# LOCUS: A Mobile Tourism Application and Recommender System for Personalized Places and Activities

Duaa AlSaeed
Information Technology Department, College of Computer and Information Sciences, King Saud University, 11451, Riyadh, Saudi Arabia
E-mail: dalsaeed@ksu.edu.sa

**Keywords:** recommender systems, mobile applications, item-item collaborative filtering, user-user collaborative filtering, tourism

**Received:** August 23, 2022

*The tourism industry is all around keeping tourists happy, occupied and equipped with the things they need during their time away from home. On the other hand, mobile technologies have a considerable impact on user experience, particularly in the tourist and entertainment areas. This paper presents a tourist and entertainment-related mobile application. It utilizes a personalized experience approach and seeks to provide good user experiences by making it adaptable to their unique interests while considering many criteria such as the user's gender, age, location, and other characteristics. The system will propose locations to visit or activities to do in any city to the user. As the user continues to use the application, the suggestions offered will constantly be improved; it will learn more about the user's preferences by recording the user's past and what they enjoy. The application implements and integrates two types of recommender systems, the item-item collaborative filtering algorithm and the user-user collaborative filtering algorithm. The user acceptance testing was conducted on 10 users from a variety of backgrounds and ages. Each participant has performed a set of 17 asks that covers the functionality of the application. Effectiveness results showed that about 70% of the tasks were completed without errors by all participants. And the tasks that were completed with some errors had an average of errors ranges from (0 - 0.4) which is a promising result when compared to the normal average number of errors per which is 0.7. Regarding the efficiency, results show that the longest completion time was in 3 tasks (register task, log-in, and edit profile) which is expected since they require the entry of detailed information. On the other hand, for the remaining tasks the average completion time was 5.4s which is accepted. User satisfaction was measured through a System Usability Scale (SUS) survey, the achieved score was 87.75 which is higher than the threshold to pass the SUS test which is 68, thus LOCUS has fulfilled the user satisfaction measure.*

*Povzetek: Članek predstavi izvirno mobilno aplikacijo za turizem in zabavo, ki zagotavlja personalizirano uporabniško izkušnjo glede na interese uporabnika, se nenehno uči in izboljšuje.*

## 1 Introduction

The tourism industry is linked to the idea of people travelling to other locations, either domestically or internationally, for leisure, social or business purposes. It is one of the fastest growing global industries [1]. It is closely connected to the hotel industry, the hospitality industry which is a broad category of fields within the service industry that includes food and drink service, event planning, theme parks, and transportation [2]. The tourism industry is based around keeping tourists happy, occupied and equipped with the things they need during their time away from home. On another hand, mobile technologies have recently evolved to significantly influence the user experience especially in the tourism and the entertainment domains [3].

Numerous IT solutions have emerged to enhance the tourism industry. Recommender systems, utilizing personalized recommendations [1], [4], [5], and Natural Language Processing (NLP), which can be used to extract important keywords and phrases or conduct sentiment analysis of tourist reviews and comments [6], are among the different technologies leading to varied solutions. These solutions can enhance the tourism experience and meet the growing demand for personalized and efficient services.

Also, with the increase in usage of social media and social networking applications, companies and business owners are relying more on these platforms to advertise and promote their business and services for the aim of attracting more customers [7].

The rapid growth in the tourism field and the entertainment industry led to a huge demand for technological solutions and has motivated developers to create mobile applications that provide innovative solutions and services that will make entertainment and tourism more pleasing and satisfying for people. One of the areas of innovative solutions is to enhance and facilitate the process of finding and deciding the appropriate places/activity that matches one's needs and preferences.

This paper proposes a mobile application related to the tourism and entertainment fields. The application will use a personalized experience manner and aims to present good user experiences by making it customizable to their specific interests taking into consideration several factors such as the user's gender, age, location, and more. The system will provide the user with recommendations for places to visit or activities to do in any city (including restaurants, cafes, etc.). The suggestions provided will always be enhanced as the user keeps using the application, it will learn more about the user's preferences by tracking the user's history and what they like.

In the context of the current tourism industry, there is a clear need for a reference framework for mobile applications that focus on personalized recommendations. Without such a framework, businesses may struggle to keep up with the evolving needs of tourists and may not be able to offer the high-quality experiences that are now expected in the industry. By providing a reference framework for tourism-related mobile applications, our proposed solution can help businesses stay ahead of the curve and improve their overall user satisfaction, retention, and revenue.

Furthermore, in the Saudi Arabian tourism industry, there are no current attempts to develop a specialized tourism application with recommender systems. This further emphasizes the need for such applications in the local tourism sector, especially given the increasing demand for personalized experiences among tourists.

The rest of the paper is organized as follows. Section 2 presents a brief background of recommender systems. Section 3 introduces the work that has been done in the field of applications using recommender systems. Section 4 we present our proposed solution LOCUS mobile application. Section 5 presents our evaluation results and the discussion. In section VI. We conclude the output of this study.

# 2 Recommender systems

"Recommendation system" refers to a system that is able to predict the future preference of a group of items for the user, so it aims to suggest relevant items to users and tries to make predictions on user preferences and make recommendations that should interest the users [2], [3]. Recommendation systems are present in most successful internet companies such as Google, Netflix and YouTube [7]. The reason is that users expect targeted marketing, so providing the same notification or the same offers to everyone is no longer working. Recommendation systems are used in a variety of fields and examples of their uses are: the generators for videos and music playlists used in YouTube and Spotify, suggestions for purchasing products that the user may prefer in Amazon, or suggestions for reading content of user interests such as the recommendation system in Twitter [7].

Lately, different approaches have been developed to build recommendation systems, which can use either collaborative filtering, content-based filtering, or hybrid. Figure 2.2 illustrates the different approaches of recommender systems.

Some of the most well-known websites like Amazon, YouTube, and Netflix use one or more (hybrid) of these recommendation techniques [2], [7].

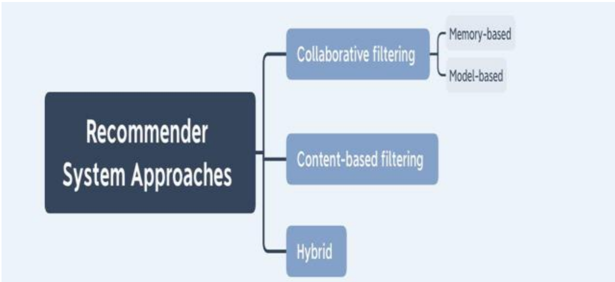


Figure 1: Recommender system approaches.

## 2.1 Collaborative filtering

This filtering methodology is usually based on collecting and analyzing information on user's behaviors, their activities or preferences and anticipating what they will like depends on the similarity with other users. Collaborative filtering assumes that individuals who concurred in the past will concur later on, and that they will like similar sorts of items as they preferred previously. Further, collaborative filtering algorithms can be categorized into two types: memory-based and model-based [2].

### a) Memory-based algorithm.

Another name of the algorithms of memory-based is lazy recommendation algorithms (similar to the k-nearest neighbor method) [2]. There are two different memory-based algorithms:

- User-user collaborative filtering: it finds similar users and recommends items based on items that other similar users previously preferred.
- Item-item collaborative filtering recommends

items similar to those previously favored by the user.

#### b) Model-based algorithm.

The model-based algorithm is considered as a probabilistic approach, so it is about envisioning the future from learning from the past. The models are developed using different algorithms to predict the user rating of unrated items based on the user's previous ratings[2].

## 2.2 Content-based filtering

The concept of content-based filtering is that if a user likes an item, the user will also like a comparable item. The filtering methodology depends on the description of an item and a profile of the user's favored choices, so it compares the content to find items that are similar to those previously perused or preferred by the user and determine the similarity between the items from the keywords that are extracted from the item content descriptions [2], [8].

## 2.3 Hybrid recommendation system

The concept of hybrid recommender systems is combining more than one recommendation technique with one another. Hybrid recommendation systems can be implemented by generating content-based and collaborative predictions separately and then merging them to produce the output. Figure 2 below illustrates the way a hybrid recommender system works.

The accuracy of the hybrid recommendation system is usually higher than the accuracy of any individual ones so combining collaborative and content-based recommendations may be more effective and powerful [2].

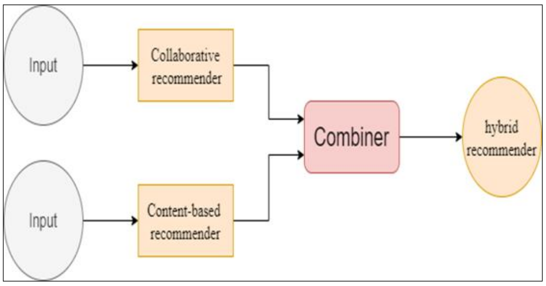


Figure 2: Illustration of hybrid recommender

# 3 Related work

Several studies have proposed recommender systems (RS) in the tourism domain. Where its one of the ways to deal with deal with the information overload problem. This section contains a review of recent studies that used a mobile recommender system.

Soha et al. (2016) [8] proposed a context-aware recommender system, which recommends places to users based on the current weather, the time of the day, and the user's mood. Places are suggested based on what other users have visited in the similar context conditions. proposed system puts rates for each place in each context for each user. The rates of a location are calculated using the Genetic algorithm, which is based on the Gamma function.

Li et al. (2010) [9] propose a location-aware recommender system for tourism mobile commerce service platform. They integrate GPS into the recommender system to create a location-aware recommender system. The location-aware recommendation system can recommend attractions to the customer with the customer's sensitivity to location and the rating of its attractions.

Abu-Issa et al. (2017) [10] implemented a proactive, multitype, and context-aware recommender application in the environment of Internet of Things (IoT) for smart cities. The recommendation system has the ability to suggest multi-types of services (such as Restaurant, Attraction... etc) and proactively pushes explicit query suggestions to users. The application was developed on Android platform and tested by 50 users. The results shows that the application reach 91.2 % accuracy.

Missaoui et al. (2019) [4] proposed a mobile recommender system for tourism and travel related services called LOOKER. It was developed for the Android platform. It considers basic contextual information, such as location and time. Also, its implements based on content-based filtering (CBF) strategy to make personalized suggestions based on user-related tourism-based user-generated content (UGC) spread across social media. User studies have been conducted to evaluate the usability and the usefulness of the proposed application. A first user study was conducted using two popular questionnaires (SUS and CSUQ) to test the usability of the proposed application with a group of users in four major Tunisian cities. A second user study allowed for a quantitative evaluation of the effectiveness of the proposed system's recommendations. The positive results achieved show the potential of LOOKER. Where, the LOOKER's usability and the recommendations made have been judged to be satisfactory by users.

In conclusion, the reviewed studies have explored the use of recommender systems in the tourism industry to improve the user experience of mobile applications. Recommender systems have been proposed as a way to address the problem of information overload and provide personalized recommendations to users based on their location, context, preferences, and other factors. These systems have shown promise in improving user satisfaction and retention, as well as increasing revenue for businesses. While there is a body of literature on the topic, there is still room for innovation and unique approaches to design and implementation of recommender systems for tourism-related mobile applications.

We also believe that it is important to highlight the need for the development of a personalized and context-aware mobile application for tourism in Saudi Arabia, which could potentially address the unique needs and preferences of tourists visiting the country. Up to our knowledge, there is a lack of existing tourism-related mobile applications in Saudi Arabia. This further emphasizes the importance of developing a special tourism mobile application with a recommender system tailored to the country's specific tourism industry.

# 4 Proposed LOCUS mobile application

The rapid growth of the entertainment field in the world and the huge demand on the technological solutions has motivated developers to create a lot of mobile applications. Unfortunately, most applications are using the browsing feature without giving personalized suggestions. Furthermore, these mobile applications require some effort from the user since they depend on filtering things manually to get a list of suggestions.

To enhance and facilitate the process of finding and deciding the appropriate event/activity that matches ones needs and preferences, comes the idea behind developing a mobile application related to the entertainment field that satisfies the users' requirements. The application will use a personalized experience manner and aims to present good user experiences by making it customizable to their specific interests. It will use and integrate recommender system models to provide suggestions for possible activities / based on user's preferences, characteristics, and behavior which will be learned while using the application. The suggestions provided will always be enhanced as the user keeps using the application, it will learn more about the user's preferences by tracking his/her history and what he/she likes. Furthermore, the application will provide information about each activity or place such as pictures, reviews, and location.

The following subsections will discuss the proposed mobile application in detail, covering data sources, architecture, main features, design and development.

## 4.1 Data sources

For the proposed LOCUS mobile application, real-time and real-world data will be used. The data will be collected mainly from APIs such as google places API [11]. Seasonal or temporary events information will be entered manually from trusted media sources, applications, and websites. Then, the collected information will be stored on a server-based database and will be retrieved upon request from the application. Figure 2 below shows how in general APIs are used to collect data.

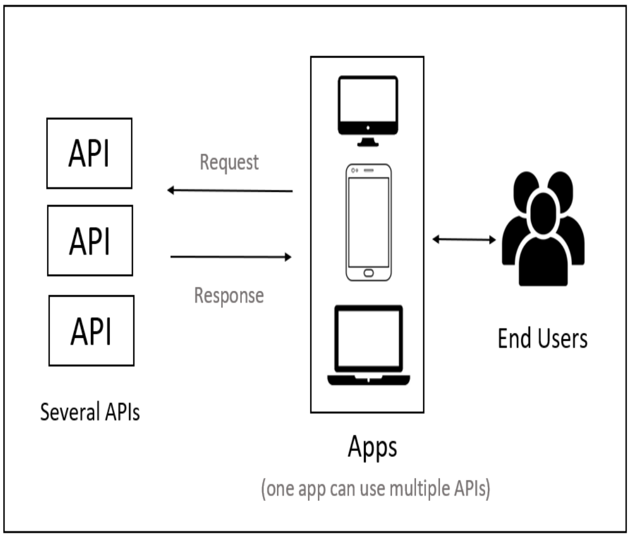


Figure 2: An illustration of how apis are used for collecting data.

## 4.2 System architecture

The LOCUS system is a client-server architecture in order to enable different users from many different locations to access the shared data that our application provides at any time. Client-server architecture is a shared architecture system in which a group of clients' requests services, a network that allows clients to access these services, and a set of servers that provide services. So, we will implement client-server architecture to fulfill requests by responding to them with the requested services as shown in Figure 3:

LOCUS system architecture Figure 3 below

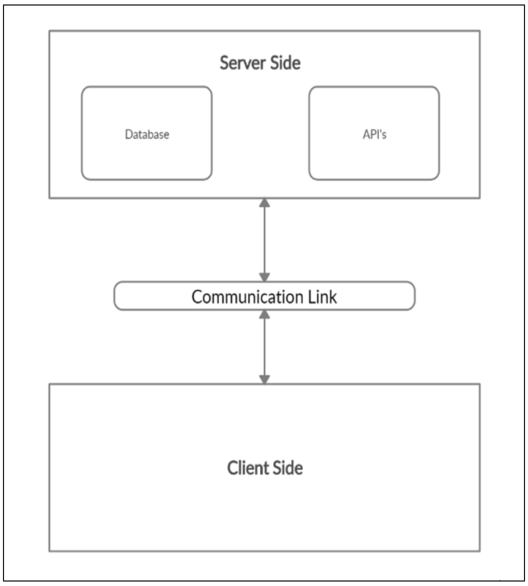


Figure 3: LOCUS system architecture.

On the client side of LOCUS system are the mobile phones that users run the application on, while the server side includes the following:

a) A database to store users' information and manually added activities or seasonal events.

b) The APIs, the application will use multiple APIs to provide information about places, description of each place and its location. Also, the APIs offer some services we want to use such as the nearest places.

## 4.3 User characteristics and roles

LOCUS application targets the users with the age range from teenagers to adulthood, including both genders and who have interests in finding entertainment activities in Riyadh, whether they are citizens or tourists. there are no restrictions on the education level as long as they have enough knowledge about using smartphones and are able to understand the English language.

## 4.4 Main features

It has general features such as: add an activity to the favorite list, view ratings, and view reviews/comments. Also, It has customized features such as search by name, Filter the search, and the content. LOCUS uses collaborative filtering (CF) as a recommender system approach. In Figure 4, a use-case diagram is presented, which shows the main features provided by the system and are represented as actions that can be performed by the user.

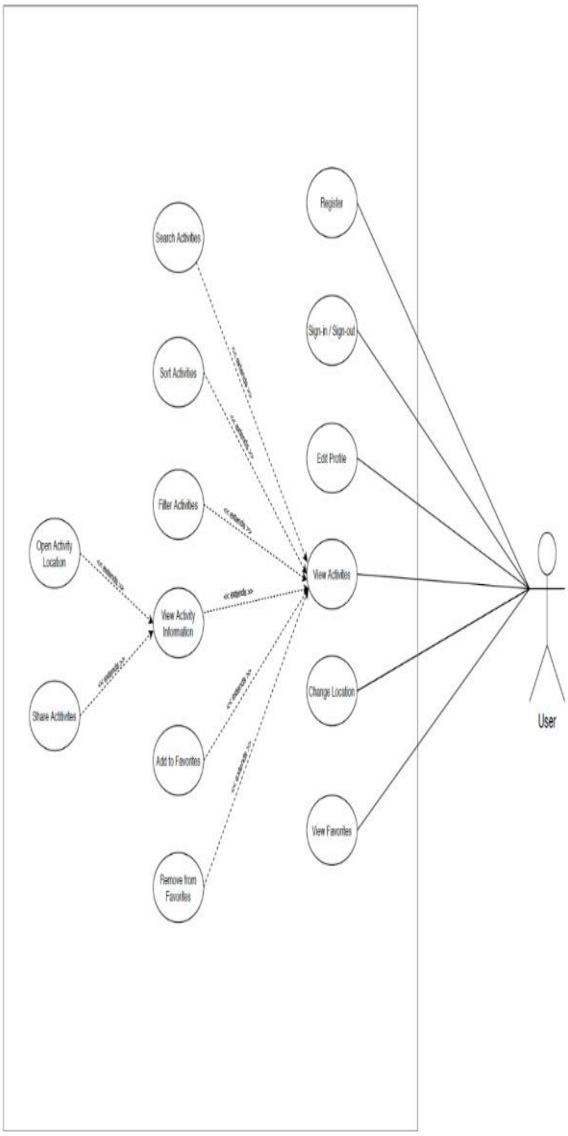


Figure 4: Use case diagram for LOCUS mobile application.

## 4.5 Implementation

LOCUS application is developed to run on iOS [12] and Android [13] platforms, using React Native framework [14] which uses JavaScript language. In addition, the Firebase database [15] was used to store the data. For the user recommendation functions such as view recommended activities based on user interests or similar users, a module was developed using Python programming language to compute the recommendation algorithms then the Flask Library [16] provided by Python which transforms the code into an API then Heroku [17] was used to deploy it to make the API accessible for LOCUS platform. After that, the APIs in our React Native application were used using HTTP request calls.

## 4.6 Algorithms for developed recommenders

As mentioned earlier the proposed system will implement and integrate two types of recommender systems, the item-item collaborative filtering algorithm and the user-user collaborative filtering algorithm. In the following subsection, we will discuss the implemented algorithm for both modules.

### c) Item-Item collaborative filtering algorithm

An item-item collaborative filtering recommendation system builds a model to provide a suggested activity based on the user's interest. The model predicts the items, or in this case activities, that the user might be interested in based on computing the similarities between the activity and the user's interests using the vector similar might be interested in based on computing the similarities between the activity and the user's interests using the vector similarity (cosine) and its result will be a value between 0 and 1, the value closer to 1 represents the similarity, and the value closer to zero represents the opposite(cosine) and its result will be a value between 0 and 1, the value closer to 1 represents the similarity, and the value closer to zero represents the opposite [18].

$$s(a,u) = \sum_{j\in 1} \frac{v_{aj}}{\sqrt{\sum_{k\in I_a} v_{ak}^2}} \frac{v_{aj}}{\sqrt{\sum_{k\in I_a} v_{ak}^2}} \quad (1)$$

$$similarity = \cos(\theta) = \frac{A \cdot B}{\|A\|\|B\|} = \frac{\sum_{i=1}^{n} A_i B_i}{\sqrt{\sum_{i=1}^{n} A_i^2}\sqrt{\sum_{i=1}^{n} B_i^2}}$$

(2)

Where a means the activity vector and b the user vector.

In our application, we give each activity and user a vector representing a number of features (category, key words of activity features such as Wi-Fi, credit card, outside seating, ... Etc.) Those features will be extracted from the activity information and for the user it will be extracted from the user's (interests, viewed activities and favorite list). Our system will compute the cosine vector similarity between the activity and user vectors, then test if it passes a 0.5 threshold to decide if it shall recommend it to the user [18]. Figure 5 shows a flowchart for the developed module for item-item recommender.

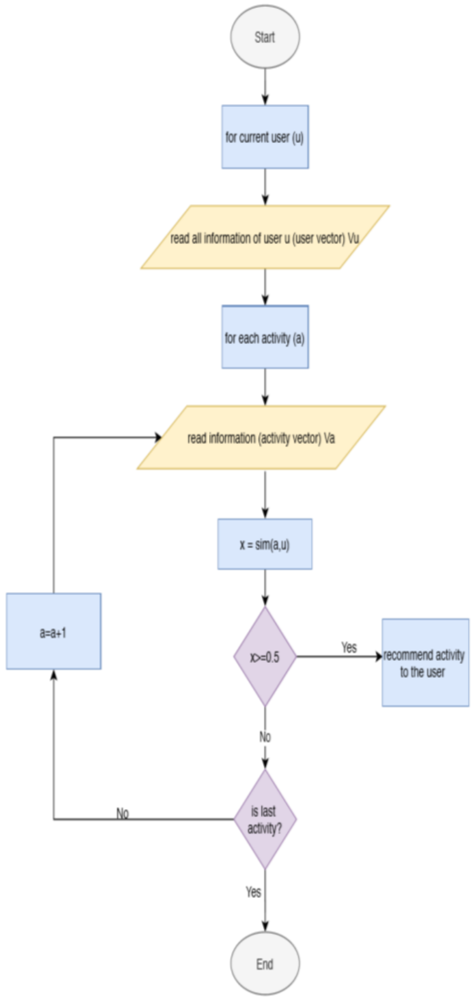


Figure 5: A flowchart for the developed item-item recommender.

### d) User-user collaborative filtering algorithm

A user-user collaborative filtering recommendation system builds a model to find similar users and suggest items (activities) based on items (activities) that similar users previously preferred. In our case first, we will select all the users that have the same gender, city and age group, then we will measure the similarity between those users' preferred activities and the user's preferred activities and we will assume that the preferred activities are user's (interests, viewed activities and favorite list) using the vector similarity (cosine) where a means the users' activities (activity) vector and b the user vector using same technique in the item-item collaborative filtering that we mentioned previously [18]. Figure 7 shows a flowchart for the developed module for user-user recommender.

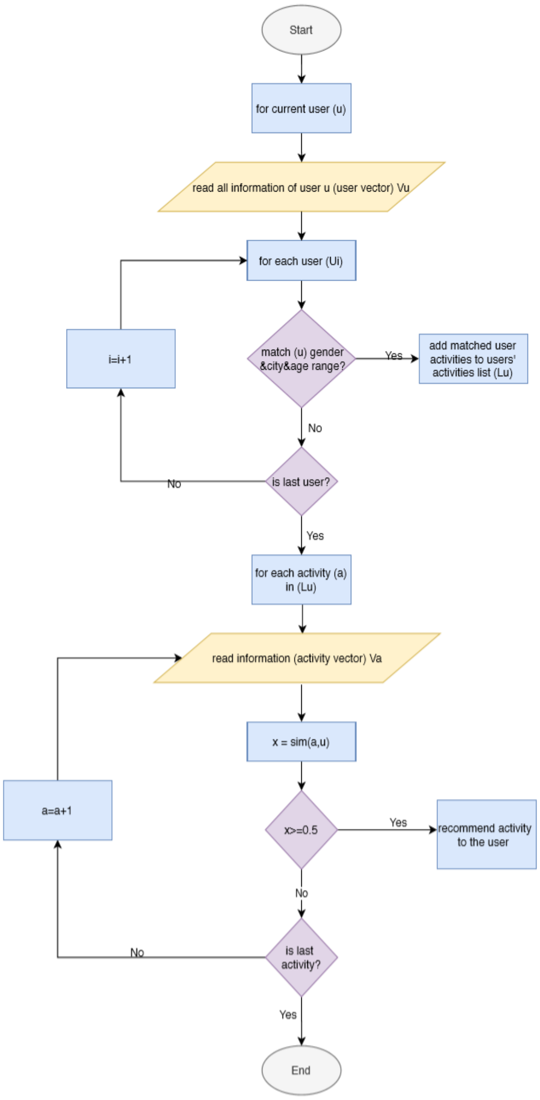


Figure 7: A flowchart for the user-used developed recommender module

## 4.7 Application layout and user interface

While developing LOCUS, maintaining the ease of use with the functionality of the proposed application were a priority. This section provides a detailed description of the application's user interface. The visual presentation of the user interface (UI) elements has a great impact on the user experience of a product\application. For a good user experience, the content components should be well organized to help people navigate easily within a product or interact with it properly. The main screen of LOCUS gives the user access to the main functions of the application. Screen shots of some of the main screens in the application are shown in Figure 6Figure 6 below.

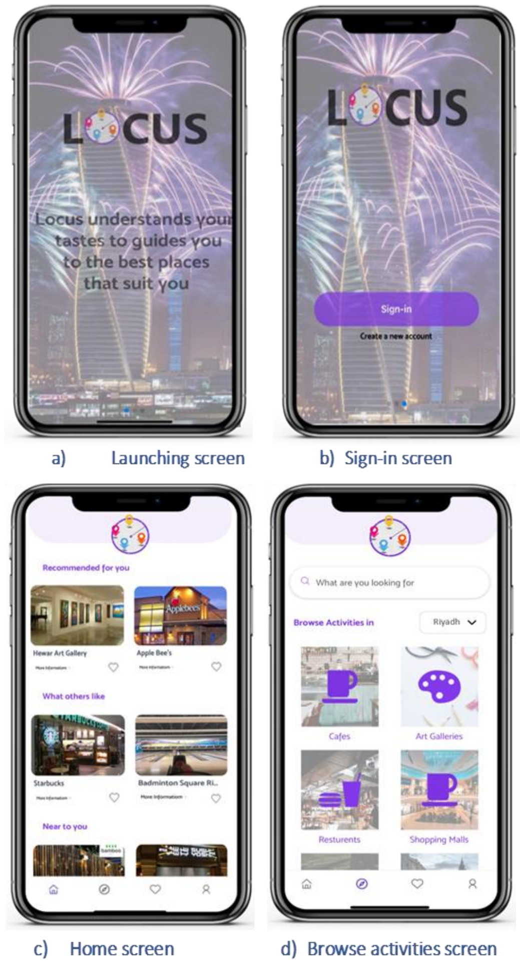


Figure 6: Sample screens of LOCUS mobile application.

# 5 User acceptance testing and discussion

User Acceptance Testing (UAT) is an important step in the software testing process where actual users test the software to assure that functions work as expected and the application can process the tasks in real-world scenarios [19]. In addition, a user acceptance study serves as a proof-of-concept that demonstrates the applicability and effectiveness of the proposed reference framework.

In this work, the user acceptance testing was conducted on 10 users from a variety of backgrounds and ages. The test measured the system usability in three criteria [20]:

- Effectiveness: counting the number of errors, the participant makes when attempting to complete a task.

- Efficiency: calculating the time that participant takes to successfully complete a task.

- Satisfaction: using System Usability Scale (SUS) form to filled by the user, which includes 10 questions to evaluate user satisfaction.

The following subsections describe the conducted user acceptance testing, including the participants' information, experiment design, and testing results.

## 5.1 Participants

For the user acceptance testing in this study, the participants were a random sample of 10 people covering the characteristics of the targeted users. Their ages are between 20 to 32 with an average of age 24 years. In this study, only nine females and a male volunteered to participate. All participates can understand the basic English language, and capable of using smartphones. Table 1 below shows the demographics of the participants.

Table 1: Participant’s demographics

| Participant number | Age | Gender | Can understand basic English language | Capable of using smart phones |
|---|---|---|---|---|
| **1** | 23 | Female | Yes | Yes |
| **2** | 23 | Female | Yes | Yes |
| **3** | 25 | Female | Yes | Yes |
| **4** | 20 | Male | Yes | Yes |
| **5** | 32 | Female | Yes | Yes |
| **6** | 24 | Female | Yes | Yes |
| **7** | 23 | Female | Yes | Yes |
| **8** | 25 | Female | Yes | Yes |
| **9** | 22 | Female | Yes | Yes |
| **10** | 23 | Female | Yes | Yes |

## 5.2 Experiment design

The user acceptance testing was conducted with the participants. participants tested the application using an iOS iPhone prepared for the testing (Have an internet connection, Locus app downloaded). In addition, different data collection methods were used to collect quantitative and qualitative data. These methods included pre-questionnaires, direct observation during the test sessions, and post-questionnaires.

- Pre-questionnaire: A pre-questionnaire was used to collect data on the participants. Participants were asked about the places they visit often. Also, if they are facing any difficulties finding suitable places with full information about the places. The pre-questionnaire also included a question about what is the features they would like to have in the recommendation system that could help them to determine their destination easily?
- User acceptance testing sessions: During the UAT sessions, several types of data were collected while observing the participants performing a total of 17 tasks. A timer was used to record the completion time for each task, also we took notes of their mistakes and difficulties faced by the participants during performing the tasks and their feedback.
- Post-questionnaire: After performing the tasks for the LOCUS application, we requested the participants to fill in a postquestionnaire that includes questions to assess their satisfaction. As well as gathering their feedback and suggestions for improvement.

## 5.3 Testing results and discussion

Each participant has performed a set of tasks that covers the functionality of the application. For each participant, quantitative measures were recorded, this includes: the number of errors and completion time for each task. The list of the 17 tasks performed by participants and a summary of users’ performance results are shown in Table 2. Next, the results for effectiveness, efficiency and user satisfaction will be discussed.

- **Effectiveness**: To measure the effectiveness (task completion), we counted the number of participants who completed the tasks without errors and the number of participants who completed the tasks with errors, and the average number of errors made by those participants for each task. Around 70% of the tasks were completed without errors by all participants. And the tasks that were completed with some errors had an average of errors ranges from (0 - 0.4), According to MeasuringU Research Centre [21], the normal average number of errors per task is 0.7,the achieved results are promising since the highest average number of errors for a task is 0.4 and the average number of errors for all tasks is 0.059 and all are below 0.7 which is a good indicator of the effectiveness of the system. Figure 8 shows a summary of number of errors results.

- **Efficiency**: In order to evaluate the efficiency was measures by the time taken to complete a task by participants. The completion time for each task for each participant was recorded and analyzed, the maximum, minimum, and average time per task were computed. Figure 9 shows a summary of completion time results. The results show that the longest time spent was in 3 tasks register task, log-in, and edit profile which is expected since these tasks require the user to enter detailed information, looking into the remaining tasks the average completion time was 5.4s which is accepted.

- **User satisfaction**: To evaluate the user satisfaction of LOCUS app all 10 participants have filled the System Usability Scale (SUS) survey which has 10 questions and results were collected and analyzed. Participants ranked each of the 10 questions from 1 to 5, based on their level of agreement. To calculate the user satisfaction score using SUS we performed the following steps:
    - For each of the odd numbered questions, subtract 1 from the score.
    - For each of the even numbered questions, subtract their value from 5.
    - Add up the total score. Then multiply the value by 2.5.

The SUS score for LOCUS was found to be 87.75, and the threshold to pass the SUS test is 68 according to MeasuringU Research Centre [22] which means that LOCUS app has fulfilled the satisfaction measure.

below shows the survey's measuring scale and results. After completing the survey, we asked them about their opinions and suggestions regarding Locus application interface, most of them (87%) agreed on the necessary of the following:

- Developing an Arabic version interface.
- Separating the filter and sort buttons for better accessibility.
- Adding more user interaction such as adding reviews and ratings.

Those suggestions will be considered in the development of the next version of LOCUS as a future work.

Table 2: Summary of user testing results

| **Measure\ Task** | **Effectiveness** | | | **Efficiency \| Task completion time (seconds)** | | |
|---|---|---|---|---|---|---|
| | # of users complete the task without errors | # of users complete the task with errors | Average of errors | Max. | Min. | Average |
| **Register** | 6 | 4 | 0.4 | 120.0 | 30.0 | 59.7 |
| **Sign-in** | 8 | 2 | 0.2 | 59.6 | 16.7 | 34.9 |
| **Edit profile** | 9 | 1 | 0.1 | 38.0 | 6.2 | 19.3 |
| **View categories** | 10 | 0 | 0 | 17.4 | 3.0 | 7.7 |
| **View activities based on category** | 10 | 0 | 0 | 7.4 | 1.0 | 4.3 |
| **Filter activities** | 9 | 1 | 0.1 | 27.4 | 5.6 | 14.3 |
| **Search for activities** | 10 | 0 | 0 | 15.6 | 3.1 | 9.4 |
| **Sort activities** | 8 | 2 | 0.2 | 27.8 | 5.0 | 9.2 |
| **View recommended activities** | 10 | 0 | 0 | 11.1 | 2.6 | 6.9 |
| **View nearby activities** | 10 | 0 | 0 | 11.3 | 2.1 | 4.5 |
| **View activity details** | 10 | 0 | 0 | 8.3 | 2.9 | 5.6 |
| **Share activity** | 10 | 0 | 0 | 14.5 | 2.3 | 7.4 |
| **Open in maps** | 10 | 0 | 0 | 20.0 | 1.6 | 8.3 |

| | | | | | | |
|---|---|---|---|---|---|---|
| **Add to favorite** | 10 | 0 | 0 | 6.1 | 2.2 | 4.2 |
| **View favorites** | 10 | 0 | 0 | 5.10 | 2.2 | 3.8 |
| **Remove from favorite** | 10 | 0 | 0 | 5.5 | 1.2 | 3.6 |
| **Sign-out** | 10 | 0 | 0 | 10.3 | 1.1 | 4.0 |

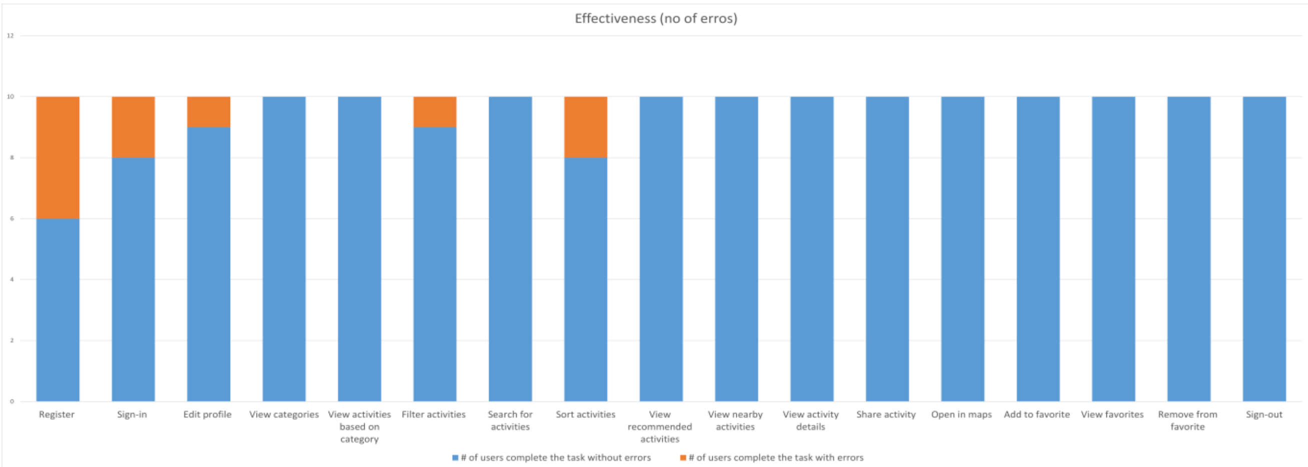


Figure 8: Summary of user's performance (no of errors) – effectiveness results.

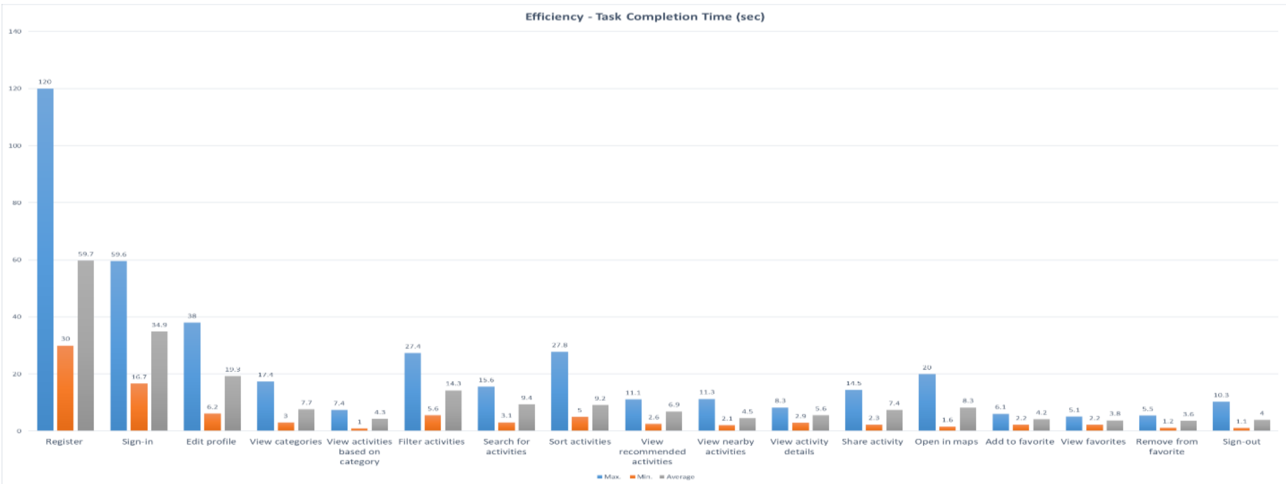


Figure 9: Summary of user's performance (completion time) - efficiency results.

Table 3: User satisfaction survey results

| **Questions \ Measure scales** | # Of users who | | | | |
|---|---|---|---|---|---|
| | Strongly agree. (5) | Agree (4) | Neutral (3) | Disagree (2) | Strongly disagree. (1) |
| **I think that I would like to use LOCUS App frequently** | 6 (60%) | 3 (30%) | 1 (10%) | 0 | 0 |
| **I found LOCUS App unnecessarily complex** | 0 | 0 | 0 | 6 (60%) | 4 (40%) |
| **I think LOCUS App was easy to use** | 6 (60%) | 3 (30%) | 1 (10%) | 0 | 0 |
| **I think that I would need the support of a technical person to be able to use LOCUS App** | 0 | 0 | 0 | 3 (30%) | 7 (70%) |
| **I found the various functions in LOCUS App were well integrated** | 4 (40%) | 4 (40%) | 2 (20%) | 0 | 0 |
| **I think there was too much inconsistency in LOCUS App** | 0 | 0 | 1 (10%) | 4 (40%) | 5 |
| **I would imagine that most people would learn to use LOCUS App very quickly** | 8 (80%) | 1 (10%) | 1 (10%) | 0 | 0 |
| **I found LOCUS App very cumbersome to use** | 0 | 0 | 0 | 2 (20%) | 8 (80%) |
| **I felt very confident using LOCUS App** | 5 (50%) | 4 (40%) | 1 (10%) | 0 | 0 |
| **I needed to learn a lot of things before I could get going with LOCUS App** | 0 | 0 | 1 (10%) | 2 (20%) | 7 (70%) |
| **I found LOCUS App unnecessarily complex** | 0 | 0 | 0 | 6 (60%) | 4 (40%) |
| **I think LOCUS App was easy to use** | 6 (60%) | 3 (30%) | 1 (10%) | 0 | 0 |
| **I think that I would need the support of a technical person to be able to use LOCUS App** | 0 | 0 | 0 | 3 (30%) | 7 (70%) |
| **I found the various functions in LOCUS App were well integrated** | 4 (40%) | 4 (40%) | 2 (20%) | 0 | 0 |
| **I think there was too much inconsistency in LOCUS App** | 0 | 0 | 1 (10%) | 4 (40%) | 5 |
| **I would imagine that most people would learn to use LOCUS App very quickly** | 8 (80%) | 1 (10%) | 1 (10%) | 0 | 0 |
| **I found LOCUS App very cumbersome to use** | 0 | 0 | 0 | 2 (20%) | 8 (80%) |
| **I felt very confident using LOCUS App** | 5 (50%) | 4 (40%) | 1 (10%) | 0 | 0 |
| **I needed to learn a lot of things before I could get going with LOCUS App** | 0 | 0 | 1 (10%) | 2 (20%) | 7 (70%) |

# 6 Conclusion and future work

The continuous growth in the tourism industry in addition to the high impact of mobile technologies on user experience in many fields and in particular the tourist and entertainment areas have led to high demand on mobile applications that provide a personalized experience and services to the tourist.

Derived by this demand, LOCUS was developed as a mobile application that targets tourists and aims to suggest for them activities, places and events that suit their interests and are based on the user's characteristics, behavior. It implements and integrates two types of recommender systems, the item-item collaborative filtering algorithm and the user-user collaborative filtering algorithm. It proposes places and activities to the user and those suggestions will constantly be improved as the user keeps using the application.

Through user acceptance testing with 10 participants, we found that the system achieved promising results in terms of effectiveness, efficiency, and satisfaction. We plan to expand the application's capabilities by adding multilingual support and new features such as rating, writing reviews, and crowdsourcing to allow users to add places and activities. Overall, the development of LOCUS shows the potential for mobile applications to enhance the tourist experience, and we hope to continue exploring ways to improve and expand its functionality in the future. This could include, for example, exploring the potential benefits of incorporating user-generated content and social media features, as well as investigating ways to integrate the application more effectively with other tourism-related technologies and platforms.

# References


[1] B. Cvetković *et al.*, "e-Turist: An Intelligent Personalised Trip Guide," *Informatica*, vol. 40, no. 4, Art. no. 4, Dec. 2016, Accessed: Apr. 20, 2023. [Online]. Available: https://www.informatica.si/index.php/informatica/article/view/1385

[2] A. H. Nabizadeh, N. Rafsanjani, N. Salim, A. Rezaei Aghdam, and K. Fard, "Recommendation Systems: a review," May 2013.

[3] F. O. Isinkaye, Y. O. Folajimi, and B. A. Ojokoh, "Recommendation systems: Principles, methods and evaluation," *Egyptian Informatics Journal*, vol. 16, no. 3, pp. 261–273, Nov. 2015, doi: 10.1016/j.eij.2015.06.005.

[4] S. Missaoui, F. Kassem, M. Viviani, A. Agostini, R. Faiz, and G. Pasi, "LOOKER: a mobile, personalized recommender system in the tourism domain based on social media user-generated content," *Personal & Ubiquitous Computing*, vol. 23, no. 2, pp. 181–197, Apr. 2019, doi: 10.1007/s00779-018-01194-w.

[5] A. Tavčar, A. Csaba, and E. V. Butila, "Recommender System for Virtual Assistant Supported Museum Tours," *Informatica*, vol. 40, no. 3, Art. no. 3, Dec. 2016, Accessed: Apr. 29, 2023. [Online]. Available: https://www.informatica.si/index.php/informatica/article/view/1433

[6] H. N. L. Huy, H. H. Minh, T. N. Van, and H. N. Van, "Keyphrase Extraction Model: A New Design and Application on Tourism Information," *Informatica*, vol. 45, no. 4, Art. no. 4, Dec. 2021, Accessed: Apr. 29, 2023. [Online]. Available: https://www.informatica.si/index.php/informatica/article/view/3493

[7] R. Python, "Build a Recommendation Engine with Collaborative Filtering – Real Python." https://realpython.com/build-recommendation-engine-collaborative-filtering/ (accessed Apr. 20, 2023).

[8] S. A. El-Moemen, T. Hassan, and A. A. Sewisy, "A Context-Aware Recommender System for Personalized Places in Mobile Applications," *International Journal of Advanced Computer Science and Applications*, vol. 7, Mar. 2016, doi: 10.14569/IJACSA.2016.070360.

[9] "A location-aware recommender system for Tourism mobile commerce | IEEE Conference Publication | IEEE Xplore." https://ieeexplore.ieee.org/document/5690282 (accessed Apr. 30, 2023).

[10] A. Abu-Issa *et al.*, "A smart city mobile application for multitype, proactive, and context-aware recommender system," in *2017 International Conference on Engineering and Technology (ICET)*, Aug. 2017, pp. 1–5. doi: 10.1109/ICEngTechnol.2017.8308181.

[11] "Overview | Places API | Google Developers." https://developers.google.com/maps/documentation/places/web-service/overview (accessed Apr. 20, 2023).

[12] "ISO - IT tools for standards development," *ISO*. https://www.iso.org/it-tools-for-standards-development.html (accessed Jun. 20, 2021).

[13] "Android 11 | Android Developers." https://developer.android.com/about/versions/11?gclid=CjwKCAjwzruGBhBAEiwAUqMR8NPJvjC5qcihfp9pWlqwm0SvXyF043xzzhUrYorNA_X9KjXM-q1k0BoCWXMQAvD_BwE&gclsrc=aw.ds (accessed Jun. 20, 2021).

[14] "React Native · Learn once, write anywhere." https://reactnative.dev/ (accessed Apr. 20, 2023).

[15] "Firebase Realtime Database." https://firebase.google.com/docs/database?hl=en (accessed Jun. 20, 2021).

[16] "Welcome to Flask — Flask Documentation (1.1.x)." https://flask.palletsprojects.com/en/1.1.x/ (accessed Apr. 20, 2023).

[17] "Cloud Application Platform | Heroku." https://www.heroku.com/ (accessed Apr. 20, 2023).

[18] F. Cacheda, V. Carneiro, D. Fernández, and V. Formoso, "Comparison of collaborative filtering algorithms: Limitations of current techniques and proposals for scalable, high-performance recommender systems," *ACM Trans. Web*, vol. 5, no. 1, p. 2:1-2:33, Feb. 2011, doi: 10.1145/1921591.1921593.

[19] "What is User Acceptance Testing (UAT Testing)? - Usersnap," *Usersnap blog - the feedback platform*, Feb. 17, 2020. https://usersnap.com/blog/user-acceptance-testing-right/ (accessed Feb. 07, 2021).

[20] "Usability Metrics - A Guide To Quantify The Usability Of Any System - Usability Geek." https://usabilitygeek.com/usability-metrics-a-guide-to-quantify-system-usability/ (accessed Feb. 07, 2021).

[21] "MeasuringU: 10 Benchmarks for User Experience Metrics." https://measuringu.com/ux-benchmarks/ (accessed Feb. 07, 2021).

[22] "MeasuringU: Measuring Usability with the System Usability Scale (SUS)." https://measuringu.com/sus/ (accessed Feb. 07, 2021).