Bazaar: A Situated Crowdsourcing Market

Abstract
In this paper we present a situated crowdsourcing market named Bazaar. Our platform enables researchers to deploy crowdsourcing tasks on simple crowdsourcing kiosks, which can then be placed in different locations. We describe the different components that compose Bazaar, discuss a successful deployment using the platform and reflect on possible future work using Bazaar.

Author Keywords
Crowdsourcing; situated technologies; market; tasks; kiosks; tablets.

ACM Classification Keywords
H.5.m. Information interfaces and presentation (e.g., HCI): Miscellaneous.

Introduction
In recent years crowdsourcing has been growing in popularity both in academia and industry. The emergence of online crowdsourcing platforms (such as Amazon’s Mechanical Turk, CrowdFlower and oDesk) makes it convenient to pay for workers willing to solve a range of different tasks. Many of these tasks are difficult to solve by computers, like for example, audio transcribing and recognizing objects in images [15]. The most common motivator for these workers to
conduct such crowdsourcing tasks is payment. However, there are examples in which workers contributed altruistically [4,6] or contributed without even being aware (e.g. reCAPTCHA [17]).

While the majority of research on crowdsourcing has been conducted on online and mobile platforms, like the ones mentioned above, there has been a growing interest in using situated technologies for crowdsourcing. Situated crowdsourcing consists of embedding input mechanisms (e.g., public displays, tablets) into a physical space, to leverage users' serendipitous availability [14] or idle time (“cognitive surplus” [16]). It allows for a geofenced and more contextually controlled crowdsourcing environment, thus enabling targeting certain individuals [4], leveraging people's local knowledge [5] or simply reaching an untapped source of potential workers [9].

In this paper we present a platform, named Bazaar, designed to facilitate such situated crowdsourcing deployments. Next, we describe some of the work conducted within the situated crowdsourcing domain. We then describe our platform in more detail.

Related Work
There has been little research conducted using situated technologies for crowdsourcing. In one of the few examples, Heimerl et al. reported Umati [8], which used a vending machine with a touch display for locally relevant tasks. Workers could earn credits performing these tasks and exchange them for physical rewards directly on the vending machine (snacks). In another paper, Huang explored if micro-volunteering can be leveraged by a situated crowdsourcing platform to enable problem-solving efforts with high-quality results [12]. In a position paper, Marshall et al. [13] outline a project, where its goals were to reduce staff stress levels and increase restorative opportunities through situated crowdsourcing.

Recently, researchers have studied the use of public displays as a viable crowdsourcing medium. For instance, crowdsourcing on these devices has been used to gauge fluctuations on a community's diurnal collective emotion [7], generate a repository of keywords that are relevant to a location [5] or polling public opinion [10]. Bazaar is designed for tablets mounted on stands, that together form a type of crowdsourcing kiosk, and enables the deployment of a number of simultaneous tasks, contrary to the previously mentioned single-purpose situated crowdsourcing applications.

Bazaar
Bazaar consists of three main components: a grid of client devices (kiosks), a server and a researcher/administrator hub. We explain each of these components in more detail next.

Client Device (kiosks)
Each kiosk contains an Android tablet with a 10.1” touch-screen, a charger, and uses WiFi to connect to the server (Figure 1). Our client software for the tablets is set to “kiosk mode” to ensure that it was always visible on screen. It also recovers from crashes, and disables unwanted OS functionality (notification bars, etc.). The physical buttons of the tablet are hidden by the kiosk's enclosure. The tablets host the crowdsourcing application that allows workers to access the tasks. Bazaar continuously accesses the server side database, so a reliable Internet connection is required.
The welcome screen of the kiosks contains a brief description of the system, and prompts users to create an account or login (Figure 2). Registration requires only a username and password because a lengthy process would reduce participation when using situated technologies [1].

Figure 1: One of the kiosks that contain an Android tablet with a 10.1” touch-screen.

Figure 2: Welcome screen.

Subsequently, workers are shown a main screen (Figure 3). At all times the upper bar of the application displays the HexaCoin balance of the worker, the virtual currency awarded for completing tasks. In the main screen, workers can see how many of their completed tasks have been approved, rejected or still pending moderation. In addition, workers can also see a list of the most recent news set by the system administrators.

By swiping right workers can access the list of tasks and the leaderboard with the nickname of workers with the most earnings to date (Figure 4). The tasks include a brief description, a thumbnail and the reward assigned to it. Upon selecting a task, an embedded Internet browser (webview) is opened within the Android application, where workers can then complete the tasks to earn HexaCoins, the virtual currency of Bazaar.

Figure 3: Main screen.

Figure 4: Task list and leaderboard.

By swiping left workers can access the shop (Figure 5). Here, workers can get money or goods in exchange for their HexaCoins. Workers can then pickup their rewards from a physical shop we setup in our laboratory premises. When visiting our lab, workers have to verify their identity by logging into Bazaar on a dedicated computer.

Figure 5: Shop screen.
Server
All communication between the Android application and the MySQL database happens via a REST-like API. The API was based on the Slim micro framework [3]. Slim is a lightweight PHP framework for writing web applications and APIs. From Bazaar’s point-of-view, it provides two key features: an URL router and HTTP request/response abstractions. Slim router allowed us to map resource URLs to call back functions for specific HTTP request methods while request/response abstractions allowed easy access to request/response bodies and headers. Slim also provides a route middleware that makes authentication of API requests easy and straightforward.

All the interaction with the database is made via a separate dbHandler class. This class provides all the necessary functions to connect and perform operations on the database. Other middleware functionalities include request authentication, defining HTTP methods and endpoints with associated call back functions.

All the resources accept only GET or POST (or both) HTTP methods. All data exchanges are made via the API using JSON-arrays. When calling a POST method, the client appends a Capsule in the request body containing all the required parameters in JSON format to successfully perform the POST request. If authorization is required (with GET or POST method), the client adds a specific authentication key (i.e. session key) to the HTTP request Authorization header. Authentication of the request allows the API to restrict or limit the access to its resources. The user is identified by a user key or session key. After an API request has been processed, a HTTP response is sent back to the client. The response includes a corresponding HTTP status code and a body with response data in JSON format.

Researcher/Administrator Hub
The researcher/administrator hub allows researchers to upload their tasks and administrate them. Also system administrators can oversee the whole system through the hub. The hub is built using common Web technologies and languages like HTML, CSS, JavaScript and PHP. It allows researchers and administrators to easily see different database tables and manipulate them. For this purpose we used a jQuery plugin called DataTables [3]. It enables users to easily poll the data from the database and display it in simple yet flexible tables. Furthermore, PHP scripts were implemented to fetch the data from the database and the front end.

The two tabs of the researcher panel are depicted in Figure 6. In the first one, researchers can upload tasks by setting the name, description, reward, url and thumbnail. They can also see an overview on their current tasks and contact the administrators. In the second tab, researchers can explore all answers by the workers, which they can either approve/reject individually or in batches.

Figure 6: Researcher panel (approve answers tab).
Similarly, Figure 7 shows the administrator panel. Here, the platform administrator can perform a number of actions including: 1) approve/reject/delete tasks uploaded by the researchers, 2) update task rewards, 3) manage user/researcher accounts, 4) manage the shop (add items, edit price, update stock), 5) manage the purchases, 6) add/delete/update news items to appear on the client’s main screen, and 7) manage the information snippet that can be shown to users upon logging in as a dialog box.

![Administrator panel](users tab).

**Conclusion & Future Work**

An earlier version of the platform presented in this paper was used successfully to investigate the role of price-setting in a situated crowdsourcing market to control labour supply with a number of different tasks [11]. The results showed that a situated crowdsourcing market can attract a populous workforce with comparable quality of contributions to its online and mobile counterparts while maintaining higher task uptake.

In the future we will continue to improve the different components of Bazaar, as well as deploy several different crowdsourcing application that can benefit from a situated setting. We also would like to extend an invitation to other researchers that might want to collaborate and make use of our platform and/or infrastructure.

**Acknowledgements**

This work is partially funded by the Academy of Finland (Grants 276786, 285062, 286386), and the European Commission (Grants PCIG11-GA-2012-322138 and 645706-GRAGE). We also thank Nemanja Vukota, Teemu Partanen, Miika Keisu and Joonas Nikula for their technical contributions.

**References**


