A System for Recommending Rental Properties

Bernard Shibwabo Kasamani, Guy Nahimana

Strathmore University, Kenya

bshibwabo@strathmore.edu, guyniclas@gmail.com,

DOI: 10.20470/jsi.v8i2.310

Abstract: This paper presents an implementation of recommender technology to online search of rental properties. In particular, the paper uses the preference-based search approach combined with a technique called example-critiquing. Rather than perform a query against the database, this approach prompts the user to express some preferences on rental properties, uses them to construct a preference model for the user, and finally generates a list of properties that best match that preferences. The system is developed as Web application using the Ruby on Rails framework.

Keywords: Preference-based search, example-critiquing, Recommender System, Property Search

1. Introduction

With increased access to Internet through personal computers and mobile devices, many individuals turn to the Web to search for items of interest to them. This equally applies to searching for rental properties in Kenya where only 18 percent of urban dwellers own their homes (World Bank, 2011). However, the search for rental properties can be a challenging task. Prospective tenants may spend a significant amount of time exploring various websites advertising properties for rent. Yuan, Lee, Kim and Kim (2013) found that searching real estate assets online does not benefit homebuyers in terms of time, flexibility and intuitive results.

Search engines generally do a good job by helping users find what they are looking for, however many people find it hard to match their preferences to a query that is likely to produce the desired search results, Viappiani and Faltings (2006) and Viappiani et al., (2006). Using recommender systems technology can address the problem of mapping user's preferences to items that are likely to meet them (Viappiani et. al, 2006).

A recommender system can be defined as any system that produces individualised recommendations as output or has the effect of guiding the user in a personalised way to interesting or useful objects in a large space of possible options, Burke (2002). Automated recommender systems emerged early 1990s. A good example is Amazon that uses the collaborative filtering approach in making recommendations on the basis of the current user's behaviour and that of other similar users, Ekstrand, Riedl and Konstan (2010). Netflix uses a combination of content-based filtering and collaborative filtering to make recommendations based on the current user's preferences and those of similar users.

Traditional approaches such as collaborative filtering and content-based filtering are not suitable for high-value items such as electronics, vehicles and real estate assets since they are not purchased as frequently as are other objects. Consequently, buyers are not able to leave a sufficient number of reviews (ratings) on this objects to facilitate useful recommendation to other users, who may not be satisfied with years-old ratings, according to Felferning, Friedrich, Jannach and Zanker (2011). Knowledge-based recommender systems address this challenge by exploiting explicit user's requirements and the underlying product domain knowledge to generate recommendations, Felfering et al (2011).

One particular type of such systems, critiquing-based recommender system, will be used to implement the recommender system for rental properties proposed by this paper. Critiquing-based recommender systems have emerged and have been broadly recognised as an efficient preference-based recommender technology using a feedback mechanism called critiquing (Chen and Pu, 2012).

2. Literature Review

2.1 Rental Property Market in Kenya

The majority of Kenyan urban dwellers live in rented properties. According to World Bank estimates in 2011 only 18 percent of Kenyans living in cities lived in their own homes. This situation has led to the significant size of the market for rental properties in the country. The Hans Property Index estimated in 2014 that up to three quarters of people who bought apartments, intended to let them, (HassConsult Limited, 2014). The entire real estate market in Kenya was estimated at USD 4.5 billion in 2014, (The Standard, 2014).

A number of factors explained the low rate of homeownership in Kenyan cities. The first factor is the high population density in Kenyan cities: the World Bank estimated the urban population in Kenya to be about 11.36 million in 2016. This increases demand for houses thus pushing prices upwards. The second factor is the purchasing power of a growing middle class that likewise puts upwards pressure on home prices, (Kenya Bankers Association, 2015). The third factor is that the demand for housing that outweighed the supply by at least 156,000 units in 2011 (The World Bank, 2011).

Other factors that affects house prices include the number of bedrooms, the number of bathrooms, the type of the house, the number of floors, the location of the house, the presence of domestic servant quarters (DSQ), swimming pool, the age of the house, garage, proximity to a mall, number of parking bay(s), balcony(s), gymnasium, master ensuite, borehole, fireplace, garden/backyard, and separate dining room, among others (Kenya Bankers Association, 2017).

2.2 National Housing Policy in Kenya

The Government of Kenya, through its National Housing Policy for Kenya, recognises the need for housing as an important component of achieving adequate standards of living. It does so by subscribing to ideals expressed in various international conventions and instruments such the Universal Declaration of Human Rights of 1948, the International Covenant on Economic, Social and Cultural Rights of 1966, the Istanbul Declaration on Habitat Agenda of 1996, and the Declaration of Cities and other Human Settlements in the New Millennium of 2001 (Ministry of Housing, 2004).

The Government of Kenya also recognises the contribution of adequate housing and associated infrastructure to the dignity, the security and the privacy of individuals, families and communities. Furthermore, adequate housing contributes to the prevention of social unrest occasioned by people living in slums and other informal settlements (Ministry of Housing, 2004). Adequate housing also facilitates the reduction of poverty through employment creation, increased incomes, increased health and productivity of the labour force, (Ministry of Housing, 2004).

The National Housing Policy for Kenya has the following objectives, according to the Ministry of Housing (2004):

I. Enabling the poor to access housing and basic services and infrastructure required for a healthy living environment particularly in urban centers.

II. Encouraging integrated, participatory approaches to slum upgrading including income-generating activities that effectively combat poverty;

III. Promoting and funding of research on the development of low cost building material and construction techniques;

IV. Harmonising existing laws governing urban development and electric power to facilitate more cost effective housing development;

V. Facilitating increased investment by the formal and informal private sector in the production of housing for low and middle income urban dwellers;

VI. Creating a Housing Development Fund to be financed through budgetary allocations and financial support from development partners and other sources.

2.3 Challenges of Searching Rental Properties Online in Kenya

A prospective tenant looking for houses online is confronted by two main challenges: the vast space of options to consider, and the inadequacy of available tools to facilitate online searches. To find a rental property online in Kenya, one must explore numerous websites operated by property development and/or management firms in order to arrive at the most preferred one. This can easily lead to frustrations. Furthermore, most websites handle search through queries against the databases. Though these queries can be useful in helping users locate desired items, they do not take into account the preferences of the users thus making it more likely that many users may not find exactly what they are looking for.

2.4 Recommender Systems

Ricci, Rokach, Shapira and Kantor (2011) define recommender systems as tools and techniques providing suggestions of items of use to users. Various approaches to recommendation have been proposed by different scholars including collaborative filtering, content-based filtering, knowledge-based recommender systems, hybrid recommender systems, community-based recommender systems, and demographic-based recommender systems. The category of interest to this paper is a subset of knowledge-based recommender system referred to as critiquing-based recommender systems.

Chen and Pu (2012) identified three main types of critiquing-based recommender systems. Natural language-based recommender systems act as an artificial salesperson and interact with a client through a dialogue interface. System-suggested critiquing systems proactively generate a set of knowledge-based critiques that the user may accept as a way of improving suggestions. User-initiated critiquing systems show examples to users and stimulate users to make self-motivated critiques.

User-initiated critiquing systems aim at enabling users to make trade-off navigation; that is locating a product that has more optimal values on important features while accepting compromised values on les important attributes (Chen and Pu, 2012). User-initiated critiquing combines a preference-based tool and example-critiquing capabilities (Pu, Chen and Kumar, 2008). Preference-based search is a tool for the elicitation of users' initial preferences, whereas example-critiquing is an approach that enables users to refine their preferences so they can locate items that satisfy their requirements (Viappiani et al., 2006).

The user starts the process by expressing some preferences on products features in a search area. The system builds a preference model for the user based on the Multi Attributes Utility Theory (MAUT), which takes into account the conflicting value preferences and produces a score for each item to represents its overall satisfaction degree with the user preferences (Chen and Pu, 2012). The system then generates and displays a set of examples for the user to consider based on the user's preference model. The user either selects a desired items and terminates the process or provides feedback on the presented items (critiquing). Once the user revises the preference model by critiquing examples, the system will update its recommendation and return them for the next cycle of interaction.

Figure 1 provides a graphical representation of this user-system interaction cycle.

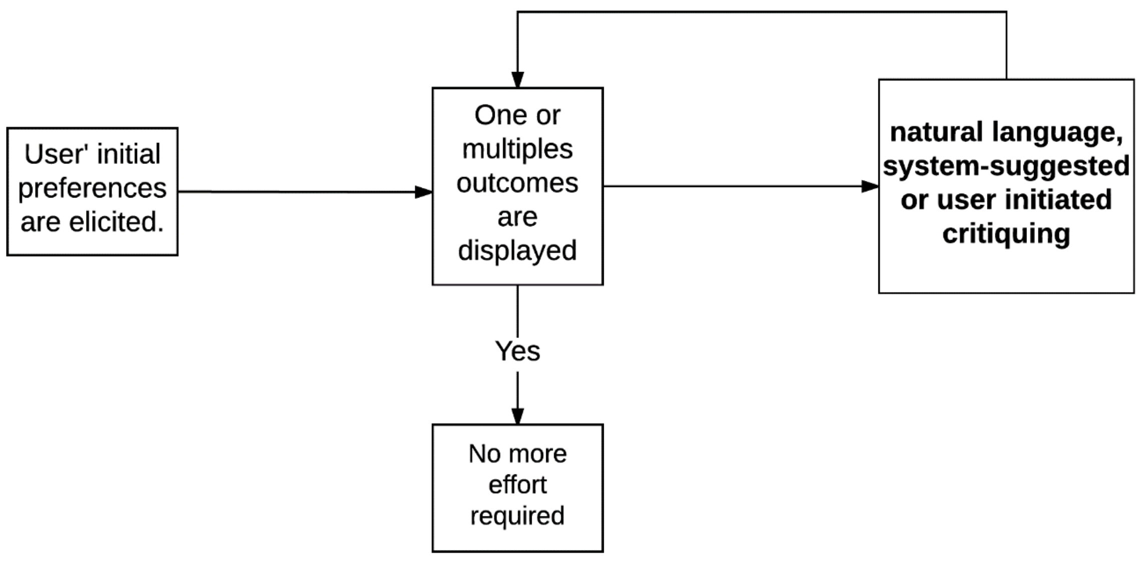

Figure 1: Example-critiquing interaction *(Adapted from Viappiani et al, 2006)*

Following the expression of initial preferences by the user, the system typically generates two sets of the examples: *candidates* and *suggestions*. Candidates are examples that are optimal for the current preference query, and suggestions are examples that are used to stimulate the expression of further preferences (Viappiani et al., 2006). Faltings, Pu and Torrens (2003) proposes that the number of examples displayed to the user should range from 5 to 20.

2.4.1. Generating Candidates

Items are modeled using a fixed set of attributes that each takes values in associated domains. Furthermore, a preference r is an order relation of the values of attribute a (Viappiani et al., 2006). To indicates whether or not an attribute of a particular item satisfies the preference on that attribute, we use cost functions. When the attribute of an item matches the preference of the user, it is mapped to the value of 0, otherwise it is assigned the value of 1 (Viappiani et al., 2006).

As a preference always applies to the same attribute, the notation $c_i(o)$ is used to express the cost that the function assigns to the value of option o for that attribute (Viappiani et al., 2006). Consequently, option o_1 is preferred to option o_2 with respect to preference i, if $c_i(o_1) < c_i(o_2)$. An additive form of value functions for all stated preferences, which is based on the multi-attribute utility theory, can be used to obtain the overall ranking of options, Viappiani et al. (2006) and Chen and Pu (2007). Thus, if $R_c = \{c_1, \ldots, c_2\}$ is the set of all cost functions of all preferences expressed by the user, the overall cost function for an item is given by $C(o) = \Sigma_{c_i \epsilon R_c} w_i . c_i(o)$. Option o_1 is preferred to option o_2 whenever it has a lower cost, i.e.: $c_i(o_1) < c_i(o_2)$, (Viappiani et al., 2006).

Candidates are the examples that best correspond to the current preference model. (Viappiani et al., 2006) suggest that the generation of candidates can be performed by a *top-k* query the database. The set of options retrieved $\{o1, \ldots, ok$ $\}$is such that $C(o_1) < C(o_2) \ldots < C(o_k)$ and for any other option o' in the database $C(o_k) < C(o')$.

2.4.2. Generating Suggestions

Suggestions (examples that are used to stimulate the expression of further preferences) can be generated using pareto-optimality and the look-ahead principle (Viappiani et al., 2006). An option is Pareto-optimal if and only if another option does not dominate it (Viappiani et al., 2006). Pareto-optimality applies to any preference model as long as the combination function is dominance-preserving. The look-ahead principle states that suggestions are not supposed to be optimal in the current preference model but have a high likelihood of becoming Pareto-optimal when an additional preference is added, (Viappiani et al., 2006).

Model-based suggestions strategies, strategies that specifically choose example to stimulate the expression of additional preferences based on the current preference model, are used to generate suggestions (Viappiani et al., 2006). They attempt to estimate the chance that a dominated option has to be come Pareto-optimal. To become optimal, argue (Viappiani et al., 2006), an option has to be strictly better than any dominating option with respect to the new preference model.

The notions of dominating set and equal set require consideration at this point. A dominating set of an option o on a set of preferences R is the set of all options that dominate o: $O_R^{>}(o) = \{o' \in O : o' >_R o\}$. A equal set of an option o on a set of preferences R is the set of all options that are equally preferred to o: $O_R^{=}(o) = \{o' \in O : o' \sim_R o\}$, (Viappiani et al., 2006). When a user expresses an additional preference r_i , an option o becomes Pareto-optimal on $R \cup r_i$ if and only if option o is better than all options that dominate it on R and not worse with respect to $r_i$than all options that are equally preferred on R, (Viappiani et al., 2006).

3. The Recommender System

In order to make recommendations, the proposed system models rental properties using five key attributes that are considered by users who search for houses for rent online. These attributes include the type of the property, the location of the property, the rent, the number of bedrooms, and the number of bathrooms.

3.1 System Requirements

The main functional requirements of the recommender system for rental properties, as proposed in this paper, include the following:

i. Ability to perform a preference-based search (PBS)

The system displays a search panel in which users can perform preference-based search by entering attributes of rental properties they consider relevant and indicating how important those attributes are to their search.

ii. Ability to display search results

Once a user has specified her preferences and clicked on the search button, the system displays a set of rental properties that best match her preferences.

iii. Ability to select a preferred rental property

Once, a prospective tenant has located a preferred rental property; the system provides him with the functionality to select this property.

iv. Ability to add rental property to the system, edit them and remove them

The proposed system provides a staff member with the functionality to add rental properties to the database, edit their details, and remove them from the database.

3.2 Use case Diagram

This system has two major types of users: a staff member of a property management firm (staff), and prospective tenants (users). A staff member performs three major actions that can each be described in a use case. These adding a rental property to the database, editing details of a rental property that is already in the database, and deleting a rental property from the database. The user perform three main actions as well. They include searching for properties, viewing properties and selecting a desired property. The three action can be summarised as locating properties. The actions performed by regular users can also be performed by a staff member.

Figure 2 presents the use case diagram derived from the above-mentioned use cases.

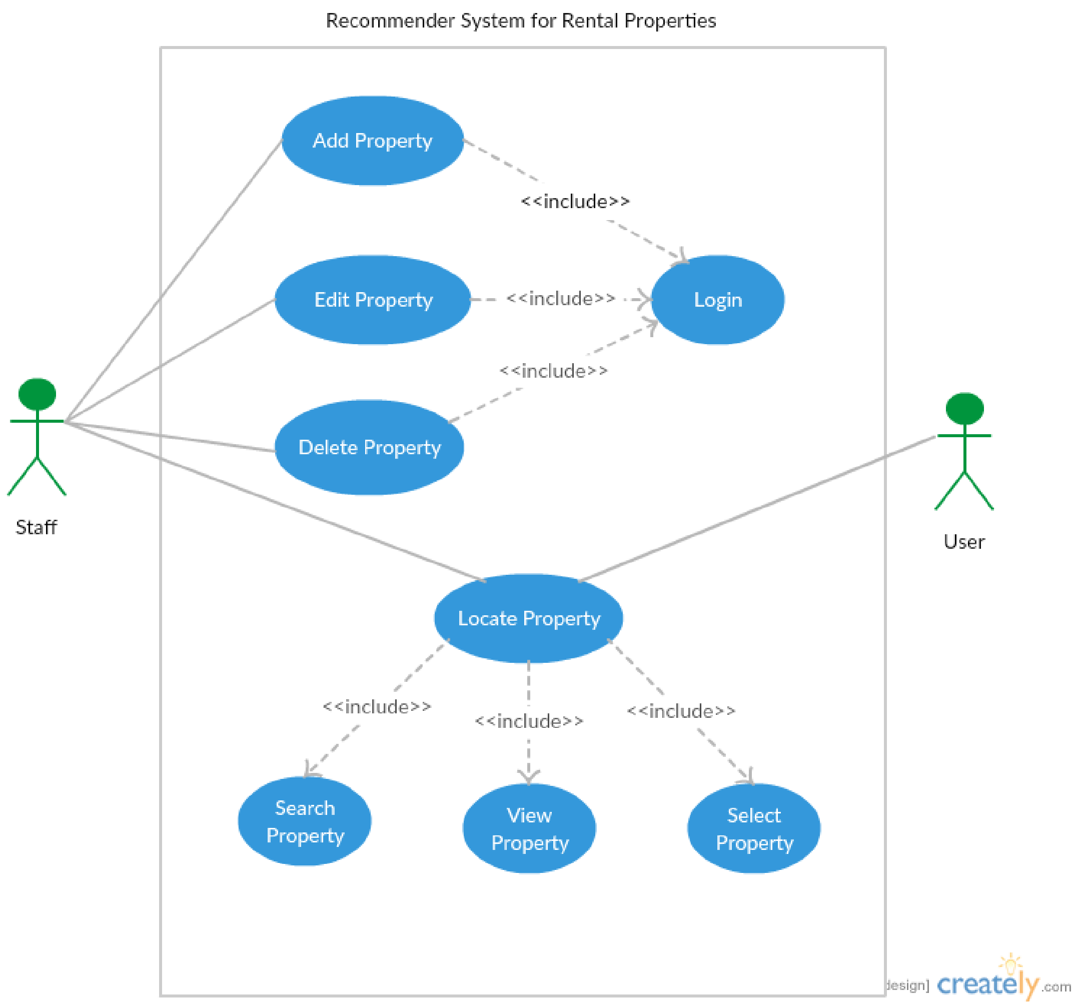

Figure 2: Use case Diagram for the Recommender System

3.3 Classes

The Staff class has three attributes and five methods. The attributes are name, email, and password. The methods are 'register', 'login', 'add new property', 'edit existing property' and 'delete existing property'. These attributes and methods adequately satisfy the requirements of this class, for the purpose of this study.

The user class has no attributes but has four methods. These are 'express preferences', 'search for property, 'view property, and 'select property'. For the purpose of the research, these methods adequately captured the behaviour of the prospective tenant is the system.

The property class has the following primary attributes: type, location, rent, number of bedrooms and number of bathrooms. In addition to these, it has derived attributes including the cost function with respect to each primary attribute, an aggregate cost function, a set of dominating properties with respect to the current preference model, and a probability of escaping these dominance relations.

The preference model class aggregates the preferences expressed by a prospective tenant on properties. Its attributes including the value and importance of the preferences expressed on each attribute of a rental property (type, location, rent, number of bedrooms and number of bathrooms)

3.4 Entity Relationship Diagram

Entity relationship diagram shows the information created, stored and used by a system, and has three main components namely entities, attributes and relationships (Dennis et al, 2012). The following ERD comprises of four entities: Staff, Property, User, and Preference Model. The entity 'Staff' has the following attributes: staff_id, name, email and password. The entity 'Property' has the following attributes: prop_id, type, location, rent, bedrooms, and bathrooms. The entity 'User' does not require any attributes for the purpose of this system. The entity 'Preference Model' has the following attributes: pref_id, type_value, location_value, rent_value, bedrooms_value, bathrooms_value, type_weight, location_weight, rent_weight, bedrooms_weight, and bathrooms_weight. The relationships between these entities are as follows: a user has one preference model and a preference model belongs to one user; a user locates zero to many properties and a property can be located by zero or many users; a staff member manages zero to many properties and a property is managed by one staff member. This is presented in Figure3.

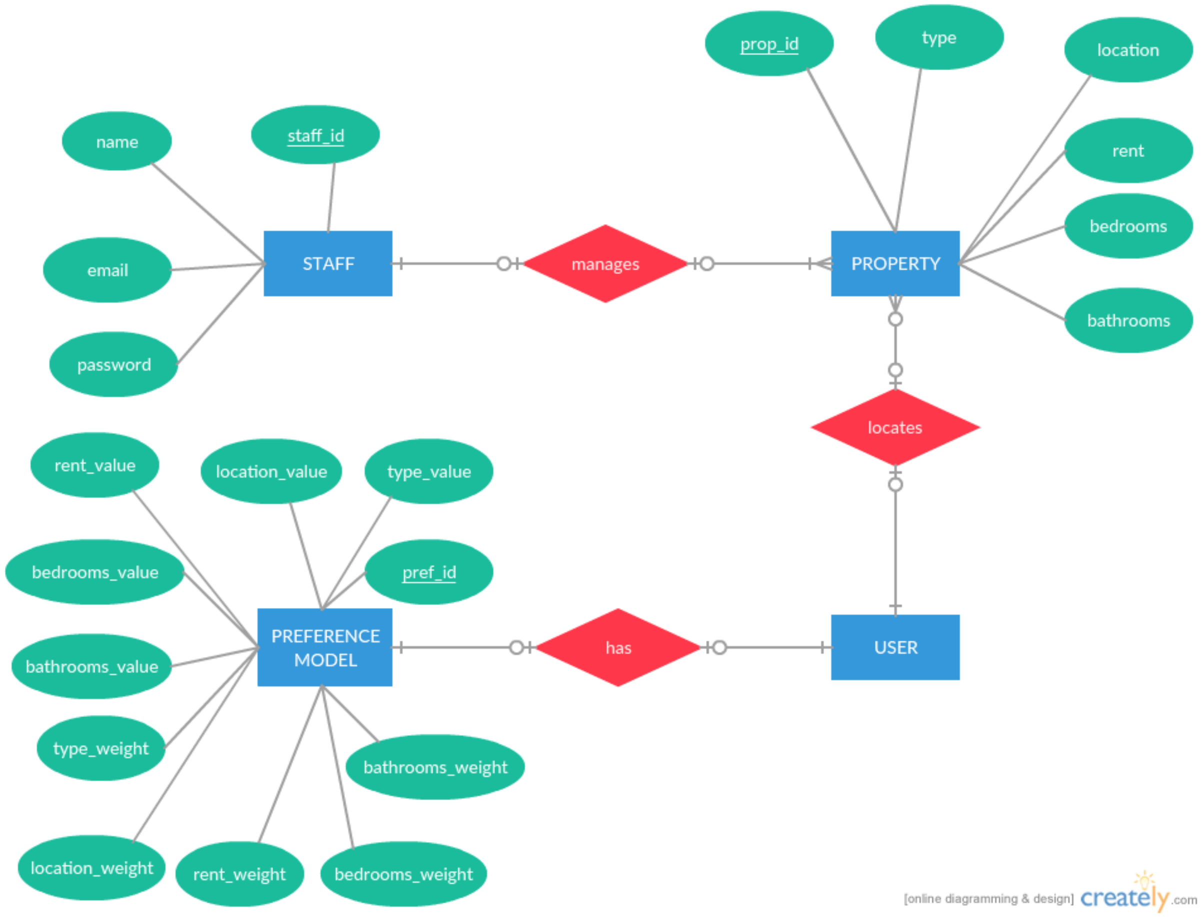

Figure 3: Entity Relationship Diagram for the Recommender System

3.5 Organisation of the Codebase

The code that implements the core algorithms used in this prototype is written in the ruby programming language. Ruby on Rails, being MVC (Model-View-Controller) framework, these algorithms are housed in models (that map to classes defined in Chapter Four). In particular, these algorithms that implement the logic of generating results for the search defined in the 'property' model. The 'Views' are responsible for displaying user interfaces, while controllers are responsible for managing the flow of data between the database, the models and the views.

3.6 The Database

This study employed the SQLite3 database in the development environment as this this the default database for the development environment. When the application was deployed to the production environment, the database used was PostgreSQL, as it is the default database for, Heroku, the cloud-based platform used in this study to deploy the application. Figure 5.10 shows a snapshot of the 'properties' table in the database in the development environment.

3.7 Search Results

The following screenshot presents the page that displays search results. The upper panel, shows the preferences expressed by the user that led to the results. the Middle panel presents the rental properties that best match the user's preferences. The lower panel, displays the suggestions that stimulates the user to provide additional preferences to improve search results.

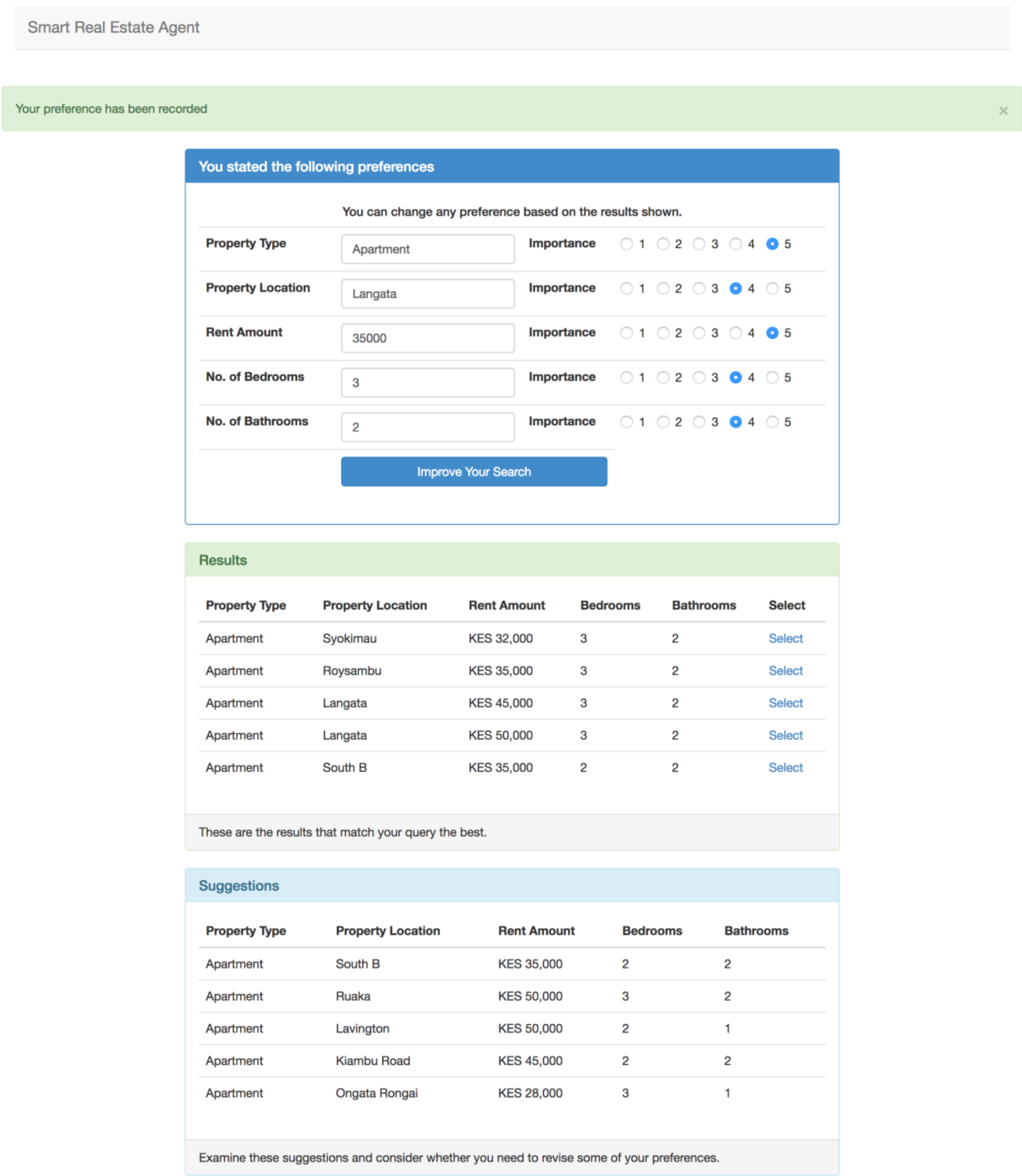

Figure 4: A Sample Results Page for the Recommender System

4. Conclusion

The use of recommendation technology can greatly benefit online users who are searching for rental properties. It has the potential of improving search results and reduce the frustrations facing prospective tenants in search of a home. The increased access to Internet connectivity world over mainly through mobile handsets makes it even more compelling for business organisations, including those who advertise rental properties online, to invest more resources in online platforms that connect consumers to products and services they are looking for. This paper presents an effort in that direction.

References

Burke, R. D., 2002: Hybrid Recommender Systems: Survey and Experiments: User modeling and User-Adapted Interaction. *The Journal of Personalization Research*, 12(4), 331-370

Chen, L., & Pu, P., 2007: Preference-based Organisation Interfaces: Aiding User Critiques in Recommender Systems. In C. Conati, K. McCoy, & G. Paliouras (Eds.), *User Modeling 2007: 11th International Conference* (pp. 77-86), Springer Berlin Heidelberg

Chen, L., & Pu, P., 2012: Critiquing-based recommenders: survey and emerging trends. *User Modeling and User-Adapted Interaction, 22*(1), 125-150

Dennis , A., Wixom, B., & Roth, R., 2012: *System Analysis and Design,* John Wiley & Sons, Inc.

Ekstrand, M., Riedl, T., & Konstan, J., 2010: Collaborative Filtering Recommender Systems, *Foundations and Trends in Human-Computer Interaction, 4*(2), 81-173

Felfernig, A., Friedrich, G., Jannach, D., & Zanker, M., 2011: Developping Constraint-based Recommenders. In F. Ricci, L. Rokach, B. Shapira, & P. B. Kantor, *Recommender Systems Handbook,* Springer

Kenya Bankers Association, 2017: *Residential House Prices Stabilises at Demand softens* [Online], Available at http://www.kba.co.ke/downloads/KBA-HPI%20issue%2010.pdf, [Accessed at 20 June 2017]

Viappiani, P., & Faltings, B., 2006: *Design and Implementation of Preference-based Search,* Ecole Polytechnique Fédérale de Lausanne, Artificial Intelligence Lab, Lausanne

The World Bank, 2011: *Developing Kenya's Mortgage Market* [Online], Available at http://siteresources.worldbank.org/FINANCIALSECTOR/Resources/Kenya_Expanding_Mortgage_Finance_May_2011.pdf [Accesses at 21 June 2017]

The Standard, 2014: *Landlords in Kenya now shunning property agents* [Online]. Available at http://www.standardmedia.co.ke/sports/article/2000138202/landlords-in-kenya-now-shunning-property-agents [Accessed at 11 March 2016]

Kenya Bankers Association, 2015: *The State of Urban Home Ownership in Kenya: A Survey*

Ricci, F., Rokach, L., Shapira, B., & Kantor, P., 2011: Introduction to Recommender System Handbook. In *Recommender System Handbook* (pp. 1-35), Springer

Pu, P., Chen, L., & Kumar, P., 2008: Evaluating product search and recommender systems for E-commerce environments, *Electronic Commerce Research, 8*(1), 1-27

Yuan, X., Lee, J.-H., Kim, S.-J., & Kim, Y.-H., 2013: Toward a user-oriented recommendation system for real estate websites, *Information Systems, 38*(2), 231-243

Viappiani, P., Faltings, B., & Pu, P., 2006: Preference-based Search using Example-Critiquing with Suggestions, *Journal of Artificial Intelligence Research*, 27, 465-503

Ministry of Housing, 2004: *National Housing Policy for Kenya* [Online], Available at www.mintoreal.com/policy-documents/kenya-national-housing-policy?download=8:kenya-national-housing-policy, [Accessed at 20 June 2017]

JEL Classification: L80, R30