform and offer more targeted curated help to users. For example, we can try to infer what tasks users are actually performing in the application and proactively recommend relevant curated help (this is in contrast to other innovations that focus on recommending commands and features [32,35]). Furthermore, techniques such as pixel-based reverse engineering of interface structure [12] could be used not only to glean what tasks are being performed, but also to improve the accuracy of our image-based note similarity metrics. Future work can also investigate how to build hybrid help systems that have knowledge about the characteristics of users participating in the curation platform and can use collaborative filtering approaches [38,40] to suggest more targeted content.

Feedback from both our formative and field studies indicated that some users learn better from visual vs. text-based instructions and novices and experts prefer different granularity and format of help. Yet, most current software learning materials rarely take individual user characteristics into account and offer the same help to all users. While our system begins to explore the idea of personalizing curated help content by allowing users to retrieve different content based on their expertise (*e.g.*, novice vs. expert) and learning style [17] preferences (*e.g.*, visual vs. textual), there are many opportunities for more automatic and adaptive personalizations. For example, it may be possible to extend and integrate HCI research on automatically detecting novice vs. expert behaviors [18] and supporting novice to expert transitions [9]. Similarly, mechanisms for automatically inferring learning styles have already been explored in educational settings [17,25] and it would be interesting to investigate how they could be adapted in the context of using software and accessing help.

Lastly, there is potential to explore the benefits of social in-application help in contexts beyond troubleshooting or seeking help with feature-rich applications. For example, there is already evidence of online sharing ecosystems around customizations [19], but there is little in-application support for this type of sharing. Perhaps platforms such as Social CheatSheet can be extended to allow users to share such content. Personalization of socially curated content can also be investigated in educational contexts, especially the potential of adding curated content to large-scale distance learning platforms, such as MOOCs.

8 CONCLUSIONS

In conclusion, we have contributed a new vision for enabling social curation of help content such that any user can add and retrieve task-specific curated visual instructions and step-by-step tutorials atop any existing web application. Our initial field study further contributes insights into the benefits of using Social CheatSheet as a curator and a consumer. We plan to make our social platform open so that together with other CSCW and HCI researchers we can investigate incentives for social sharing and community-building, innovate on in-application social help recommendation algorithms, and design new interaction paradigms for personalizing socially curated help content.

ACKNOWLEDGMENTS

This research was supported in part by the National Science and Engineering Research Council of Canada (NSERC). The authors would like to thank Adena Lin, Carman Neustaedter, Michael Terry, and Jacob Wang for their assistance.

REFERENCES

[1] Mark S. Ackerman. 1998. Augmenting Organizational Memory: A Field Study of Answer Garden. *ACM Trans. Inf. Syst.* 16, 3: 203–224. https://doi.org/10.1145/290159.290160

[2] Mark S. Ackerman, Juri Dachtera, Volkmar Pipek, and Volker Wulf. 2013. Sharing Knowledge and Expertise: The CSCW View of Knowledge Management. *Comput. Supported Coop. Work* 22, 4–6: 531–573. https://doi.org/10.1007/s10606-013-9192-8

[3] Charles Arthur. 2006. What is the 1% rule? *The Guardian.* Retrieved from https://www.theguardian.com/technology/2006/jul/20/guardianweeklytechnologysection2

[4] Andrea Bunt, Patrick Dubois, Ben Lafreniere, Michael A. Terry, and David T. Cormack. 2014. TaggedComments: Promoting and Integrating User Comments in Online Application Tutorials. In *Proceedings of the 32Nd Annual ACM Conference on Human Factors in Computing Systems* (CHI '14), 4037–4046. https://doi.org/10.1145/2556288.2557118

[5] John M. Carroll, Penny L. Smith-Kerker, James R. Ford, and Sandra A. Mazur-Rimetz. 1987. The Minimal Manual. *Hum.-Comput. Interact.* 3, 2: 123–153. https://doi.org/10.1207/s15327051hci0302_2

[6] Parmit K. Chilana, Andrew J. Ko, and Jacob Wobbrock. 2015. From User-Centered to Adoption-Centered Design: A Case Study of an HCI Research Innovation Becoming a Product. In *Proceedings of the 33rd Annual ACM Conference on Human Factors in Computing Systems* (CHI '15), 1749–1758. https://doi.org/10.1145/2702123.2702412

[7] Parmit K. Chilana, Andrew J. Ko, and Jacob O. Wobbrock. 2012. LemonAid: Selection-based Crowdsourced Contextual Help for Web Applications. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems* (CHI '12), 1549–1558. https://doi.org/10.1145/2207676.2208620

[8] Parmit K. Chilana, Andrew J. Ko, Jacob O. Wobbrock, and Tovi Grossman. 2013. A Multi-site Field Study of Crowdsourced Contextual Help: Usage and Perspectives of End Users and Software Teams. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems* (CHI '13), 217–226. https://doi.org/10.1145/2470654.2470685

[9] Andy Cockburn, Carl Gutwin, Joey Scarr, and Sylvain Malacria. 2014. Supporting Novice to Expert Transitions in User Interfaces. *ACM Comput. Surv.* 47, 2: 31:1–31:36. https://doi.org/10.1145/2659796

[10] Sebastian Deterding, Dan Dixon, Rilla Khaled, and Lennart Nacke. 2011. From Game Design Elements to Gamefulness: Defining "Gamification." In *Proceedings of the 15th International Academic MindTrek Conference: Envisioning Future Media Environments* (MindTrek '11), 9–15. https://doi.org/10.1145/2181037.2181040

[11] Sebastian Deterding, Miguel Sicart, Lennart Nacke, Kenton O'Hara, and Dan Dixon. 2011. Gamification. Using Game-design Elements in Non-gaming Contexts. In *CHI '11 Extended Abstracts on Human Factors in Computing Systems* (CHI EA '11), 2425–2428. https://doi.org/10.1145/1979742.1979575

[12] Morgan Dixon and James Fogarty. 2010. Prefab: Implementing Advanced Behaviors Using Pixel-based Reverse Engineering of Interface Structure. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems* (CHI '10), 1525–1534. https://doi.org/10.1145/1753326.1753554

[13] Mira Dontcheva, Steven M. Drucker, Geraldine Wade, David Salesin, and Michael F. Cohen. 2006. Summarizing Personal Web Browsing Sessions. In *Proceedings of the 19th Annual ACM Symposium on User Interface Software and Technology* (UIST '06), 115–124. https://doi.org/10.1145/1166253.1166273

[14] Evernote Corporation. Homepage. *Evernote*. Retrieved July 22, 2017 from https://evernote.com/

[15] Adam Fourney, Ben Lafreniere, Parmit Chilana, and Michael Terry. 2014. InterTwine: Creating Interapplication Information Scent to Support Coordinated Use of Software. In *Proceedings of the 27th Annual ACM Symposium on User Interface Software and Technology* (UIST '14), 429–438. https://doi.org/10.1145/2642918.2647420

[16] C. Ailie Fraser, Mira Dontcheva, Holger Winnemoeller, and Scott Klemmer. 2016. DiscoverySpace: Crowdsourced Suggestions Onboard Novices in Complex Software. In *Proceedings of the 19th ACM Conference on Computer Supported Cooperative Work and Social Computing Companion* (CSCW '16 Companion), 29–32. https://doi.org/10.1145/2818052.2874317

[17] Sabine Graf and Prof. Kinshuk. 2006. An Approach for Detecting Learning Styles in Learning Management Systems. In *Proceedings of the Sixth IEEE International Conference on Advanced Learning Technologies* (ICALT '06), 161–163. Retrieved from http://dl.acm.org/citation.cfm?id=1156068.1156382

[18] Tovi Grossman and George Fitzmaurice. 2015. An Investigation of Metrics for the In Situ Detection of Software Expertise. *Hum.-Comput. Interact.* 30, 1: 64–102. https://doi.org/10.1080/07370024.2014.881668

[19] Mona Haraty, Joanna McGrenere, and Andrea Bunt. 2017. Online Customization Sharing Ecosystems: Components, Roles, and Motivations. In *Proceedings of the 2017 ACM Conference on Computer Supported Cooperative Work and Social Computing* (CSCW '17), 2359–2371. https://doi.org/10.1145/2998181.2998289

[20] Björn Hartmann, Daniel MacDougall, Joel Brandt, and Scott R. Klemmer. 2010. What Would Other Programmers Do: Suggesting Solutions to Error Messages. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems* (CHI '10), 1019–1028. https://doi.org/10.1145/1753326.1753478

[21] Lichan Hong and Ed H. Chi. 2009. Annotate Once, Appear Anywhere: Collective Foraging for Snippets of Interest Using Paragraph Fingerprinting. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems* (CHI '09), 1791–1794. https://doi.org/10.1145/1518701.1518976

[22] William Jones. 2008. Keeping Found Things Found: The Study and Practice of Personal Information Management: The Study and Practice of Personal Information Management. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA.

[23] Juho Kim, Phu Tran Nguyen, Sarah Weir, Philip J. Guo, Robert C. Miller, and Krzysztof Z. Gajos. 2014. Crowdsourcing Step-by-step Information Extraction to Enhance Existing How-to Videos. In *Proceedings of the 32Nd Annual ACM Conference on Human Factors in Computing Systems* (CHI '14), 4017–4026. https://doi.org/10.1145/2556288.2556986

[24] Aniket Kittur, Andrew M. Peters, Abdigani Diriye, and Michael Bove. 2014. Standing on the Schemas of Giants: Socially Augmented Information Foraging. In *Proceedings of the 17th ACM Conference on Computer Supported Cooperative Work & Social Computing* (CSCW '14), 999–1010. https://doi.org/10.1145/2531602.2531644

[25] Aleksandra Klašnja-Milićević, Boban Vesin, Mirjana Ivanović, and Zoran Budimac. 2011. E-Learning personalization based on hybrid recommendation strategy and learning style identification. *Computers & Education* 56, 3: 885–899.

[26] Neal Krawetz. Kind of Like That. *The Hacker Factor Blog*. Retrieved June 15, 2017 from http://www.hackerfactor.com/blog/?/archives/529-Kind-of-Like-That.html

[27] Benjamin Lafreniere, Tovi Grossman, and George Fitzmaurice. 2013. Community Enhanced Tutorials: Improving Tutorials with Multiple Demonstrations. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems* (CHI '13), 1779–1788. https://doi.org/10.1145/2470654.2466235

[28] Karim R Lakhani and Eric von Hippel. 2003. How open source software works: "free" user-to-user assistance. *Research Policy* 32, 6: 923–943. https://doi.org/10.1016/S0048-7333(02)00095-1

[29] Wei Li, Tovi Grossman, and George Fitzmaurice. 2014. CADament: A Gamified Multiplayer Software Tutorial System. In *Proceedings of the 32Nd Annual ACM Conference on Human Factors in Computing Systems* (CHI '14), 3369–3378. https://doi.org/10.1145/2556288.2556954

[30] Lena Mamykina, Bella Manoim, Manas Mittal, George Hripcsak, and Björn Hartmann. 2011. Design Lessons from the Fastest Q&A Site in the West. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems* (CHI '11), 2857–2866. https://doi.org/10.1145/1978942.1979366

[31] Justin Matejka, Tovi Grossman, and George Fitzmaurice. 2011. IP-QAT: In-product Questions, Answers, & Tips. In *Proceedings of the 24th Annual ACM Symposium on User Interface Software and Technology* (UIST '11), 175–184. https://doi.org/10.1145/2047196.2047218

[32] Justin Matejka, Wei Li, Tovi Grossman, and George Fitzmaurice. 2009. CommunityCommands: Command Recommendations for Software Applications. In *Proceedings of the 22Nd Annual ACM Symposium on User Interface Software and Technology* (UIST '09), 193–202. https://doi.org/10.1145/1622176.1622214

[33] Meredith Ringel Morris, Jaime Teevan, and Katrina Panovich. 2010. What Do People Ask Their Social Networks, and Why?: A Survey Study of Status Message Q&a Behavior. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems* (CHI '10), 1739–1748. https://doi.org/10.1145/1753326.1753587

[34] Dana Movshovitz-Attias, Yair Movshovitz-Attias, Peter Steenkiste, and Christos Faloutsos. 2013. Analysis of the Reputation System and User Contributions on a Question Answering Website: StackOverflow. In *Proceedings of the 2013 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining* (ASONAM '13), 886–893. https://doi.org/10.1145/2492517.2500242

[35] Emerson Murphy-Hill, Rahul Jiresal, and Gail C. Murphy. 2012. Improving Software Developers' Fluency by Recommending Development Environment Commands. In *Proceedings of the ACM SIGSOFT 20th International Symposium on the Foundations of Software Engineering* (FSE '12), 42:1–42:11. https://doi.org/10.1145/2393596.2393645

[36] Michael Nebeling, Maximilian Speicher, and Moira C. Norrie. 2013. CrowdAdapt: Enabling Crowdsourced Web Page Adaptation for Individual Viewing Conditions and Preferences. In *Proceedings of the 5th ACM SIGCHI Symposium on Engineering Interactive Computing Systems* (EICS '13), 23–32. https://doi.org/10.1145/2494603.2480304

[37] Solomon Negash, Terry Ryan, and Magid Igbaria. 2003. Quality and Effectiveness in Web-based Customer Support Systems. *Inf. Manage.* 40, 8: 757–768. https://doi.org/10.1016/S0378-7206(02)00101-5

[38] Paul Resnick, Neophytos Iacovou, Mitesh Suchak, Peter Bergstrom, and John Riedl. 1994. GroupLens: An Open Architecture for Collaborative Filtering of Netnews. In *Proceedings of the 1994 ACM Conference on Computer Supported Cooperative Work* (CSCW '94), 175–186. https://doi.org/10.1145/192844.192905

[39] Abigail Sellen and Anne Nicol. 1995. Building user-centered on-line help. In *Human-computer interaction*, Ronald M. Baecker, Jonathan Grudin, William A. S. Buxton and Saul Greenberg (eds.). Morgan Kaufmann Publishers Inc., 718–723.

[40] Upendra Shardanand and Pattie Maes. 1995. Social Information Filtering: Algorithms for Automating "Word of Mouth." In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems* (CHI '95), 210–217. https://doi.org/10.1145/223904.223931

[41] Vandana Singh, Michael B. Twidale, and David M. Nichols. 2009. Users of Open Source Software - How Do They Get Help? In *2009 42nd Hawaii International Conference on System Sciences*, 1–10. https://doi.org/10.1109/HICSS.2009.489

[42] Vandana Singh, Michael B. Twidale, and Dinesh Rathi. 2006. Open Source Technical Support: A Look at Peer Help-Giving. In *Proceedings of the 39th Annual Hawaii International Conference on System Sciences (HICSS'06)*, 118c–118c. https://doi.org/10.1109/HICSS.2006.370

[43] Anselm Strauss and Juliet Corbin. 1990. *Basics of qualitative research*. Newbury Park, CA: Sage.

[44] Hironobu Takagi, Shinya Kawanaka, Masatomo Kobayashi, Takashi Itoh, and Chieko Asakawa. 2008. Social accessibility: achieving accessibility through collaborative metadata authoring. In *Proceedings of the 10th international ACM SIGACCESS conference on Computers and accessibility*, 193–200.

[45] Max G. Van Kleek, Michael Bernstein, Katrina Panovich, Gregory G. Vargas, David R. Karger, and MC Schraefel. 2009. Note to Self: Examining Personal Information Keeping in a Lightweight Note-taking Tool. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems* (CHI '09), 1477–1480. https://doi.org/10.1145/1518701.1518924

[46] Laton Vermette, Parmit Chilana, Michael Terry, Adam Fourney, Ben Lafreniere, and Travis Kerr. 2015. CheatSheet: A Contextual Interactive Memory Aid for Web Applications. In *Proceedings of the 41st Graphics Interface Conference* (GI '15), 241–248. Retrieved from http://dl.acm.org/citation.cfm?id=2788890.2788933

[47] Dan Whaley. Home. *Hypothesis*. Retrieved July 25, 2017 from https://web.hypothes.is/

[48] Alan M. Wilson, Valarie A. Zeithaml, and Mary Jo Bitner. 2012. *Services Marketing: Integrating Customer Focus Across the Firm.* McGraw-Hill Higher Education.

[49] Heather Wiltse and Jeffrey Nichols. 2009. PlayByPlay: Collaborative Web Browsing for Desktop and Mobile Devices. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems* (CHI '09), 1781–1790. https://doi.org/10.1145/1518701.1518975

Received June 2017; revised July 2017; accepted November 2017.