# Implementation

This chapter discusses in further detail the components proposed in the previous chapter. The implementations used a number of applications and libraries, namely, Vuforia [14], Unity Engine [14], and Python 3.7. Vuforia was mainly used to train image and model targets, while Unity is a game engine/editor used in this project to build the augmented reality application architecture, and Python 3.7 was used for image sharpening and for training and testing the recommendation system. This chapter highlights all the necessary details about data handling, feature extraction, user-query recommendation, user interface, and system architecture which was built accordingly. Vuforia was used for augmented reality, mainly because it is the “most popular SDK for developing ARapplications on a wide selection of devices” [14]. Similarly to ARCore and ARKit, Vuforia can be used on multiple devices to recognise images, objects and text.

## Data Handling

### Images

The initial step of the project entailed gathering many images of the first floor of the workplace. Indeed, a good number of images were taken of all the corridors and objects inside them, as well as the doors. Images of the interior of the oﬃce rooms on the ﬁrst ﬂoor were excluded since it is not within the scope of this project to capture them within the augmented reality application. Diﬀerent variations of the same image were captured, ensuring to capture possible diﬀerent lighting and time scenarios. The reason behind this was mainly because GPS signals are weak indoors and Vuforia does not support location-based tracking. Therefore, the Augmented Reality had to be marker-based through image and model targets. The challenge in this case was capturing the same images in diﬀerent scenarios. For example, if one door was open in one image, another image had to be captured where the door was closed thereby training the system in the various settings and contexts.

### Markers

The next step was choosing the markers which Vuforia could use to recognise and overlay the augmentation on. Choosing the best markers is important as it can affect entirely the user’s experience within the augmented reality. Initially, it was thought to use the images of the corridors themselves that were going to be used as markers. However, this meant that for one corridor, one needed to take several images capturing an inﬁnitely amount of variations which could occur in that image. For example, in Figures 4 and 5, one can see several clear sight variations of the same corridor. By ﬁnding diﬀerences in edges and contours computed in Figure 6, a structured similarity index of 0.62 was found, which is very low similarity, hence making it diﬃcult to augment corridors. Therefore, it was decided to use door signs as markers. The ﬂoor used for the application had a number of doors, each with its unique sign placed at the centre. Therefore, the signs were used as markers for their static looks, and thus, it was very unlikely for the marker to diﬀer in diﬀerent images taken at diﬀerent times and angles, especially considering their unique attributes (Figure 7).

### 3D Models

To build the model targets, one must provide a 3D object within the Vuforia Model Target Generator, as explained in [14]. The 3D models generated were those of the door markers placed around the whole ﬂoor area as well as the coﬀee machine (Figures 8 and 9). The 3D models for the door markers were generated using the online library Selva3d which generates a 3D model from a given picture.

### Recommendation Data Set

The recommendation system was made up of a combination of item-based similarity and collaborative ﬁltering techniques. For the collaborative ﬁltering techniques, there was no existing dataset which one could make use of since there were no previous similar applications for users to rate the system. Therefore, a prototype recommendation system was built through generated data, simply to analyse how the system works with the augmented reality application. As previously mentioned, three tasks were picked for a visitor proﬁle which the collaborative techniques would be applied to, namely, an interview, delivery, and site-visitation. For each task, a dataset containing the user-tasks and the rating of that task was created. Every entry had the following attributes: user id, task id, locations of interest, user-task id, and rating. The task id represents the id of an interview, a delivery, or a visit. Locations of interest is a 7-bit binary code representing the accountant’s oﬃce, the human resources oﬃce, the manager’s oﬃce, the secretary’s oﬃce, the kitchen, the toilet rooms, and the games room respectively. Each dataset has a respective smaller dataset containing several locations of interest which were activated by previous users. The primary key of the tasks’ dataset is the foreign key within the user-task dataset called ‘user-task id’.

## Feature Extraction

### Image Targets

The ﬁrst step in the feature extraction process was feeding the images into the Vuforia library which extracts features from them, as shown in [14]. The library applies natural feature extraction to identify contours and edges from within the image, and consequently, display a 5-star rating according to how augmentable the image is. Initially, when corridors were tried as markers, the library had a problem with identifying key features from plain white painted corridors as they appeared to show no distinct edges and contours, and due to lighting, the images had a glossy texture which was harder for Vuforia to track. However, the door signs were eventually used as markers, and the images representing the markers were sharpened using OpenCV, taken in proper lighting scenes and fed into the library. Figure 10 illustrates the result. Sharpening enhances the strength of certain edges and corners, making the marker more detectable for the library.

### Model Targets

The second step was feeding the 3D models into the model target generator. Vuforia’s object recognition utilises natural feature tracking by analysing the object at 3 diﬀerent axes. Vuforia does not say speciﬁcally what deep learning techniques they use. However, according to [14], they are most likely making use of Interest Point Detection, which uses a set of images of the same object at diﬀerent scenarios, angles, and lighting. The model target generator takes a maximum of 20 model targets, whose CNN is then trained on that one 3D object, and meta data is outputted and later fed into Unity. The model target generator trained a total of 10 3D objects, which consist of the coﬀee machine and the door markers. As shown in Figure 11, the model’s distance, angle and orientation were adjusted for training, hence making the object easy to track.

### User-Query

The ﬁnal step was extracting features from the tasks’ dataset for the recommendation feature. The features extracted were the user id, user-task id, and rating. All the three tasks (i.e. interview, visit, and delivery) underwent the same process separately since there is no connection between each task, and thus, the recommendation system is best trained on each one individually. The recommendation system will be explained further in the next section.

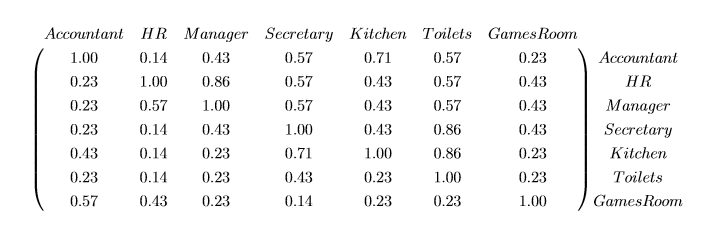
## User-Query Recommendation

### Overview

As previously explained in Chapter 4, along with the Augmented Reality, it was thought to implement a recommender system, depending on the user’s proﬁle (Visitor or Intern). The system would incorporate functionalities of an item-item similarity-based recommender system and functionalities of a collaborative ﬁltering system.

### Similarity Based

The item to item similarity-based recommendation consists of two main components, namely, a user to item matrix which, in our case, is the user query, that changes according to the user’s preference, and the item-item matrix. The latter matrix stays constant and represents the similarity between one location and another, based on whether the user passes right in front of it when trying to ﬁnd an oﬃce. The selected locations were the accountant’s oﬃce, the human resources oﬃce, the manager’s oﬃce, the secretary’s oﬃce, the kitchen, the toilet rooms, and the games room. Each row was computed using the term frequency approach. TF was chosen to enhance the relevant importance of a sought query, in this case, the oﬃce one wants to go to. TF-IDF was not used since matrix scaling was done using one-fold normalisation since a term (a location within the oﬃce) appears in every task (a task is equivalent to a document). Below is the ﬁnal normalised item-item similarity matrix.



### Collaborative-Based Filtering Techniques

The collaborative ﬁltering approach is a model based on the user’s past experiences of the decisions taken, locations of interest picked, and task they had in order to complete their own task. However, initially, there were no similar systems where users could rate their experience. Therefore, it was not possible to gather data, but instead, it was created solely to provide a prototype of the system and evaluate its performance. To build the collaborative ﬁltering techniques, the following python packages were used, namely, pandas, NumPy, and sklearn. Each dataset contains 101 user ratings. Following a similar procedure as used in [35], the system uses a truncated Single Value Decomposition to generate a utility matrix. To avoid having empty ﬁelds within the utility matrix, add-k smoothing is applied, where k is 0.01. The matrix is then transposed and decomposed using TruncatedSVD. It was decided to apply SVD-based algorithms instead of PCA or CA-CF in order to achieve higher accuracy [35]. Finally, a correlation matrix is generated, giving rise to the ﬁnal recommendation, where the results are all stored in separate CSV ﬁles and fed into Unity Editor.

The three top recommendations are then presented within the AR application. Whichever of them the user chooses, the system multiplies its item to item similarity matrix by that of the user’s query matrix using dot product, finally obtaining the result matrix, whose elements represent the size of a location’s gameobject within the 3D augmented map sketched. By applying a threshold value of 0.4 to a gameobject’s size, if its size is below this value, it simply does not appear within the 3D holographic map. The larger the size of an oﬃce pinpoint location is, the more it is recommended to the user. Therefore, that oﬃce has a higher relevance than the rest.

## User Interface and System Architecture

### System Architecture

There are three main components which make up the system architecture. The ﬁrst most essential component is the user, who is important as the system must serve as an essential tool to assist them around the workplace. Their decision-making process drives the capabilities of the Augmented Reality application to their extent. The second component is the recommendation system which, as previously explained, provides collaborative and similarity based ﬁltering techniques, depending on the user’s proﬁle. The third component is the Augmented Reality provided by Vuforia. This component can be further subdivided into two other sub-components, namely, the SDK library and the built architecture within Unity tailored for this project.

As explained in [17], Vuforia AR SDK consists of the smartphone’s camera and the target resources (the targets’ database) communicating with the tracker. The tracker then detects the real-world objects, converting each frame and snapshot to render augmented logic back on the user’s smartphone. Figure 12 shows graphically how Vuforia AR SDK works.

The built architecture within Unity consists of three features. The ﬁrst feature is called ‘Oﬃces’, which allows the user to wander around the workplace and view augmented information from the oﬃce markers. The second feature is ‘Locate’, which oﬀers oﬃce information and oﬄine directions to the user, depending on their preferred recommendation. The third feature is ‘Coﬀee Machine’, which provides employees and visitors with information on how to make use of the coﬀee machine. This information is provided through text, images, and video instructions.

The main challenge encountered was in the ﬁrst two features. Vuforia can detect multiple image targets, but can only detect one model target for every scene within Unity. However, using model targets in certain scenarios can provide a more eﬃcient user experience as the real-world object is more recognisable since deep learning techniques are utilised for recognition, rather than just traditional computer vision techniques. Therefore, a combination of image targets and model targets was used. The object is ﬁrst detected using traditional computer vision methods, and once it is recognised that the user is interested, the user prompts to view relevant information according to the marker being recognised by clicking the open button. The model target is activated once the user prompts to view the respective augmented information, and is deactivated once the user exits the augmented information. Therefore, through a combination of traditional computer vision and deep learning, a more accurate system is provided thereby offering eﬃciency and an immersive experience. It is further a solution to one of Vuforia’s lacking features which still does not enable the users to use multiple model targets in one. Therefore, the aforementioned technique served as a workaround to use multiple model targets.

### User Interface

When the application starts, the user is presented with three options: ‘Oﬃces’, ‘Locate’, and ‘Coﬀee Machine’ (Figure 13). ‘Oﬃces’ directs the user to the augmented reality system, where the user can wander around the workplace and view augmented information about the oﬃces. ‘Locate’ directs the user to select a proﬁle between an intern and a visitor. For the visitor’s proﬁle, the user chooses a task from the following options via a dropdown button: visit, interview, and delivery. As shown in Figure 16, the user is provided with three recommendations, each containing features which are recommended via an approval green sign, whereas those which are not recommended are represented by an X-sign in red. The approved signs for accountant, human resources, manager, and secretary are clickable. Once any one of them is clicked, it will direct the user to the Augmented Reality system, where they can view augmented information about the oﬃces via the door markers and locate the respective oﬃce which was previously clicked. Each oﬃce augments its own respective main menu (Figure 17).

The details panel provides details of the room, while the ‘Locate’ button augments a 3D hologram of the workplace and provides the user with hard-coded directions to the oﬃce they are interested in (Figures 19 and 20). It can be observed that the oﬃces in the hologram are colour-coded and are represented as a sphere. They are scale-wise recommended, that is, the larger the oﬃce marker appears, the more important and relevant it is to the user. The oﬃces which do appear in the hologram are the result of the collaborative and similarity based ﬁltering techniques provided by the recommendation system.

The ‘Coﬀee Machine’ functionality allows the user’s phone camera to recognise the DeLonghi Esam2600 coﬀee machine. The user can view text and video instructions on how to make a cappuccino, view a diagram of the machine’s functionalities, and view a 3D model of the machine.

## Conclusion

This chapter discussed thoroughly the implementation of the workplace assistant AR application, as well as the decisions taken for every feature provided. The next chapter will provide the tests and evaluations to highlight the system’s performance.

# References

[1] Model target generator user guide.

[2] Optimizing target detection and tracking stability.

[3] Advances in computer vision. Advances in Intelligent Systems and Computing, 2020.

[4] Gediminas Adomavicius and Alexander Tuzhilin. Context-Aware Recommender Systems, pages 217–253. Springer US, Boston, MA, 2011.

[5] Omer Akgul, H. Ibrahim Penekli, and Yakup Genc. Applying deep learning in augmented reality tracking. 2016 12th International Conference on Signal-Image Technology Internet-Based Systems (SITIS), pages 47–54, 2016.

[6] Hassan Alhaija, Siva Mustikovela, Lars Mescheder, Andreas Geiger, and Carsten Rother. Augmented reality meets computer vision : Eﬃcient data generation for urban driving scenes. International Journal of Computer Vision, 08 2017.

[7] Nader Barzegar and Shahroz Farjad. A study on the impact of on the job training courses on the staﬀ performance (a case study). Procedia - Social and Behavioral Sciences, 29:1942 – 1949, 2011. The 2nd International Conference on Education and Educational Psychology 2011.

[8] Herbert Bay, Tinne Tuytelaars, and Luc Van Gool. Surf: Speeded up robust features. volume 3951, pages 404–417, 07 2006.

[9] Gaurav Bhorkar. A survey of augmented reality navigation. ArXiv, abs/1708.05006, 2017.

[10] Carlos Plaza de Miguel. Arlodge : Context-aware recommender system based on augmented reality to assist on the accommodation search process. 2014.

[11] Nevin Deniz, Aral Noyan, and ¨Oznur Gu¨len Ertosun. Linking person-job ﬁt to job stress: The mediating eﬀect of perceived person-organization ﬁt. Procedia - Social and Behavioral Sciences, 207:369 – 376, 2015. 11th International Strategic Management Conference.

[12] James Ford and Tobias Ho¨llerer. Augmented reality: Information for workplace decision-makers, managers, workers and researchers. 02 2020.

[13] Mathieu Garon and Jean-Fran¸cois Lalonde. Deep 6-DOF tracking. IEEE Transactions on Visualization and Computer Graphics, 23(11), 2017.

[14] Ivar Grahn. The vuforia sdk and unity3d game engine : Evaluating performance on android devices. 2017.

[15] Isabelle Guyon and Andr´e Elisseeﬀ. An Introduction to Feature Extraction, volume 207, pages 1–25. 11 2008.

[16] T. Haritos and N. D. Macchiarella. A mobile application of augmented reality for aerospace maintenance training. In 24th Digital Avionics Systems Conference, volume 1, pages 5.B.3–5.1, Oct 2005.

[17] Alexandro Simonetti Ib´an˜ez and Josep Paredes Figueras. Vuforia v1.5 sdk: Analysis and evaluation of capabilities. 2013.

[18] Sumitkumar Kanoje, Sheetal Girase, and Debajyoti Mukhopadhyay. User proﬁling trends, techniques and applications. ArXiv, abs/1503.07474, 2015.

[19] Sumitkumar Kanoje, Debajyoti Mukhopadhyay, and Sheetal Girase. User proﬁling for university recommender system using automatic information retrieval. Procedia Computer Science, 78:5 – 12, 2016. 1st International Conference on Information Security Privacy 2015.

[20] Sneha kasetty sudarshan. AUGMENTED REALITY IN MOBILE DEVICES. PhD thesis, 05 2017.

[21] Stan Kurkovsky, Ranjana Koshy, Vivian Novak, and Peter Szul. Current issues in handheld augmented reality. pages 68–72, 06 2012.

[22] Andy Lee. Comparing deep neural networks and traditional vision algorithms in mobile robotics. 2016.

[23] Vincent Lepetit. On computer vision for augmented reality. pages 13 – 16, 08 2008.

[24] David Lowe. Distinctive image features from scale-invariant keypoints. International Journal of Computer Vision, 60:91–110, 11 2004.

[25] Paramjinang Moita. Adjustment to the work place by new recruits in libraries. Vol.5(2):71–85, 04 2015.

[26] Jonathan Pedoeem and Rachel Huang. YOLO-LITE: A real-time object detection algorithm optimized for non-gpu computers. CoRR, abs/1811.05588, 2018.

[27] David Proch´azka, Ondˇrej Popelka, Tomas Koubek, Jaromir Landa, and Jan Kolomaznik. Hybrid surf-golay marker detection method for augmented reality applications. Journal of WSCG, 20:197–204, 01 2012.

[28] Francesco Ricci, Lior Rokach, and Bracha Shapira. Recommender Systems Handbook, volume 1-35, pages 1–35. 10 2010.

[29] M. Romilly. 12 Best Augmented Reality SDKs. (2019, Jan 25).

[30] Edward Rosten and Tom Drummond. Machine learning for high-speed corner detection. In Aleˇs Leonardis, Horst Bischof, and Axel Pinz, editors, Computer Vision – ECCV 2006, pages 430–443, Berlin, Heidelberg, 2006. Springer Berlin Heidelberg.

[31] S. R. R. Sanches, M. Oizumi, C. Oliveira, E. F. Damasceno, and A. C. Sementille. Aspects of user proﬁles that can improve mobile augmented reality usage. In 2017 19th Symposium on Virtual and Augmented Reality (SVR), pages 236–242, Nov 2017.

[32] Rodrigo Silva, Jauvane Oliveira, and G. Giraldi. Introduction to augmented reality. 01 2003.

[33] Simon Taylor and Tom Drummond. Binary histogrammed intensity patches for eﬃcient and robust matching. International Journal of Computer Vision, 94(2):241–265, Sep 2011.

[34] Mathangi Vijayan. Impact of job stress on employees’ job performance in aavin, coimbatore. 06 2018.

[35] Manolis Vozalis, Angelos Markos, and Konstantinos G. Margaritis. Evaluation of standard svd-based techniques for collaborative ﬁltering. 01 2009.

[36] Hongzhi Yin, Yizhou Sun, Bin Cui, Zhiting Hu, and Ling Chen. Lcars: a locationcontent-aware recommender system. pages 221–229, 08 2013.

[37] Zhuo Zhang, Shang Shang, Sanjeev R. Kulkarni, and Pan Hui. Improving augmented reality using recommender systems. In Proceedings of the 7th ACM Conference on Recommender Systems, RecSys ’13, page 173–176, New York, NY, USA, 2013. Association for Computing Machinery.