San Jose State University

From the Selected Works of Magdalini Eirinaki

July, 2018

A Building Permit System for Smart Cities: A Cloud-based Framework

Magdalini Eirinaki, *San Jose State University*

Subhankar Dhar, *San Jose State University*

Shishir Mathur, *San Jose State University*

Adwait Kaley, *San José State University*

Arpit Patel, *San José State University*, et al.

Creative Commons License
This work is licensed under a Creative Commons CC_BY-NC-ND International License.

Available at: https://works.bepress.com/magdalini_eirinaki/50/

A Building Permit System for Smart Cities: A Cloud-based Framework

Magdalini Eirinaki*, Subhankar Dhar**, Shishir Mathur⁺,
Adwait Kaley*, Arpit Patel*, Akshar Joshi*, Dhvani Shah*

*Computer Engineering Dept, ** School of Information Systems & Technology, ⁺Urban &
Regional Planning Dept.
San Jose State University
San Jose, CA, USA

Corresponding Author Contact Details

Dr. Magdalini Eirinaki

Computer Engineering Department
College of Engineering
San Jose State University
One Washington Square, San Jose
95192-0180, CA, USA
email: magdalini.eirinaki@sjsu.edu
tel: +1 (408) 924-3828
fax: +1 (408) 924-4150

A Building Permit System for Smart Cities: A Cloud-based Framework

Abstract— In this paper we propose a novel, cloud-based framework to support citizens and city officials in the building permit process. The proposed framework is efficient, user-friendly, and transparent with a quick turn-around time for homeowners. Compared to existing permit systems, the proposed smart city permit framework provides a pre-permitting decision workflow, and incorporates a data analytics and mining module that enables the continuous improvement of both the end user experience and the permitting and urban planning processes. This is enabled through a data mining-powered permit recommendation engine as well as a data analytics process that allow a gleaning of key insights for real estate development and city planning purposes, by analyzing how users interact with the system depending on their location, time, and type of request. The novelty of the proposed framework lies in the integration of a pre-permit processing front-end with permit processing and data analytics & mining modules, along with utilization of techniques for extracting knowledge from the data generated through the use of the system. The proposed framework is completely cloud-based, such that any city can deploy it with lower initial as well as maintenance costs. We also present a proof-of-concept use case, using real permit data from New York City.

Keywords—building permit; data analytics; data mining; recommender system; urban planning; cloud computing; smart city.

1. Introduction

Construction projects typically require a building permit, a plan check, and inspections. Building permits are required when any type of construction is planned that will change or add structure to an existing property or land parcel. Often, more than one permit is required even for a simple construction project, such as updates to a bedroom or a bathroom. Further complications arise because even after multiple visits to a city’s permit office it is often very difficult for a permit-applicant to determine whether a permit is required. The type of permits, rules and regulations vary significantly between cities, making the process for obtaining building permits complex, non-transparent, and inefficient, often involving a significant amount of “legwork” for the citizens [1, 2, 3]. Furthermore, the application process itself is unclear. If a property owner decides to obtain a permit, the permit application process involves filling out several forms, providing many supporting documents, and often a long wait for a decision on the permit application [4, 5, 6]. Due to this lack of clarity, frustrated property owners often end up making unauthorized improvements to their properties, thereby increasing public health and safety risks [4, 5, 6, 7]. At the city-government-end, the permit-application-generated workflow is slow, difficult to track, and requires coordination spanning several departments (for example, departments of building inspection, public works, and fire protection) [3, 8, 9].

While many parts of everyday life have been largely automated, including various city services (e.g., filing complaints for illegal dumping), a recent survey of 524 cities in the US found that only 21% cities offered an on-line permitting platform [8]. This proportion is likely to

increase due to a renewed focus on public sector efficiency and transparency, especially with the recent open data and open government initiatives. However, a systematic review of the capabilities of these on-line systems is lacking, especially in the US. If studies from Europe are any indication, on-line permitting systems are very limited [10]. In this work, we attempt to address the above issues and shortcomings, by proposing a “smart permit” framework that any city can implement to streamline and facilitate the permit request process.

As noted in [11], smart city represents one of the most promising, prominent, and challenging Internet of Things (IoT) applications. According to a recent survey [12] the smart city services should be user-centric, ubiquitous, and highly integrated from the end user’s perspective, while being highly interconnected, cost-efficient, and energy-efficient from the point of view of the administrators, and this can be enabled by a “Cloud of Things” (CoT) infrastructure. While this cloud-based approach is broadly accepted as the optimal way of implementing such applications ([13, 14, 15, 20, 21]), this is the first work that focuses on the smart building permit application domain.

Briefly, this work has two objectives:

a) To develop a cloud-based architecture to support the permit process and address the shortcomings of most existing permit systems by making it more transparent, efficient, user-friendly, and with quick turn-around time for property owners. As further discussed in Section 3, contrary to the majority of existing systems that automate the permit process starting with the submission of a permit request, this work focuses also on pre-permitting. The novelty also lies in employing data analytics and data mining techniques to improve the permit applicant’s experience as well as the process itself. Therefore, this paper develops a framework that streamlines the permitting process by introducing a business-rule-driven pre-permitting system that allows permit applicants to identify, through a user-friendly interactive interface, the exact permits and the associated paperwork. Additionally, the framework incorporates a recommendation engine that inputs past permit requests to a collaborative filtering process in order to generate permit recommendations to new users. For the city government, the framework provides an analytics module that incorporates various permit-related functionalities to achieve usability, reliability, transparency, performance guarantee, and reduction in processing time—characteristics that are extremely important for both city governments and property owners. Finally, the proposed framework is completely cloud-based, meaning that the system does not need to be installed on a local server, but instead is managed by a cloud service provider (like AWS, Azure, etc.). Our proposed technology stack is completely new, highly scalable, and based on emerging open source technologies. This design decision is significant given the budget constraints many cities face, allowing any city to deploy the framework faster and with much lower cost, making it easier to maintain (no need to install software, additional hardware, or do the necessary updates), and enabling access from everywhere given Internet connection.

b) To present a proof-of-concept implementation, using real-life permit data (this paper uses permit data collected by the New York City [16, 17]). The prototype showcases the novel functionalities of the proposed framework and highlights ways in which both the public and the city officials can benefit from such a system.

The rest of the paper is organized as follows: In Section 2 we provide a brief overview of the building permit process. In Section 3 we discuss related work in the field of on-line permitting. In Sections 4 and 5, we discuss the high-level architecture and functionality of the main modules of the proposed framework, namely the pre-permitting process, the permit functionality, and the data analytics and mining processes. The proof-of-concept prototype using real-life data from New York City is presented in Section 6, and we conclude with our plans for future work in Section 7.

2. Building Permit Process Overview

Viewing the building permit process from an on-line, cloud-based, permit-system-design perspective, we identify three major phases: pre-permitting, permitting, and post-permitting. In this section, we outline permit process workflow, introducing important concepts. In the sections that follow, we focus on the pre- and post-permitting phases, proposing how these can be streamlined, enhanced, and improved.

The following needs to be ascertained during the pre-permitting stage: a) whether any permit(s) is (are) required for a building project; b) if yes, the kinds of permits required; c) the required application forms and supporting documents; d) whether sample application forms and supporting documents are available as exemplars; e) the specific steps to be undertaken and fee to be paid for obtaining permits; and f) how changes in the design and/or scope of the building project would impact points a) through e). If the applicant is provided above-described information through an on-line, user-friendly platform that provides answers customized to each building project, it would help the permit-issuing city government department reduce the time and resources devoted to in-person interactions with the applicant at the pre-permitting stage. Furthermore, this data could provide the city government leading indicators of the demand for permits across various dimensions such as permit-type and geographical location of the building projects. Success of such an on-line platform hinges on strong linkages between the front-end of the system with the back-(city government)-end of the system. For example, the system should be able to link the data provided by the applicant (such as the project address, building type, and project type) to the project's zoning, applicable building code, and design in order to ascertain the permit requirements.

During the permitting stage, the applicant submits the permit application, including the fee. A permit is issued once the application is reviewed and approved by various city government departments. During the construction stage, building inspectors may periodically visit the project site to verify the project's adherence with applicable building regulations and other rules. The inspectors issue completion certificates once the entire projects are complete as per rules. During

this phase, a contact person is typically assigned to an applicant. This contact person is usually a city staff (often from the city's permitting department).

At the city-government-end, a complete building permit application in a format that is compatible with the application review systems of the various reviewing departments (such as the planning, police, fire, and public works departments) is desirable. Further, the back-end system should enable: a) the reviewing departments to view, comment, and adjudicate upon an application, b) the building inspectors to schedule visits, c) the coordinating staff to track the application status and communicate with the applicant and the city government staff, d) automatic generation of notifications (such as, assigning a project case number and reminding about an upcoming building inspection), and e) the coordinating staff to receive notifications, for example, regarding application updates including an applicant's queries and feedback, and building inspector's notes and decisions/approvals.

During the post-permitting phase, the applicants could desire to: a) access history of past permit applications, including the submitted applications and supporting documents and notifications from the city government; and b) have the ability to reference past permits while applying for new permits. Additionally, city residents might be interested in understanding the real estate development trends in their city and use permit data to gauge these trends.

3. Related Work

According to a recent survey of 524 cities in the US, only 21% cities offered an on-line permitting platform [8]. Unfortunately, a systematic review of the capabilities of these on-line systems is lacking, especially in the US. Studies in Europe identify a similar trend. For example, no other European country, apart from England and Wales, offers the ability to submit building permits electronically on a large scale [10]. In the absence of an existing systematic review of online-building permit systems, we conducted a review of on-line permitting systems of three large cities in the US—Chicago, Cincinnati, and San Jose—and found Cincinnati's system to be the most advanced among the three, especially because it was the only case study city with pre- and post-permitting modules. The Cincinnati's pre-permitting module describes the common building permit types. The module also contains a permit guide that provides answers to commonly asked questions such as whether a building project needs a permit and if yes, which supporting documents are needed. Furthermore, it provides samples of such supporting documents, including project drawings. Finally, through a simple address search it allows users to identify the types of projects for which permits applications can be submitted on-line. However, the module has limitations. For example, it does not provide customized answers to the applicants at the pre-permitting stage, which our smart permitting system aims to achieve. The post-permitting module allows review of permit records and links the permit application database to the city's Geographic Information System (GIS), thereby allowing easy spatial analysis of the permit data. For example, maps showing the permit issuances across the city by major permit types are available on-line [18].

Our proposed conceptual framework employs state-of-the-art design principles and technologies to implement the various components of the permitting process. The novelty of the proposed framework lies in a nimble pre-permitting module that provides customized answers to users and in the use of data analytics and data mining techniques, in combination with business process modeling, to improve both the process itself, as well as the user experience. There is little work done in this area. Huaiming et al. [19] proposed the use of agent technology to assist several administrative permit operations such as filling forms, auditing material, and deciding on permits. While the authors provide a very high-level description of the agents' desired primary characteristics (including residence, autonomy, responsive, pro-active, social), they do not provide any technical details on how these agents will be implemented and what will be the underlying algorithms that will enable them to perform the suggested functionality.

Ye et al. [20] proposed modeling the administrative permit businesses using business objects. Their focus is on describing the business rules and proposing the general business process of one-stop administrative permit system to increase the flexibility of the system. Eirinaki et al. [22] present a similar framework that integrates a data analytics and mining module, as well as a pre-permit system. In that work they only provide a high-level overview of the framework and cover the functional characteristics of the Permit Processing module in detail. In this paper, we adopt a very high-level workflow, assuming that it will be refined further after performing requirement analysis with the city who's going to implement the permit system.

A cloud-based IoT framework has been broadly accepted as the optimal way to design and implement smart city applications [11, 12]. In recent years, cloud-based solutions have been proposed for traffic management [12], harnessing and analyzing open city big data [14, 15], noise mapping [13], and waste management [21], among others. In this work we leverage and extend the cloud-based conceptual framework for smart permit systems that has been presented in [22].

The main modules of the proposed framework presented in this paper are the *Pre-permit & Permit System Front-end*, the *Permit Processing Module* and the *Data Analytics & Mining Module* that are integrated together to coordinate and streamline all business processes. Since previous work has covered the functional characteristics of the *Permit Processing Module* [22], this paper focused on highlighting how the permit application system can become "smarter" through pre- and post-permitting modules aided by data analysis and data mining/machine learning technologies at various steps of the process. To this end, we extend our work by providing additional details on the pre- and post-permitting data mining and analytics components, and presenting a proof-of-concept prototype implementation, employing real-life permit data from New York City.

4. A Framework for Smart Permit System

As mentioned previously, in this paper we focus on improving and enhancing the pre- and post-permitting processes. One of our objectives is to reduce the time for identification of the specific permit type in any scenario related to residential buildings and remove the need for professionals to perform the identification process, thereby reducing permit-applicant's financial burden. Leveraging past permit request data, the framework also provides recommendations for other similar permit types, which the user may require in tandem with the permit applicable to their selection. These recommendations should save time for the users in the future.

The proposed system will interface with a variety of systems such as a city's GIS, and integrate the following processes to improve efficiency: Planning and Zoning; Building Permit-Plan Review and Inspection; Code Enforcement Case Management; Engineering Permits, Fire Prevention Plan Review and Inspection; Special Events; and Customer Relationship Management.

4.1. Pre-permit & Permit System Frontend

The proposed smart pre-permit system would benefit the end users by simplifying and streamlining the process of identifying the building permits needed for specific tasks, thereby reducing the need to visit the city permit office or hire a consultant to navigate the city bureaucracy. The *Pre-permit & Permit System frontend* module, as shown in Figure 1, determines whether a user needs a permit. It consists of various questionnaires and forms along with a chatterbot—a computer program that interacts with the users via auditory or textual methods [23, 24].

The core component of the pre-permit system is a decision-tree-like rule-based engine, which determines the flow (in other words, the paths of the decision process flow) through a series of questions regarding their building/construction plans that an end user answers¹. The flow of questions is not static; instead, it adjusts dynamically depending on the answers provided by the end user. At the end-point of this flow, the user is notified about the type and subtype of the permit(s) per job-type required for the construction project. Figure 2 shows a generic flow of a pre-permit decision process, inspired by our case study (NYC building permit process). This rule-based engine, that guides the users at every step to help them identify the appropriate permits, consists a significant advance over existing permitting systems.

This module acts like a gateway to the *Permit Processing Module* by allowing the user to fill out and submit permits online. The data collected from the users' permit requests and their navigation on this system feed into the Web usage logs, and is subsequently used as input by the *Data Mining and Analytics module*. The output of this process, discussed later in this paper, is a

¹ Please note, this is not the same as the decision tree that the respective machine learning family of algorithms would generate. Instead, this is a decision process flow diagram derived by the underlying processes it is depicting.

set of permit recommendations to the end user. Finally, the city officials can collect user feedback and use it for improving the user experience and the overall permitting process.

Figure 1. *Pre-permit & Permit System Frontend*

Figure 2. Remodeling-Focused Decision Process Decision Flow Diagram for Pre-Permit Processing

4.2. Data Analytics and Mining

The novelty of the proposed framework lies in the improvements to both the end-user experience, as well as to the permit and planning processes through the utilization of techniques and algorithms for data analytics and data mining, and the use of recommendation algorithms to extract knowledge from the data collected using this system. Accordingly, the analytics module functions in two layers, namely the permit and planning process layer, and the user experience layer, as discussed next. Figure 3 provides a high-level architecture, highlighting how the various sub-modules of the analytics module interact with the other modules and data repositories of the framework.

Figure 3. Permit System Architecture - Data Analytics & Mining module

4.2.1. Improving the end-user experience

The design of the proposed permit framework is user-centric. The goal is to improve the user experience by streamlining the process as well as by improving the interactions of the users with the system. To this end, we apply data mining and collaborative filtering techniques to permit application data, web usage data, as well as explicit user feedback in the following ways:

- *Permit Recommendations:* The permit-related data can be used as input to a permit recommendation process. The premise behind recommendation systems is simple: if user A and user B have similar preferences (expressed, for example, in terms of ratings or purchases), then if user A likes item x, user B is also likely to like item x. We transfer this paradigm to the permit process, based on the assumption that similar users (for example,

users with similar houses and past permit requests) might have similar permit needs in the future as well. Based on this assumption, we propose the implementation of a permit recommendation system. Similar to the web-based recommendation systems, we can employ collaborative filtering techniques to generate recommendations for the active users, in the form of “other users who submitted permit X also submitted permit Y”. These recommendations can be provided in addition to the outcome of the pre-permit decision process, which is an initial list of suggested permits. To the best of our knowledge, this is the first work to propose a recommendation systems’ approach in this context.

- *Clickstream analysis*: The web usage logs can be mined to identify possible bottlenecks in how users interact with the user interface of the system. This knowledge can be subsequently used to redesign the workflow of the interface to minimize the number of dropped (i.e. unfinished) sessions.
- *Explicit feedback and opinion mining*: At the end of each session, the end users will have an opportunity to provide feedback for the system, both through numerical rating of their experience, as well as through textual comments. Opinion mining and sentiment analysis techniques can be applied to the textual comments to automatically identify issues/themes that concern the users, as well as features they liked. This knowledge can be used to improve the user experience.

4.2.2. Improving the permit and planning processes

The proposed system architecture includes a data repository where all permit-related interactions, including permit requests, types, location, and date are stored. Using predictive analytics and data mining techniques, city governments can gain useful insights on the needs of their permit applicants and more efficiently plan their resources. Some examples of applying data analytics on the permit data set, along with possible use of the results, include the following:

- *Spatiotemporal visualization of permit requests*: The city officials and the public will be able to visually explore which types of permits are more frequently requested in what parts of the city during which time period.
- *Statistical analysis and predictive modeling on permit request data*: The data can be used to generate a predictive model that will enable early identification of spikes in demand by permit type and by time. This would allow the city to allocate resources appropriately to avoid long wait periods for the permit applicants (e.g., allocate two more employees in March to process kitchen remodeling permits filed in District 9).
- *Statistical analysis and predictive modeling on permit processing data*: Data related to permit approvals or denials could be analyzed to identify possible bottlenecks in the workflow and provide insights on how the process can be further streamlined (for example, discover that a specific type of permit has a higher than average rate of denial).

5. Permit System Architecture and Design

We have designed the proposed smart permit system to automate the current permit process and to recommend permit types, thereby helping a city transform into a smart city. The web-based smart permit system provides highly responsive and interactive user interfaces so that it is easier for the users to identify the permits needed for their projects. The intuitive user interface also enhances the user experience by allowing users to identify the type of permits needed, thereby reducing or obviating the need to request services of a consulting professional to file permits and saving multiple visits to the city permit department. The system attempts to simplify the principal task for both the end users by identifying the permits they need and for the city governments by saving the time needed to process the incorrect/incomplete permit applications.

5.1. System Architecture

Following the conceptual framework described in Section 4, we designed a three-tier system architecture, shown in Figure 4. We should point out that some components of the conceptual framework (e.g. opinion mining and web usage mining) are not part of the current prototype design and implementation, and are left as part of our future work. The three tiers include the *Pre-permit and Permit System Frontend* (client), the *Business Logic* (middle tier), and the *Recommendation Engine* (backend). The three tiers interact and interface as follows:

1. The permit applicant (end user) interacts with the system using *Pre-Permit & Permit System Frontend User Interface*. The applicant will either sign up or sign in to the system through the *Registration* sub-module.
2. Every user request is routed to the system via the *Business Logic layer*.
3. A server that acts as a controller and calls the *Decision-making module* in the system then processes the request.
4. The *questionnaire* is displayed to the applicant who responds to these questions. The decision-making module decides the type of questions and the series of questions displayed to the user.
5. After recording all the responses of the applicant, the system identifies the required permit type(s) that are appropriate for the end user's needs. The permit(s) are presented to the end user.
6. Concurrently, and as soon as the required permit type(s) are identified, they are used as input to the *Recommendation Engine*.
7. The Recommendation Engine uses the *item-based collaborative filtering* algorithm to identify relevant permit types that similar applicants applied for in the past.
8. After processing relevant permit types, the system recommends additional permit types that the applicant may require in the future.

Figure 4. Smart permit system design and technologies

5.2. System Prototype Technologies

Due to the big amount of permit data collected on a daily basis by each city, to enable scalability and real-time availability, we employ various state-of-the-art technologies for the system prototype as described next.

5.2.1. Pre-Permit & Permit System Frontend (Client Tier)

The permit applicant can access the smart permit system from the client interface that is a web application. Through the interface, the permit applicant can login or register with the system, as well as view and answer the permit type questionnaire. The succeeding set of questions in the questionnaire is fetched from the backend server via REST APIs in the JSON data format. The main logic of the client-tier implementation is written with AngularJS framework. We have tried to make this a single-page application. In a single-page application, the aim is to load the HTML and CSS once initially. Then just the content or little HTML of the application is updated and the entire page is not loaded again. This helps improve the network bandwidth and provides quicker page load to the end user. The end users experience faster

browsing and do not have to wait long to view content when navigating between the pages. Additionally, HighCharts and Google Maps are used for data analytics and visualizations.

5.2.2. Business Logic (Middle Tier)

The middle tier hosts the main business logic for the entire permit system. The permit processing module is hosted entirely on Amazon EC2, which is a cloud-based infrastructure for high performance, scalability, and security. This tier consists of two components: Node.js and Java applications. The Node.js application is a web application that provides an interface to the user to access and benefit from the Smart City Pre-permit and Permit System frontend. The predefined decision rules, which are an integral part of the system, are stored in the middle tier. The Node.js application manages dynamic questionnaire and identifies the permits needed by the user. The Business Logic tier applies predefined rules on user responses to achieve both of the above functionalities. User authentication, user session, and permit recommendations are handled here. The system employs the state-of-the-art Apache Mahout (Java) libraries for item-based collaborative filtering.

5.2.3. Server Side (Backend Service Tier)

For the proof-of-concept implementation, we employed permit issuance data acquired from the open dataset provided by the city government. The open data are made available for public use by Socrata through RESTful APIs. The columns are filtered from the source data table as per the system requirements with SOQL (Socrata Query Language) functionality available. The free service of Socrata restricts the amount of data that can be collected at one time to one thousand records. We saved these data in flat files. We then cleaned the collected data to remove null values and redundancies. Next, we split and saved the clean data in respective MySQL database tables. Then, we created a data model from these tables to feed it to the recommendation engine to generate permit recommendations. Additionally, this layer consists of the geolocation data. Finally, we mapped the geolocation data from the area zip codes available in permits open data and translated to respective location coordinates.

5.3. Recommendation Engine

Collaborative filtering techniques are based on the principle that users with similar needs prefer similar items. In the building permit system context, this means that users who need the same types of construction work in their houses, will most probably need the same set of permits. In order to generate permit recommendations, we employed item-based collaborative filtering. This type of filtering has proven superior to the user-based collaborative filtering due to its ability to scale well and work with sparse user-item matrices (i.e. less knowledge for each user). This is because in item-based collaborative filtering, similarities are calculated among items, not users.

More specifically, each user profile is represented as a vector $u = \langle p_{u,1}, \dots, p_{u,m} \rangle$, where $p_{u,i}$ represents a permit subtype i filed by user u and m is the total number of distinct permit subtypes offered by the city. The vector can be binary (i.e. whether the user has submitted the permit p_i or not), or weighted (representing how many such permits have been submitted by one user). Each user can be uniquely identified either explicitly by their login credentials, or implicitly, using their house address.

The existing user profiles form an $n \times m$ user-permit matrix (where n is the total number of users U in our database). We then calculate the permit similarities $sim(p_s, p_t)$ for all permits that were filed together by users in the past. This can be done in multiple ways, for instance using the Pearson correlation coefficient:

$$sim(p_s, p_t) = \frac{\sum_{u \in U} (p_{u,s} - \bar{p}_s)(p_{u,t} - \bar{p}_t)}{\sqrt{\sum_{u \in U} (p_{u,s} - \bar{p}_s)^2} \sqrt{\sum_{u \in U} (p_{u,t} - \bar{p}_t)^2}}$$

where $\bar{p}_s$ represents the average number of permits of this subtype filed by the users in U (this normalization can be especially useful in case of apartment buildings that have different permits for each apartment, but the same lot). Other similarity metrics (e.g. Tanimoto coefficient) can also be used. A high similarity score implies that many users filed the two permits together. So in turn, these can be used to predict which other permits a user might need to file, given their current search, by calculating the following “weight” for each “missing” permit subtype j for user u (i.e. permits that the user has not yet submitted):

$$pred(p_{u,j}) = \frac{\sum_{i=1}^m [sim(p_{u,j}, p_{u,i}) * p_{u,i}]}{\sum_{i=1}^m sim(p_{u,j}, p_{u,i})}$$

The higher the weight, the more likely that the user would need the respective permit. The top- n ranked “unseen” permits are recommended to the user. In our prototype, we have used the Apache Mahout implementation of the item-based collaborative filtering algorithm, because it is one of the most stable implementations (at the time of developing this framework), and it integrates very well with the rest of the framework since it is written in Java.

6. Use Case: New York City Permit System

After considerable research and exploration, we ascertained that New York City open data website was suitable for illustrating proof-of-concept for our proposed system. For finding relevant permit types, we used New York government’s building permit data set available through NYC Open Data [16, 17]. The dataset currently includes 662,000 records and consists of 54 columns, covering permit-related information, such as, Borrow (e.g., Manhattan, Bronx etc.), House number and street name, zip code, Job type, Permit type and subtype, Filing date, Issuance date, and Expiration date. Figure 5 shows a snapshot of this dataset available in public

domain. While our prototype employs the NYC permits dataset, the application can work with other similar datasets as long as the same type of information is available.

Consistent with the system architecture presented previously, the prototype includes two main components: an Analytics/Visualization module, and a Pre-permit and Permit module, incorporating a recommendation engine. Next, we discuss how these are implemented in the prototype.

NYC OpenData 1500+ Data Sets Available

DOB Permit Issuance
A list of permits issued for a particular day and associated data. Prior weekly and monthly reports are archived at DOB

	BOROUGH	Bin #	House #	Street Name	Job #	Job doc. #	Job Type	Self_Cert
1	QUEENS	4400979	30	PELHAM WALK	420050864	01	NB	
2	BROOKLYN	3203302	2463	EAST 22 STREET	321477184	01	A2	Y
3	MANHATTAN	1000632	5	HANOVER SQUARE	122939632	01	A2	Y
4	MANHATTAN	1042369	160	EAST 65TH STREET	140300948	01	A2	
5	STATEN ISLAND	5042739	211	CLIFTON AVE	520218504	01	A1	Y
6	MANHATTAN	1014387	1	PENN PLAZA	122909086	02	A2	Y
7	BROOKLYN	3031589	715	NOSTRAND AVE	320840890	01	A2	Y
8	QUEENS	4020577	26-18	18 STREET	440103193	01	A3	Y
9	BROOKLYN	3399039	2394	DEAN STREET	320763063	01	A2	
10	MANHATTAN	1080722	547	10 AVENUE	122534169	01	A3	Y
11	BRONX	2009421	800	HUNTS POINT AVENUE	220411579	02	A2	
12	MANHATTAN	1006128	82	EAST 3 STREET	140533215	01	A2	Y
13	MANHATTAN	1083346	55	WATER STREET	122899090	01	A2	Y
14	MANHATTAN	1088646	56	LEONARD STREET	121452817	01	A3	Y
15	MANHATTAN	1087440	1955	1ST AVENUE	122674062	01	A3	
16	QUEENS	4010134	36-01	28TH AVENUE	420933921	01	NB	
17	BROOKLYN	3088367	626	LORIMER STREET	301919761	01	A1	
18	MANHATTAN	1022893	1221	AVENUE OF THE AMERICAS	122932032	02	A2	Y

Export

API Endpoint:
<https://data.cityofnewyork.us/resource/8>

Field Names:

BOROUGH	borough
Bin #	bin_
House #	house_
Street Name	street_name
Job #	job_
Job doc. #	job_doc_
Job Type	job_type
Self_Cert	self_cert
Block	block
Lot	lot

Figure 5. NYC Open Data Web Page

6.1. Analytics/Visualizations

The permit system prototype provides an interactive GUI that allows the city officials to perform exploratory data analysis that can be used as the first step in the big data analytics workflow and can be provided to the public as a more user-friendly interface to the open data. Currently, the prototype includes several different views of the same dataset. We have provided various trends based on season, expiration and popularity. To enable these visualizations we performed several data pre-processing tasks, such as deriving fields like year, month and quarter based on date fields, and converting the zip code to latitude and longitude using Google Maps API.

6.1.1. Seasonal Permit Trends

The first type of data visualization and exploration consists of seasonal permit trends. Figure 6 shows an example of this visualization. We grouped the eight permit types (Alteration, Demolition, Construction Equipment, Foundation Work, New Building, Plumbing, and Sign) among the four quarters and displayed them temporally. The user can also select a subset of the

permits and drill down on the time dimension (e.g. reviewing only the permits issued during specific quarters or months). This provides an easy and intuitive way to preview the seasonal trends. For instance, for year 2011 we observed that the permit applications peaked during the 2nd quarter and dropped during the 3rd, with “Other” and “Plumbing” being the most popular permits requested. Figure 7 shows an alternative visualization for quarterly popular permits.

Figure 6. Seasonal Trends for NYC (for year 2011)

Figure 7. Quarterly Trends (for year 2010)

6.1.2. Permits Expiration Analysis

Our prototype has an important functionality—the ability to predict spikes in permits demand by analyzing the expiration dates of already issued permits. This ability can help the city with

human resource planning, allocating more staff to the permit office during periods where a spike in demand is expected.

Similar to the permit trends data, the expiration data are organized by year and displayed by quarter and permit type. For example, looking at the data displayed in Figure 8, we observe that for year 2010 the majority of permits expired in the second quarter, with the “Alteration” permit type registering the highest number of expiring permits. Users can view the exact permit count for each data point in the graph by hovering on the data point as shown in the figure.

Figure 8. *Permit Expiration Analysis*

6.1.3. Geolocation analytics

Using the zip code and the longitude/latitude data for each permit request, we can also generate location-based analytics. For instance, the user can review how many permits have been filed per zip code/borough. Additionally, the data can be mapped to an actual New York City map, providing a way to review the dataset from different views. For this purpose, we used Google Maps APIs and their visualization library. Figure 9 shows the permit map for New York City. Each marker represents the number of permits issued for that location. By clicking on a particular marker the user can see the permit count for the particular location (grouped by zip code).

Figure 9. Geolocation Permit Map

Figure 10. Geolocation Permit Map by Permit type

A more detailed view of the same data is shown in Figure 10. This map drills down the same detail by permit type. Moreover, the size of the bubble is analogous to the number of permits, providing a very useful overview of permit popularity per neighborhood. The prototype implements additional geolocation-based visualizations that we have omitted from this paper due to space constraints.

6.2. Pre-permit process and recommendation engine

6.2.1. Pre-permit process

The client tier of the permit system is a web application that we currently host on a cloud-based server, Amazon EC2. Our goal is to provide a simple and easy-to-use user-interface for the end users. The interface is divided in two components—*End user* and *Admin*. The end user can benefit from the pre-permit process in that, through a predefined set of questions, the system will

guide them to the permits required for a specific set of jobs. Moreover, the end user can also view additional recommendations provided by the system. The admin panel of the permit system contains all the visualizations of the permit data obtained from the New York City's website as described in Section 6.1.

Figure 11: Decision Process Flow Diagram for NY City Bathroom Remodeling Use Case

As previously mentioned, one of the most important components of the pre-permit process is a rule engine that determines the flow of questions that are shown to each user. This flow of questions ultimately guides the user to the end-point, where the user is notified about the type of permit, the subtype of permit, and the job type required for their construction project. The rule engine forms a questionnaire series for decision-making. This happens at the backend and middle layer of the architecture.

In what follows, we demonstrate a use case of requesting permit/s for a bathroom remodeling construction project in New York City. The path down the decision process flow that determines the flow of questions in this scenario is highlighted in Figure 11.

Figure 12: Pre-permit Process – Target Area Remodeling

Figure 13: Pre-permit Process – Component or Subcomponent Selection

Figure 12 shows the third step of the process, assuming that the user has already selected “Remodeling” and “Residential” in the first two steps. In this step, the user may select one or more building areas that need remodeling or alteration. This question allows the user to select one or more options. Depending on the user’s selection(s), the next step will display various options on types of alterations available (e.g. for “Bathroom” this would include

“Addition/Installation”, “Demolition”, “Relocation”, “Repairing/Styling”). The user has again the option to select one or more alteration types.

In our example, we assume that the user selects “Addition/Installation”. Figure 13 shows the pre-permit question regarding the element that the user intends to add to their bathroom. The two options are “Component” and “Subcomponent”. A Component is a major addition to an existing structure. Examples of a Component include "Bathroom", "Basement", "Walk-in Closet", and “Fence”. On the other hand, a subcomponent is a minor addition to an existing structure. Examples of Subcomponent include "Bathtub", "Faucet", "Bulb", "Switches", "Vent", and “Stove”. This explanation is presented to the user on the browser as a popup window that pops whenever the user clicks on the (?) icon. The explanation is made clearer to the user through the use of images. Again, in this step the user is able to select one or more options.

The system cycles through the decision process flow to cover all choices made by the user (in terms of areas, types of work, components, and sub-components). In the end, the system generates a summary page, outlining all the permits that are required based on each user’s needs. In our running example, we assume that the user has decided to add “Lighting/Switching” and “Sprinkler” as sub-components. Figure 14 shows the summary page for this user. The application suggests permit type “Equipment work” for both appliances: lighting/switching and sprinkler. According to the application, for lighting/switching, the user needs “Mechanical Work” and “Other” permit subtype. Whereas for sprinkler, the user needs permit for “Plumbing” along with “Equipment Work”, “Mechanical”, “N/A” and “Other” permit subtypes. Both these appliances fall under job type “Alteration 2”. There is also a button to apply for these permits below the results (not shown). On clicking on this button, a form will appear so that the user can fill to request the applicable permits suggested by the system.

As per your responses, you need following permits

Appliance	Permit
LIGHTING/SWITCHES	Job Type: A2 (Alteration 2)
	Permit Type: EW (Equipment Work)
	Permit Subtype: MH (Mechanical)
	Job Type: A2 (Alteration 2)
	Permit Type: EW (Equipment Work)
	Permit Subtype: OT (Others)
SPRINKLER	Job Type: A2 (Alteration 2)
	Permit Type: EW (Equipment Work)
	Permit Subtype: MH (Mechanical)
	Job Type: A2 (Alteration 2)
	Permit Type: PL (Plumbing)
	Permit Subtype: N/A (Not Applicable)
	Job Type: A2 (Alteration 2)
	Permit Type: EW (Equipment Work)
	Permit Subtype: OT (Others)

Figure 14: Pre-permit Process – System Generated Required Permit

6.2.2. Permit Recommendations

In the system prototype, we employed the Apache Mahout libraries to implement the recommendation engine, detailed in Section 5.3. The engine incorporates a data pre-processing component that performs data extraction and cleaning prior to feeding the data to the collaborative filtering algorithm.

The data extraction module is responsible for extracting data from the data source located at the official data store of the New York City, Department of Building Permits. The dataset is publicly available and is exposed via REST endpoints using SOCRATA API [16, 17]. This API provides different configurable components to enable extraction based on the users' selection criteria. As in the case with any real-life dataset, the NYC permit dataset had several issues, including noise (random entries), duplicate entries and inconsistencies, making data cleaning an important part of the process.

The particular implementation employed by our prototype uses Tanimoto coefficient to calculate item-based similarities, used subsequently to form the user neighborhood and generate recommendations but can be easily adapted to employ other similarity metrics (e.g. Pearson or cosine). The final step of the recommendation engine occurs real-time, as soon as the user selects a permit, or the pre-permit process generates a list of permits that are relevant to the user's needs. The recommendation engine selects additional permits that are similar to the one(s) selected, and presents them in a separate section, as recommendations. Figure 15 shows a screenshot of the output of this process. The top portion of the screen (not shown) shows the permit that the user has selected to apply. The bottom part, shown here, includes the automatically generated recommendations.

Figure 15: Pre-permit Process – Permit Recommendations

6.2.3. *Applying for permits*

In the end of the process, the user is able to submit a permit request via the New York City permit application system, available online. The prototype links directly to the City’s online forms, via the “Apply for Permits” link.

7. Discussion and Conclusions

In this paper, we have presented a cloud-based smart permit system that is more efficient, user friendly and comprehensive than the existing smart permitting systems in various ways. First, through a highly interactive and intuitive web application, the permit system’s pre-permit module provides users customized guidelines regarding which permits to apply. Second, the system uses permit request data as well as the permit processing data to conduct data analytics through visualizations and predictive modeling that provides actionable insights to the city officials. Third, applying collaborative filtering on past data and each user’s permit needs (as identified by their browsing and answers to the questionnaire), the system generates personalized recommendations of other permits that the users might need. Finally, the proposed framework provides a scalable cloud-based solution based on the best-of -the-breed technologies that can help move toward the vision of smart cities in a scalable and cost-efficient way. To the best of our knowledge, this is the first framework that includes a fully automated pre-permitting process supported in the back-end by business analytics and collaborative filtering algorithms. The modular design of the framework enables for easy updates to the back-end engines, in case such changes are required. Moreover, following the paradigm of other smart city applications that leveraged the advantages of a cloud-based design, our framework is completely cloud-based. We should point out that, even though such a solution is much more scalable and cost-effective for a city, it still includes the cost of hosting the application on the cloud. However it delegates the need of day-to-day maintenance to a third entity.

Apart from outlining the design and architecture of such a smart city framework, we have also implemented a system prototype, employing real-life data available from the New York City government open dataset for building permits of residential buildings. The system prototype implements the major functionalities of the proposed framework, including the pre-permitting process, the recommendation engine, and the data visualization, and provides a proof-of-concept that other cities can follow if they want to deploy a similar system. The proposed framework can be applied to any city, as long as the required data are available.

As part of our future work, we plan to expand the existing functionality to cover additional permit types, other than just residential permits and implement additional functionality in the prototype. One such extension includes the ability for users to leave verbal feedback, very similar to a “review”, on the usability and usefulness of the system. We will then apply aspect-based opinion mining on the collected reviews to identify components (i.e. aspects) of the system

as well as the entire pre-permitting and permit process that the citizens think are not working as well as they should, giving the city another way to identify and address problems early.

References

- [1] Frits Meijer, Henk Visscher, “The Deregulation of Building Controls: A Comparison of Dutch and other European Systems”, *Environment and Planning B Urban Analytics and City Science*, Vol 25, Issue 4, 1998, pp. 617-629.
- [2] Peter May, “Regulatory Implementation: Examining Barriers From Regulatory Processes”, *Cityscape*, Vol. 8, No. 1, 2005, pp. 209- 232.
- [3] Jacob Barker, “City reforming building permit system”, *Columbia Business Times*, September 19, 2008. Retrieved from <http://columbiabusinesstimes.com/2008/09/19/city-reforming-building-permit-system/>
- [4] Mona Mahrous Abdel Wahed, Requirements for establishing an information system to manage issuing building permits, *HBRC Journal*, Volume 13, Issue 1, 2017, pp. 83-88
- [5] Hyunsoo Lee, Jin-Kook Lee, Seokyoung Park, Inhan Kim, Translating building legislation into a computer-executable format for evaluating building permit requirements, *Automation in Construction*, Volume 71, Part 1, 2016, pp. 49-61
- [6] Mona M. Abdel Wahed, Mohammad M. El Barmalgy, Mohammad Reda Haggag, “Towards an advanced mechanism to benefit from information systems in issuance of building permits”, *HBRC Journal*, Volume 8, Issue 1, 2012, pp. 58-63
- [7] Stephen Agyeman, Herbert Abeka, Sampson Assiamah, “Are the challenges in the processing of building permits a precursor for development of illegal structures in Ghana?”, *US-China Law Review*, 13 (4), 2016, pp. 337-350.
- [8] William Riggs, Abhijeet Chavan, Chris Steins. “City Planning Department Technoogy Benchmarking Survey 2015.” Planetizen, 2015. Retrieved from <http://www.planetizen.com/node/73480/city-planning-department-technology-benchmarking-survey-2015>
- [9] Massachusetts Municipal Lawyers Association, “Small Business Permitting Reforms in Senate 2144, “An Act Promoting the Planning and Development of Sustainable Communities”, Zoning Reform Working Group - September 25, 2015. Retrieved from <http://www.massmunilaw.org/wp-content/uploads/2016/03/S2144-Permitting-Reforms-in-S-2144.pdf>
- [10] Christos Bellos, Kleopatra Petroutsatou, Leonidas Anthopoulos, “Electronic building permission system: The case of Greece”, *Procedia Engineering*, vol. 123, pp. 50-58, 2015.
- [11] Taewook H, Kwangsoo K, Hyunhak K, Changwon L, Jae HR, Youn TL, Jong AJ, Chulsik P, Seung-Mok Y, JeongGil K, “Escaping from ancient Rome! Applications and challenges

- for designing smart cities”, Transactions on Emerging Telecommunications Technologies, 25(1), 2014, pp. 109–119
- [12] Petrolo Riccardo, Loscri Valeria, and Mitton Nathalie, “Towards a smart city based on cloud of things, a survey on the smart city vision and paradigms”, Transactions on Emerging Telecommunications Technologies, 28: e2931, 2017. doi: 10.1002/ett.2931
 - [13] J. Jin, J. Gubbi, S. Marusic and M. Palaniswami, "An Information Framework for Creating a Smart City Through Internet of Things," in IEEE Internet of Things Journal, vol. 1, no. 2, pp. 112-121, April 2014.
 - [14] Chaowei Yang, Manzhu Yu, Fei Hu, Yongyao Jiang, Yun Li, “Utilizing Cloud Computing to address big geospatial data challenges”, In Computers, Environment and Urban Systems, Volume 61, Part B, 2017, pp. 120-128.
 - [15] Zaheer Khan, Ashiq Anjum, Kamran Soomro, Muhammad Tahir, “Towards cloud based big data analytics for smart future cities”, Journal on Cloud Computing (2015) 4:2. <https://doi.org/10.1186/s13677-015-0026-8>
 - [16] NYC Open Data, “DOB Permit Issuance”. Retrieved June 1, 2016, from <https://data.cityofnewyork.us/Housing-Development/DOB-Permit-Issuance/ipu4-2q9a/data>
 - [17] Zip codes states, GasLamp Media, (2016). [CSV Data file]. Retrieved from https://www.gaslampmedia.com/wp-content/uploads/2013/08/zip_codes_states.csv
 - [18] Cincinnati Area Geographic Information System (CAGIS), “ezTrak online services”. Retrieved from <http://cagis.hamilton-co.org/opal/> (accessed April 07, 2016).
 - [19] Li Huaiming, Ma Tian, Wang Ning, “The research of Network Administrative Permit Management System based on Agent technology”, IEEE/WIC/ACM Intl. Conf. on Web Intelligence and Intelligent Agent Technology Workshops (WI-IATW’06), 2006.
 - [20] Xin Ye, Na Wang, Yanzhang Wang, Hui Li, Shengju Han, “Research and Application on Business Rules for One-stop Administrative Permit System”, IEEE/WIC/ACM Intl. Conf. on Web Intelligence and Intelligent Agent Technology Workshops (WI-IATW’09), 2009.
 - [21] Sadia Sharmin and Sikder Tahsin Al-Amin, “A Cloud-based Dynamic Waste Management System for Smart Cities”, in Proceedings of the 7th Annual Symposium on Computing for Development (ACM DEV '16). ACM, New York, NY, USA, 2016.
 - [22] Magdalini Eirinaki, Subhankar Dhar, Shishir Mathur, “A Cloud-based Framework for Smart Permit System for Buildings”, in Proceedings of IEEE Smart City Conference (ISC2), Trento, Italy, September 2016.
 - [23] Jizhou Huang, Ming Zhou, Dan Yang, “Extracting Chatbot Knowledge from Online Discussion Forums”, Intl. Joint Conf. on Artificial Intelligence, 2007.
 - [24] Francesco Agostaro, Agnese Augello, Giovanni Pilato, Giorgio Vassallo, Salvatore Gaglio, “A Conversational Agent Based on a Conceptual Interpretation of a Data Driven Semantic

Space,” AI*IA 2005: Advances in Artificial Intelligence, Volume 3673, Lecture Notes in Computer Science, pp. 381-39.