

American University in Cairo

AUC Knowledge Fountain

Theses and Dissertations

6-1-2017

Context aware advertising

Youssef Youssef

Follow this and additional works at: <https://fount.aucegypt.edu/etds>

Recommended Citation

APA Citation

Youssef, Y. (2017). *Context aware advertising* [Master's thesis, the American University in Cairo]. AUC Knowledge Fountain.

<https://fount.aucegypt.edu/etds/677>

MLA Citation

Youssef, Youssef. *Context aware advertising*. 2017. American University in Cairo, Master's thesis. *AUC Knowledge Fountain*.

<https://fount.aucegypt.edu/etds/677>

This Thesis is brought to you for free and open access by AUC Knowledge Fountain. It has been accepted for inclusion in Theses and Dissertations by an authorized administrator of AUC Knowledge Fountain. For more information, please contact mark.muehlhaeusler@aucegypt.edu.

The American University in Cairo

Computer Science and Engineering Department

Context-Aware Advertising

A thesis submitted to the Computer Science and Engineering Department in partial fulfillment of the requirements for the degree of Master of Science in Computer Science

By Youssef Khater Erian Youssef

B.Sc. Computer Science, AUC, February 2005

Under the Supervision of Dr. Sherif G. Aly, Professor

May 2017

The American University in Cairo

Context Aware Advertising

A Thesis Submitted by

Youssef Youssef

To Department of Computer Science and Engineering

May/2017

In partial fulfillment of the requirements for the degree of Masters of Science

Has been approved by

Thesis Committee Chair / Adviser: SHERIF G. ALY

Affiliation: PROFESSOR, DEPT. OF COMPUTER SCIENCE AND ENGINEERING

Thesis Committee Reader / examiner: AHMED RAFAA

Affiliation: PROFESSOR, DEPT. OF COMPUTER SCIENCE AND ENGINEERING

Thesis Committee Reader / examiner: TAMER ELBATT

Affiliation ASSOCIATE PROFESSOR, FACULTY OF ENGINEERING, CAIRO
UNIVERSITY

Department Chair/ Date: _____

Dean/Date: _____

Program Director/Date: _____

Acknowledgements

I would like to thank everyone who supported this research to the well roundedness and maturity levels it has reached. I would like to thank Microsoft Azure for funding this research through the Azure for Research funding program for USD 20,000. This fund was directed towards consuming azure credits to implement and test my research.

From AUC, I would like to thank my advisor Dr. Sherif Gamal Aly for his patience and support throughout the duration of this work and for the previous mentorship and guidance I received during my undergraduate and graduate course work. His invaluable ideas, suggestions and friendly talks have not only contributed to my research, but also advanced me as a person. I would like to thank Dr. Sherif for the time and dedication he has put with me and support despite the time and distance challenges. I would also like to thank Dr. Ahmed Rafea, the internal examiner for this thesis, whose feedback helped in validating and further establishing the contributions of this work.

My friends at Media Line agency who contributed to this work by providing a corpus of 421 ads that were displayed during Ramadan 2016, thank you!

I would also like to thank my amazing wife, Mary Shawky, for her tremendous support throughout the years, pushing me forward and availing the environment to be productive. I would like to thank all my family members for their encouragement, support and prayers.

I would like to thank all who dedicated their time and efforts to contribute in the different surveys and experiments conducted; whether by submitting ratings through the data gathering app, which were used to train the recommendation engine, or by conducting the experiment to validate the methodology and compute its effectiveness. Your invaluable contribution have been crucial to the success of this work.

Abstract

IP Television (IPTV) has created a new arena for digital advertising that has not been explored to its full potential yet. IPTV allows users to retrieve on demand content and recommended content; however, very limited research has been applied in the domain of advertising in IPTV systems. The diversity of the field led to a lot of mature efforts in the fields of content recommendation and mobile advertising. The introduction of IPTV and smart devices led to the ability to gather more context information that was not subject of study before. This research attempts at studying the different contextual parameters, how to enrich the advertising context to tailor better ads for users, devising a recommendation engine that utilizes the new context, building a prototype to prove the viability of the system and evaluating it on different quality of service and quality of experience measures.

To tackle this problem, a review of the state of the art in the field of context-aware advertising as well as the related field of context-aware multimedia have been studied. The intent was to come up with the most relevant contextual parameters that can possibly yield a higher percentage precision for recommending advertisements to users. Subsequently, a prototype application was also developed to validate the feasibility and viability of the approach. The prototype gathers contextual information related to the number of viewers, their age, genders, viewing angles as well as their emotions. The gathered context is then dispatched to a web service which generates advertisement recommendations and sends them back to the user. A scheduler was also implemented to identify the most suitable time to push advertisements to users based on their attention span.

To achieve our contributions, a corpus of 421 ads was gathered and processed for streaming. The advertisements were displayed in reality during the holy month of Ramadan, 2016. A data gathering application was developed where sample users were presented with 10 random ads and asked to rate and evaluate the advertisements according to a predetermined criteria. The gathered data was used for training the recommendation engine and computing the latent context-item preferences. This also served to identify the performance of a system that randomly sends advertisements to users. The resulting performance is used as a benchmark to compare our results against.

When it comes to the recommendation engine itself, several implementation options were considered that pertain to the methodology to create a vector representation of an advertisement as well as the metric to use to measure the similarity between two advertisement vectors. The goal is to find a representation of advertisements that circumvents the cold start problem and the best similarity measure to use with the different vectorization techniques. A set of experiments have been designed and executed to identify the right vectorization methodology and similarity measure to apply in this problem domain.

To evaluate the overall performance of the system, several experiments were designed and executed that cover different quality aspects of the system such as quality of service, quality of experience and quality of context. All three aspects have been measured and our results show that our recommendation engine exhibits a significant improvement over other mechanisms of pushing ads to users that are employed in currently existing systems. The other mechanisms placed in comparison are the random ad generation and targeted ad generation. Targeted ads mechanism relies on demographic information of the viewer with disregard to his/her historical consumption. Our system showed a precision percentage of 69.70% which means that roughly 7 out of 10 recommended ads are actually liked and viewed to the end by the viewer. The practice of randomly generating ads yields a result of 41.11% precision which means that only 4 out of 10 recommended ads are actually liked by viewers. The targeted ads system resulted in 51.39% precision. Our results show that a significant improvement can be introduced when employing context within a recommendation engine. When introducing emotion context, our results show a significant improvement in case the user's emotion is happiness; however, it showed a degradation of performance when the user's emotion is sadness. When considering all emotions, the overall results did not show a significant improvement. It is worth noting though that ads recommended based on detected emotions using our systems proved to always be relevant to the user's current mood.

Table of Contents

Acknowledgements.....	3
Abstract	4
Table of Contents.....	6
Table of Figures	8
Chapter 1: Introduction	10
1.1. Background.....	10
1.2. Problem Definition.....	11
1.3. Thesis Statement.....	12
1.4. Contribution.....	13
1.5. Highlights of our approach.....	15
Chapter 2: Related Work.....	17
2.1. Context Aware Advertising.....	17
2.2. Context Aware Multimedia	22
2.3. User Modeling	33
2.4. Recommendation Engines.....	35
2.4.1. Context-Aware Recommendation Systems (CARS)	39
2.5. Targeting Advertisements	42
2.6. Quality of Service and Quality of Experience.....	45
2.7. Quality of Context (QoC).....	47
2.8. Nomadism	53
Chapter 3: Proposed Approach.....	56
3.1. Contextual Parameters	56
3.2. Recommendation Engine	57
3.2.1. Vectorization of Advertisements	58
3.3. High Level System Architecture	59
3.4. Class Diagram.....	61
3.5. Interface descriptions	62
3.6. Communication Model.....	63
3.7. Context Sources	66
3.8. Adaptation Modules.....	67
3.8.1. Recommendation Engine	67

3.8.2.	Delivery Engine	68
3.8.3.	Utilities Maximization	68
3.8.4.	Security	69
3.8.5.	Video Synthesis	69
3.9.	Client-Side Architecture.....	69
3.9.1.	Context Information.....	69
3.9.2.	Client Components	71
3.9.3.	Client Use Case.....	73
3.9.4.	Context Data Flow	73
Chapter 4:	Implementation	78
4.1.	Client Side	78
4.2.	Recommendation Engine	82
4.3.	Media Rendering.....	89
4.4.	Storage.....	89
4.5.	Back-end processes.....	90
4.5.1.	Create ad vectors.....	92
4.5.2.	Ads similarity matrix creation	94
4.5.3.	Ad similarity matrix index creation	99
4.5.4.	Latent context-item preferences matrix creation	99
4.6.	Data Gathering.....	100
Chapter 5:	Evaluation and Experimentation.....	104
5.1.	Evaluation Criteria	104
5.2.	Metrics.....	105
5.3.	Recommendation engine modeling	105
5.3.1.	Experiment setup	107
5.4.	Back-end performance	109
5.5.	Front-end performance.....	110
5.6.	Human survey.....	110
5.7.	Quality of context	112
Chapter 6:	Results and discussions	114
6.1.	Contextual parameters.....	114
6.2.	Recommendation model approach.....	115
6.3.	QoS results of the solution	116

6.4. QoE results of the solution	119
6.5. QoC results of the solution	120
6.6. Other results.....	123
Chapter 7: Conclusion.....	125
Appendices	126
Appendix I - API References.....	126
Appendix II - Detailed experimental readings.....	128
Appendix III - Detailed advertisement vectors representation	132
Appendix IV – Context representation.....	135
References	140

Table of Figures

Figure 1. High level context aware architecture in UP-TO-US	23
Figure 2. Proposed context aware system by UP-TO-US project	24
Figure 3. IPTV implementation architecture using IMS [17]	28
Figure 4. Resulting conceptual framework from [18].....	29
Figure 5. User modeling in [19].....	34
Figure 6. Middleware architecture proposed by [20].....	35
Figure 7. Recommendation engine architecture in UP-TO-US project	38
Figure 8. Comparison of recommender system libraries adapted from [23].....	38
Figure 9. Adopting DISC psychological model for content recommendation [25].....	39
Figure 10. Paradigms of incorporating context into a recommendation engine [54]	40
Figure 11. Probability precision used in UP-TO-US project	45
Figure 12. Estimated Browsing time employed in UP-TO-US project	46
Figure 13. Video quality indicators adapted from [35].....	47
Figure 14. Classification of quality parameters adapted from [58]	49
Figure 15. Classification of QoC information adapted from [64]	52
Figure 16. System high-level architecture.....	60
Figure 17. Class Diagram	61
Figure 18. IContextCollector interface	62
Figure 19. IAnalyzer interface.....	63
Figure 20. Communication model between system components	64
Figure 21. Detailed system components and employed information.....	66
Figure 22. Client components.....	71
Figure 23. Client use case.....	73
Figure 24. Client data flow	75
Figure 25. Sequence diagram on the client side	77
Figure 26. Sample home screen.....	78

Figure 27. Snippet of captured user information using login.....	80
Figure 28. Snippet of captured user information without logging in.....	81
Figure 29. Recommendation engine decision flow.....	83
Figure 30. Context based ad recommendation for one user.....	84
Figure 31. Ad recommendation for multiple users.....	85
Figure 32. Predicting ad ratings for specific user.....	88
Figure 33. System ER Diagram.....	90
Figure 34. Summary of back-end process.....	91
Figure 35. Flow chart for advertisement vectorization.....	94
Figure 36. Flow chart for ad similarity matrix computation.....	96
Figure 37. Average similarity measure computation.....	97
Figure 38. Cosine similarity and Pearson coefficient computation flowchart.....	98
Figure 39. Latent context-item preferences computation.....	100
Figure 40. Snapshot of the data gathering application.....	102
Figure 41. Snapshot of web experiment application.....	112
Figure 42. Ads similarity matrix creation time.....	116
Figure 43. Ads similarity matrix index creation time.....	117
Figure 44. Latent context-item preferences computation time.....	117
Figure 45. Correctly identified happy face.....	122
Figure 46. Correctly identified neutral face.....	122
Figure 47. Correctly identified sad face.....	123
Figure 48. Unidentified sad face.....	123

Chapter 1: Introduction

1.1. Background

The introduction of IPTV introduced a new realm for giving users control over the entertainment industry. With IPTV, users get to choose the exact content and timing of delivery. Accordingly, many research efforts have been made to identify and push relevant content to users to better engage them and maintain their loyalty. On the other hand, the associated advertisement industry followed pace, and marketers started shifting their efforts to include advertising plans in IPTV systems. To the disappointment, the methodology for selecting and delivering advertisements in IPTV systems is far from mature. Despite the efforts conducted in the fields of content recommendation in general, and context-aware advertising for mobile devices, little investment had been conducted in the fields of context-aware advertising for IPTV systems in specific.

To the best knowledge of the author, the contextual information that is realistically used to recommend advertisements in IPTV systems is far from significant. Accordingly, a need arises to build systems that are capable of gathering and using richer context information from both the physical and virtual worlds, such as from smart devices and social networks, and using them to create better advertisement recommendations in IPTV systems. A recommendation engine must also be devised to capitalize on the richer set of context parameters and allow marketers to better target their advertisements in a way that best satisfies their marketing strategy (either aim for larger reach or higher frequency for target groups).

This work studies related efforts in context-aware advertising including digital media, mobile advertising, as well as IPTV systems. We also study the related efforts in the broader field of context-aware multimedia, which in itself is deficient in the way it focuses primarily on recommending valid content alone instead of also recommending valid advertisements. Due to the higher investment in context-aware multimedia, richer research efforts have been conducted and lessons learned from them will be adapted to this research. We also study different efforts conducted on user modeling and recommendation engines as sub-problems of this research, as well as marketing techniques used and how to adapt them in our work.

In a nutshell, we will be evaluating new contextual parameters that have not been fully utilized in the field of context-aware advertising in IPTV systems. We will also build a prototype to gather the different context information from smart devices and utilize cloud services to enrich the context data into context information. A recommendation engine will be developed that is capable of utilizing the new contextual parameters while taking into consideration the defined marketing strategy and objectives set by marketers (or ad bidders). The goal is to offer advertisements recommendation in a way that enhances the view-ability of the advertisement by its target audience to the end (watch the whole advertisement not only part of it). The systems in comparison will be comprised of two parallel systems, one that offers random recommendations and one that offers targeted ads based on user demographic information. A set of metrics are defined for the evaluation of the performance of our approach from various aspects including quality of service, quality of experience and quality of context. Details about the metrics used for each aspect are listed in the Experimentation section.

1.2. Problem Definition

The invention of IP Television (IPTV) created an opportunity for product and service providers to better target their audience and cater for personalized interest in media content. Much of the targeted content delivered through IPTV relies on an on-demand type media delivery. In most cases, user preference and history of usage influence relevant media delivery, similar to what happens in social media. However, very limited research contribution has been made to use the extremely rich set of contextual data of viewers to be used as an influence for media delivery to target audience. Although context could be as simple as location and physical presence, it can be as diverse as viewer gender, attention span, estimated age, interest, emotion, and likelihood of making purchasing decisions. The selection of an appropriate and effective set of contextual parameters that will be used to customize advertisement delivery must be carefully chosen and experimented with a way that can maximize certain utilities aligned with the objective of the advertisement. Not only should this allow for an increase in the viewership of each advertisement, and expected revenue, but can also enhance the impact of advertisements on the target audience. This could be seen as a form of behavioral targeting of content to users, which is a twenty billion dollar industry that is exceptionally growing at an unprecedented pace by major market players.

1.3. Thesis Statement

The objective of this work is to create and evaluate a system that is capable of recommending ads to users with adequate precision. This is meant to increase the effectiveness of ads by establishing context aware targeting. Our measure to imply interest in an ad is by a viewer watching the ad till its end. Our approach in this work is to survey the domain of context-aware advertising and other related domains such as context-aware multimedia to identify the current and potential contextual parameters that can be employed to better target advertisements to viewers in IPTV systems. We also study the underlying enabling domains such as recommendation engines, and context-aware recommendation engines to identify the approach to follow and different methodologies for incorporating context information within a recommendation engine. We also survey the fields of targeting advertisements to explore the various techniques used in the field of marketing, user modeling and nomadism. Finally, we also survey the different quality aspects that pertain to similar applications such as quality of service, quality of experience, and quality of context to identify the various metrics used in the domain and design ones to use for evaluating our approach.

A demo application was built based on the results of the various studies to prove the viability of the approach and test its effectiveness. The application incorporates contextual information from multiple sources such as social media, camera device and location services, to identify the profile of the current viewers. The application also detects user emotions to recommend ads that are suitable for the viewer's current mood. Recommendations are supplied to individual users and group of users viewing from the same device.

From a recommendation engine perspective, we propose an approach that incorporates contextual information using pre-filtering and post-filtering. We also present a methodology for vectorization of advertisements to be used in item-based filtering recommendation approach that is not vulnerable to the cold start problem.

Various metrics have been designed and measured to qualify the various quality aspects of the applications. Quality of service, quality of experience and quality of context have been measured and reported on. An experiment is designed to allow users to receive recommendations under varying moods to test the precision of the recommendation engine and the suitability for the user's current mood (quality of experience). We also use this experiment to measure the accuracy of the context gathering techniques (quality of context). Execution times and turnaround times for all components in the system have been measured to identify the quality of service.

1.4. Contribution

The main contribution of this research is to employ richer context information to enhance the effectiveness of advertisements delivered in IPTV environments in a way that surpasses current practices of regular TV, Set-Top-Boxes (STBs), and online advertising. Multiple specific contributions will be achieved throughout this research. To achieve and build confidence in the viability of our contribution, multiple objectives need to be achieved.

1. The first objective is to define the appropriate contextual parameters for recommending advertisements. A prototype will be developed for gathering this context information and applying it to a recommendation engine. A study of relevant types of contextual parameters, and their effectiveness in advertisement recommendation will be conducted. This study will cover the available context information and cross-reference them with other related fields such as context-aware multimedia delivery. The contribution will include the context sources involved, collection mechanism and inference or enrichment procedures (getting higher level context information from raw context data).
2. Building a recommendation model that is context aware and scalable. It also includes building a delivery engine that supplies users with recommended ads in appropriate times.
3. Identifying the best approach to represent advertisements in a vector representation and selecting the most appropriate similarity measurement in a manner that is not vulnerable to the cold start problem.
4. Scheduling and prioritization of advertisements based on a confidence level that is calculated according to the proximity of the ad to user interests either statically gathered by the system profile or dynamically from social networks as well as suitability for the current user's emotion context. Different techniques for utilizing content intrinsic properties (such as brand, genre, target age group, etc.) along with historical transactions (e.g. user ratings) will be proposed to use together for computing content similarity. Current approaches either calculate similarities based on content properties or based on historical transactions (such as collaborative filtering and item-based filtering) but not both. The former approach is sometimes used as default when no historical transactions are available [36]. This research will attempt to apply different techniques in constructing the vectors describing each content item in a way that expresses the content properties itself along with its

historical transactions. The challenge posed by this approach is mixing nominal attributes, such as genre, with numerical attributes, such as ratings. The proposed approach section will discuss different options as well as their advantages and disadvantages.

5. Techniques for supporting nomadism will be discussed and a methodology for this work will be proposed.
6. The recommendation engine shall support recommending advertisements to a group of users sitting in front of the same viewing device (like a group watching a football match together or a family watching a show on the same device).
7. Experimentation will be conducted upon the prototype and overarching used architecture to demonstrate effectiveness in delivering advertisements to target audience as compared to other systems. Other systems include random ad generation and ad targeting. We will be using metrics of Quality of Service (QoS) as well as Quality of Experience (QoE) for the purpose of evaluation as well as Quality of Context (QoC). The main metrics used for evaluation are:
 - a. Delay of delivery: How much time it takes to deliver a recommendation to the user by the recommendation engine.
 - b. Precision of recommendation: Calculated by checking the items recommended by the system and relevant to the user, divided by the total recommendations sent to that user. In our case the expression of interest can be implied by viewing the advertisement till its end.

The first metric proposed measures the quality of service, while the second metric measures the quality of experience as perceived by the user from recommendation precision perspective. More details about the metrics and experiments design can be found in the Experimentation section.

This research neither handles advertisement bidding, nor filtering advertisements based on remaining budget, that, which is left to an industrial scale application. Bidder information may include information about the target audience profile, target location, bid per viewership (how much they are willing to pay per viewership) and total budget to limit their spending. In addition, the advertising strategy can be taken into consideration. Advertising strategies can target for maximum reach (reaching as many unique number of viewers) or for maximum frequency (targeting the same set of users multiple times that can be capped at a certain

frequency). These strategies are important for optimizing the advertiser's budget according to the marketing objectives.

1.5. Highlights of our approach

Our work identifies approaches for two main areas, context gathering and advertisement recommendation. For context gathering, multiple sources are considered to retrieve different types of information. We also consider redundant sources supplying the same information, one is used as a primary source of information and the other is a secondary source of information when the primary source is not available. The main sources of contextual information are the social network (Facebook), camera device and location services. The access to social network information requires users to log in using the social network credentials and granting access on profile information to the application, thus social networks are used as the primary source for profile information. In case the user denies access to social network profile, the camera device is used to identify demographic attributes of the user such as age and gender, secondary source of information. It is worth noting that it is not always the case that social networks are the primary source of information. In case of location information, the location services are the primary source of information and social network data can be used as a secondary source in case of the unavailability of the primary source of information. Our context gathering approach also includes context enrichment techniques to retrieve abstract level of information from raw context data, such as GPS coordinates data enriched into region level information.

The second main area is the recommendation engine approach. The first decision that needed to be made is how to represent advertisements. This is referred to as vectorization of advertisements and the various options are detailed in the proposed approach section. The second decision that needed to be made is the similarity measure to use which is one out of three options: cosine similarity, Pearson coefficient and average similarity. All the available techniques have been implemented to be evaluated for use in real-life scenarios. A data gathering application which presents random advertisements to users was implemented to gather explicit ratings from users. These ratings were then used to train the recommendation engine and test the precision of the various approaches. The test relied on computing a recommendation and finding out if it existed in the list of ads rated positively by users. The decided upon approach is further evaluated in another experiment with real users.

Further details on the approach is presented in chapters 3 and 4. Details on the methodology for experimentation are detailed in chapter 5.

Chapter 2: Related Work

Context awareness has been applied in multiple related fields. The concept was applied in the field of online advertising, mobile advertising and IPTV systems. It was also applied in Context-Aware Multimedia to recommend relevant content based on user's context. Another related field to this study is Recommendation Engines. Recommendation Engines are covered here to identify the most suitable engine for this research. Similarly some sub-problems of this research have been tackled separately by researchers such as User Modeling, Advertisement Targeting and others. This section will be divided into several subsections, each tackling the related work in the relevant field of study.

2.1. Context Aware Advertising

Several efforts have been conducted in the field of context-aware advertising. The efforts span mobile advertising and IPTV systems as well. In IPTV, the research in [1] proposes a mechanism for ad placement as well as a bidding model on user profiles that is inspired by Google AdWords. Their bidding model requires advertisers to bid on user profiles as well as time slots in which the ad is to be displayed. The model they created is designed to support legacy models of ad placement, which is bidding on programs based on viewership rather than bidding on user profiles. Their ad placement algorithm is aimed to maximize service provider revenues. Their personalization mechanism factors in the web browsing activities of users as well as the TV viewership activities. In their discussion, they realize that full personalization is not feasible as this will cause a major overhead on the network, so instead they recommend ads to groups of users to increase bandwidth efficiency. However, we realize that the communication technology infrastructure is growing in capacity and we choose to create a model that supports recommending advertisements to individuals as well as groups of users. The authors in [1] mention developing a working prototype for their model, but they did not report on its precision, effectiveness or performance.

The patent in [2] offers a system and method for personalized advertising in IPTV systems. Their method relies on periodically pushing ads to PVR's in a household to be displayed during the ad insert period. Their decision for choosing ads relies on geography, demographics, time of day and current program on TV. Like [1], they realize that full personalization is not feasible due to limited bandwidth, so they evaluate the decision parameters over multiple users grouped by distribution areas and delivering the same package of ads to the distribution area. The patent

in [2] factors in user interests to filter ads. It also includes micro-level granularity of location such as room in a household which is pre-set during the installation of the system. This patent offers a business model that is closest to current advertisements business model in broadcast networks. Implementation and performance metrics are not mentioned in the patent. Due to the fact that grouping of users is mainly done over a distribution area that is defined geographically, a lot of the parameters will average out such as age or gender. We believe that this will result in defaulting back to the broadcast model where advertisers bid on content not user profiles.

Context-Aware Advertising does not only pertain to IPTV systems, but it also applies to mobile advertising and perhaps there are lessons learned in mobile advertising that can be applied or adapted in IPTV systems. Several efforts have been conducted in context-aware mobile advertising that focus on developing a research model, a recommendation algorithm and a delivery system.

The research in [3] focuses on developing a research approach for personalized mobile advertising with focus on user modeling techniques. Their modeling technique is based on the factors that influence a buyer's decision such as buyer's individual characteristics, the environment, price and promotion. Their recommendation model is based on information on three main categories. The first category pertains to the context and includes information like location, weather, user activities and time of day. The second category focuses on content which is defined by the price, brand, and the promotion. The third is related to demographics and the fourth is user preference. Since they focused on Food industry, user preference is defined by cuisine, ambience, service, and food to recommend restaurants. Their research model is based on surveying users then set prior probabilities on a Bayesian network using the survey data and use this information to recommend restaurants to users. They performed an experiment and their results statistically show that sending recommended advertisements is more effective than sending random advertisements by enhancing users' attitude towards the ad and increasing willingness to utilize these ads. However, there are some limitations to their research as it was applied in China and needs to be proven in other countries. There is no report on the performance of their implementation so there is no indication on how many users can it support or how many advertisements it can recommend in a certain period of time.

The research in [4] proposes a new recommendation engine that is a hybrid between collaborative filtering (CF) and genetic algorithm (GA). They also focus on mobile advertisement, therefore their context information is defined by location (visiting area), time which defined by visiting day (working day or week end) and time (morning, lunch, dinner, afternoon, night). They also include information on user needs: utilitarian (practical needs) or

hedonic (pleasure needs). The user need information is not detected but manually input by the users. In brief their model is used to predict user rating on a product so they use Pearson Coefficients to detect user similarities and weigh those coefficients by weight factors generated using GA which is used to find the similarities between contexts. Their results show that they can gain the most precision by combining all the sources of context information and improve the precision of predicted ratings only using collaborative filtering approach. They also prove that the precision of predictions are statistically better using paired-samples t-test. The research suffers from the limitations of data scarcity and, therefore, the results of measuring similarities between users may be uncertain. Their approach is still invasive since users have to input their type of need (utilitarian or hedonic). The effectiveness of this approach is yet to be proven in a real-world scenario. In addition, they do not report on performance, so we cannot estimate the scale of users or rate of recommendations they can handle.

A sample mobile advertisement delivery mechanism is portrayed in [5] where they attempt to have a non-intrusive method of delivering mobile advertisements that are personalized to the target users. They implement the advertisement delivery mechanism as an overlay on a tourist guide mobile app. Their objective is to deliver advertisements to users when they need them, where they need them and how they need them (in a form sensitive to their technological context). Their context information is divided into four main categories: User context, computing context, physical context and history context. User context is defined by user identity, profile, location and orientation. Computing context is defined by network connectivity and bandwidth, type of device, and type of operating system. Physical context is defined by the surrounding environment and nearby objects. History context is a trace of all the above contexts recorded across a time span. Their research was performed in 2004 and used outdated devices, however this paper proposes a method of detecting users' emotions to be included in the context information and used in the recommendation engine. To detect emotions it is necessary to read information on blood volume (BVP), heart rate (EKG), galvanic skin conductance (SC), and respiratory rate which are commonly used in emotion research experiments. These measurements are used to detect emotion information on 2 axes: valence which is the type of emotion and arousal, the intensity of emotion. Perhaps it was not feasible to read such information to detect user emotions in 2004, but today with the penetration of wearable devices some of these measurements can be recorded and perhaps turn out to be useful in identifying user emotions in terms of type or intensity or both. Again, the authors do not report on the precision or effectiveness of the recommendation engine. They also do not mention the scale of users they can handle.

When it comes to the information taken into consideration when making context-aware decision, each research took a different set of parameters as can be seen in the previous section. The following table summarizes the set of context information used in each recommendation model. This table contains attributes that pertain to Context-Aware Multimedia in addition to Context-Aware Advertising. While some of these parameters do not apply to advertising, other parameters do apply and can be adapted to the field of advertisement. For the applicable parameters, it is crucial to see how the contributions in Context-Aware Multimedia incorporated those parameters in their recommendation model.

Table 1. Comparison of context information in context aware advertising

Feature/Paper	[1]	[2]	[3]	[4]	[5]
Type	Paper	Patent	Paper	Paper	Paper
Target	Advertising	Advertising	Advertising	Advertising	Advertising
Context Information					
User Context					
Age		Yes	Yes		
Identity					Yes
Profile/Preferences		Yes	Yes		Yes
Location		Yes	Yes	Yes	Yes
Granularity of location		Room	GPS	Visiting Area	GPS and Abstract Street/Building
User Activities	Web Browsing Activities TV Viewership Activities		Yes		
Emotions					Yes (proposed)
Needs				Utilitarian Hedonic	
Agenda					
Consumption History					
Gender					

Purchase History					
Service Context					
Current Program		Yes			
Brand Information			Yes		
Promotion			Yes		
Content Language					
Content Format					
Content Description					
Access rights					
Location					
Current Content Viewership Rating/Actual Viewership					
Computing Context					
Network and Bandwidth					Yes
Device Type			Yes		Yes
Device Status					
Operating System					Yes
Physical Context					
Surrounding Environment					Yes
Nearby Objects					Yes
Weather			Yes		
History Context					
Time		Day		Week day Week end Visiting Time	Time of Day Day of Week Month Season
Privacy					

Session Mobility					

As we can see in the table above, most of the information comes from statically input data (such as user profile and interests). Very few parameters are automatically read by the different systems (dynamic data are highlighted in the table above). This increases the level of personalization of displayed advertisements, but leaves room for enhancing the context-awareness of the recommendation engine.

2.2. Context Aware Multimedia

Related work in Context-Aware Multimedia is studied due to the common background and similar functionality. In multimedia, the same devices are used for viewership and the same users are in concern. A lot of the context parameters also apply like location, age, gender, etc. The only difference is that in Multimedia, the context is used to recommend relevant content while in advertising, the context is used to recommend relevant advertisements. The UP-TO-US project described in [8-13] is perhaps the most extensive work in context-aware multimedia covered in this section. UP-TO-US project focuses on three main areas of personalization. First it focuses on Electronic Program Guide (EPG) personalization through inclusion of context-aware content recommendation. Second, it focuses on content mobility in user's domestic sphere. This means that content should follow the user by detecting the location he is currently in within his/her household. Third, it focuses on content personalization during nomadism. Nomadism is the process of accessing content as a guest from another environment such as visiting a friend's place. The UP-TO-US project focuses on how to maintain personalization when a user is a guest in a foreign environment (friends place, hotel room, etc.). The project is divided into several modules. The first module is the Application Layer and consists of several components. The first component is the recommendation system, the second is the user profile management, the third is for the User Equipment (UE) Service continuity, and the fourth is for the nomadic service component. The second module is the Privacy and Security Module which focuses on verifying user's policies, allowing users to set flexible policies regarding who and in which situation may access their context status and allows users to block access to some applications if necessary. The data model used by this project is an Ontology language which

is stored in a relational database. The data inference is performed using Pellet and the rules are specified using SWRL. Figure 1 is a high level architecture of UP-TO-US project.

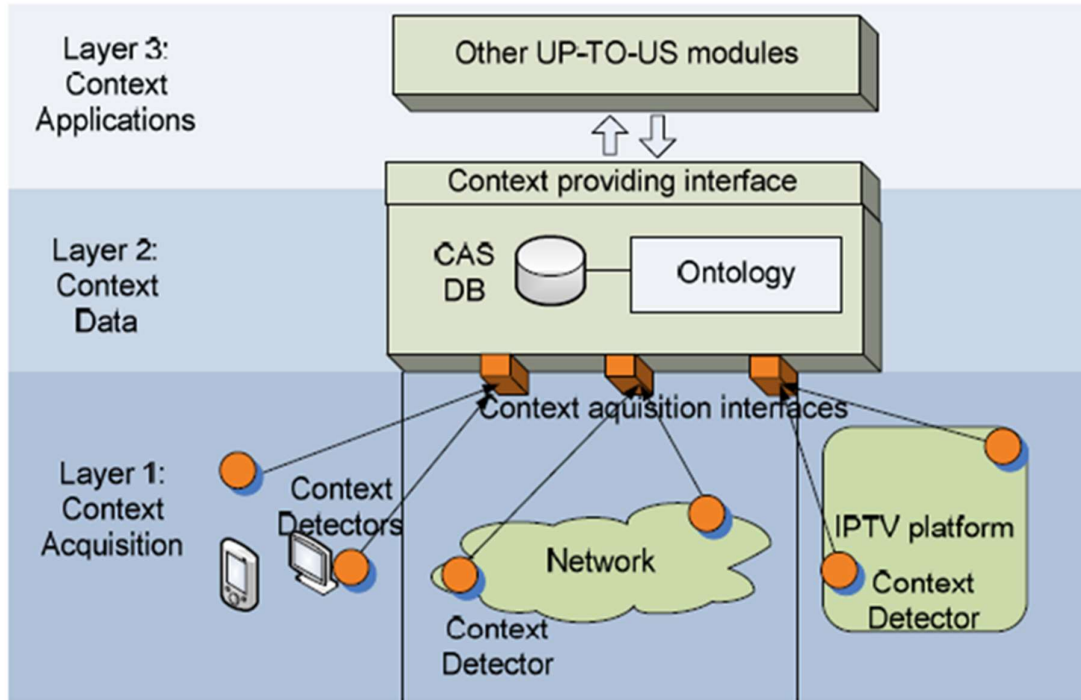


Figure 1. High level context aware architecture in UP-TO-US

Based on the described purpose for UP-TO-US project, the following are the supported use cases:

- Content adaptation according to each individual user and group users' preferences: allowing each user or a group of users to have personalized content matching their preferences
- Content customization according to the user context and QoE (Quality of Experience): allowing each user to have personalized content matching the user context (age, gender, region, preferences, location in the home environment or outside, activity) and thus optimizing the level of satisfaction.
- Content following the user during his mobility in his domestic sphere: allowing the user to move around within his domestic sphere while continuing accessing his IPTV service personalized according to the characteristics of the device in his proximity.
- Content personalization during nomadism: allowing the user to access his personalized IPTV content in a nomadic situation like in a hotel, in a friend's house or anywhere outside his domestic sphere.

According to UP-TO-US there are four main categories of context information. The first category is the User Context. This is an aggregation of static and dynamic information about the user. Static information mainly consists of the user profile, while the dynamic information are the ones collected by either sensors or other services. Information collected by sensors is mainly the user location. Other dynamic information collected by services include the user's usage history, calendar agenda, ratings on content, and previous content purchases. There is also a second layer of information called Inferred Information that is high level information deduced by a change in pattern in the previously described information. The second type of context is the Device or Terminal Context. This is mainly composed of the device identity, device capabilities, and available network connectivity. The third type of context information is the Network Context which is mainly described by technology type (ADSL, 3G, fiber optics, etc), transmission capacity, current load, available bandwidth, packet loss ratio, delay and jitter. The fourth type of context information is the Service context. Service context relates to content information coming from the content provider. Such information contains the data about the service provider himself, content language, content format, content metadata (title, genre, duration, etc.), access rights and content location. The following diagram describes the proposed context-aware system by UP-TO-US project.

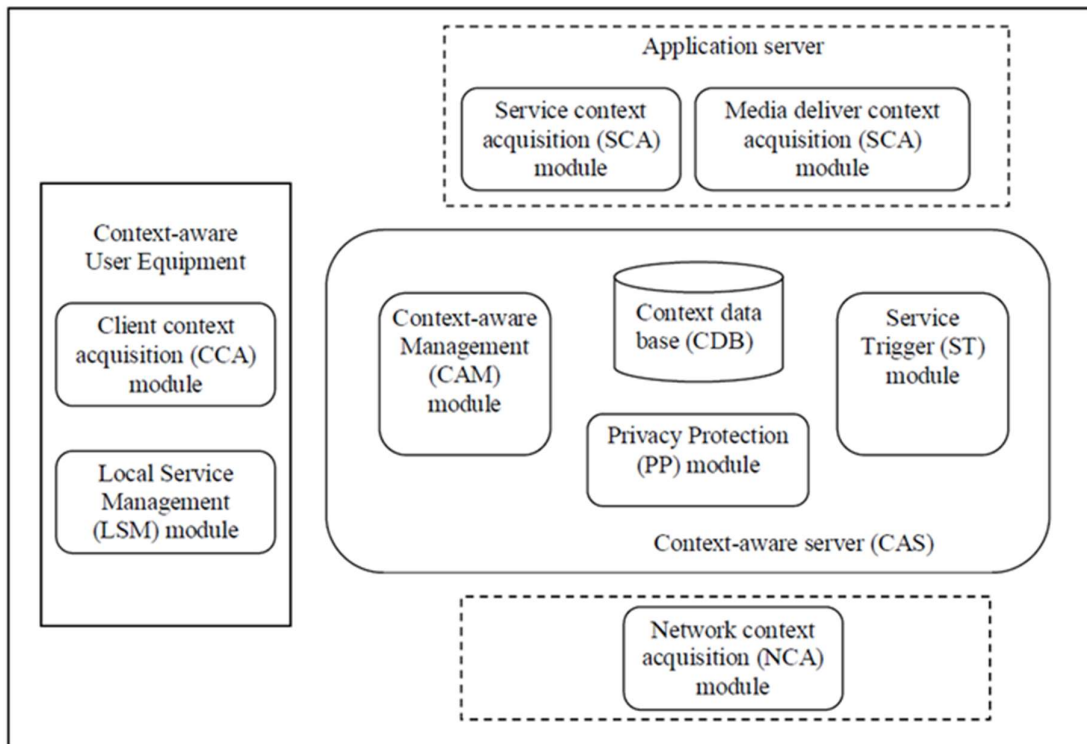


Figure 2. Proposed context aware system by UP-TO-US project

According to UP-TO-US a proposed context-aware system should have a context-aware server (CAS). This CAS can be composed of several modules such as management, privacy, service trigger modules and a database. Context-Aware Management (CAM) module gathers the context information from the user, the application server and the network. CAM supports the context inference which helps in transforming lower level context information to a higher level context. The reasoning techniques such as rule based reasoning, probabilistic reasoning, etc. could be used here. Context Database (CDB) module stores the gathered and inferred context information and provides query interface to the Service Trigger (ST) module. Service Trigger (ST) module has two functionalities, personalization of the established services according to the different context information, and discovering and setting up a personalized service for users according to the different contexts. The ST module communicates dynamically with the CDB module to monitor the context information before triggering the services, and communicates with the Privacy Protection (PP) module to verify if the services can use the context information or there are privacy constraints. Privacy Protection (PP) module controls what data might be published, through verifying if the "ready to activate" services are authorized to access the required user context information or a part of it considering different privacy levels. The Context-aware User Equipment subsystem consists of a Client Context Acquisition module and Local Service Management module. Client Context Acquisition (CCA) module discovers the context sources in the local sphere and collects the raw context information about user, device and environment. Sensors are the frequently used context sources which can be present in the user sphere, in the environment or in the device and retrieve context information from them. Different context information can be derived from these sensors, such as noises, lighting, proximity, user's location, etc. These sensors detect the values and report this value to the CCA which then represents the received information in the predefined XML format and forwards it to the CAM module located in the CAS. Local Service Management module controls and manages the local services execution through monitoring the CCA module and dynamically comparing the context with its stored rules in order to activate the corresponding service in a personalized manner. The application server consists of two modules, the Service Context Acquisition and the Media Delivery Context Acquisition modules. The Service Context Acquisition (SCA) module collects the service context information and sends it to the CAM. Most service related context information is contained in the Electronic Program Guide (EPG). The SCA collects the EPG from the IPTV application or from the internet, and retrieves the information about title of the channel, description, starting

time, ending time and other information like categories. SCA then represents the information in an XML format and forwards it to the CAM. Media Delivery Context Acquisition (MDCA) module monitors the content delivery and dynamically acquires the network context information during the content delivery and sends it to the CAM. This information reflects the state of the network such as packet loss, jitter, and round-trip delay. In the network domain, a Network Context Acquisition (NCA) module is responsible for collecting the bandwidth information before each service session establishment and sends the acquired information to the CAM. It is noteworthy to mention that UP-TO-US project offers different implementations for the previously described architecture for deployment in Internet Multimedia Subsystem (IMS) environment and in non-IMS environments or New Generation Network (NGN) environments. Performance metrics used in UP-TO-US project rely on three metrics. The first metric is the delay of the personalized content selection (DPS) - that is the performance of the recommendation engine itself. The second metric is the delay in service initiation (DSI). The third metric is the EPG Browsing Time (EBT) which measures the quality of the experience by users in finding the relevant content and they offer a formula to calculate this metric based on the precision probability of the recommendation engine and the estimated time the user takes to judge on whether he likes that content or not.

Other efforts were conducted in the field of context-aware multimedia. The research in [14] proposes a design that focuses on augmenting Internet Multimedia Subsystem (IMS) to make it context aware. It provides an interface for sensor networks to report data to be included in the computing context. According to the design in [14] the architecture is composed of Pervasive Services Management (PSM) subsystem which is in turn composed of Service Delivery Manager (SDM), Composition Manager (CoM), and Deployment Manager (DeM) and is responsible for managing the discovery, filtering, composition, deployment and lifecycle of services. It also has modules such as Preference Manager (PM), Preference Condition Monitor (PCM) and Learning Manager (LeM). It has a Context Manager (CM) that manages the collection and storage of context information on any entity, including services and users. Other components include Identity Manager (IdM), Location Manager (LM), Coordination Engine (CE), Multimedia Service Provisioning Broker (MMSPB) and Media Resource Manager (MRM). The research in [15] proposes an architecture for adaptive IPTV services. The purpose of the proposed architecture is to introduce new functionalities in IPTV over IMS which optimize satisfaction of the end-user and resource utilization of the operator's networks. It uses a context sensitive user profile model to deliver IPTV streams adapted to the user's environment. The research proposes a novel IMS-compatible user-centric network

management solution that employs user profile management and adaptive techniques for IPTV services in order to (a) compensate network impairments according to the time varying conditions of the network delivery chain, (b) perform a content dependent optimization of the encoding and/or streaming parameters and (c) improve the end user experience/satisfaction by maximizing the delivered quality of service level and delivering content adapted to the end-user environment. The research in [16] proposes another architecture for a framework for interactive personalized IPTV for entertainment. The architecture in [16] proposes content selection and adaptation based on delivery conditions, viewer's interest and type of devices used. They split their architecture into a server side and a client side architecture. The server side holds the different data models such as viewer profile, concept model, quality of experience model (QoE) and adaptation engine. Viewer profile is composed of interests, demographics, history of viewed content, access device, and network connectivity. The concept model is a hierarchy for organizing multimedia content. The concept model organizes content in four levels: abstract concepts, topics, multimedia items, different quality versions. The QoE model makes decisions based on viewer's profile and concept hierarchy to provide personalized content suggestions. The adaptation engine controls the adaptation of the streamed multimedia content based on QoE suggestions and client feedback. The client side consists of a viewer observer, network monitor, device detector, and client feedback unit. The viewer observer acquires data such as demographical data, subjective preferences, etc. and monitors the viewer behaviour such as content selection, play, abort, etc. The network monitor observes the network performance related parameters that include delay, jitter, and loss and describes the status of the transmission medium. The device detector detects the characteristics of the used device. The client feedback unit collects data from the viewer observer, network monitor and device detector, computes feedback grades and regularly sends them to the server. The work in [17] offer an implementation for IPTV session mobility. The purpose of this implementation is to transfer and retrieve an active media session to one device, the ability of a session to be split across multiple devices, and ability to transfer supplementary services to the destination device along with the IPTV media. To measure their performance, they measure transfer delay, and media disruption. The below diagram shows the implementation architecture used in [17].

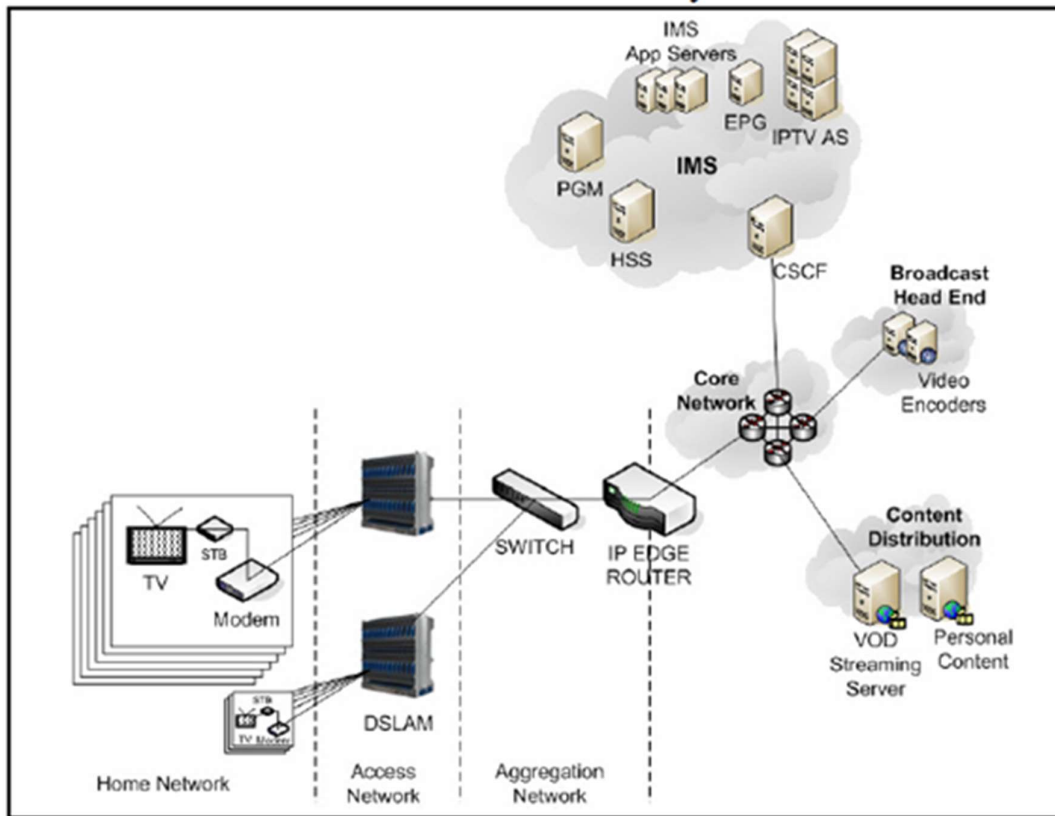


Figure 3. IPTV implementation architecture using IMS [17]

The research in [18] offers a conceptual framework for applying semantic web services to provide context-aware multimedia. They propose a semantic web services based framework which abstracts from both the annotation schemes and vocabularies and the available software interfaces - such as web services. The following diagram summarizes the resulting concept framework.

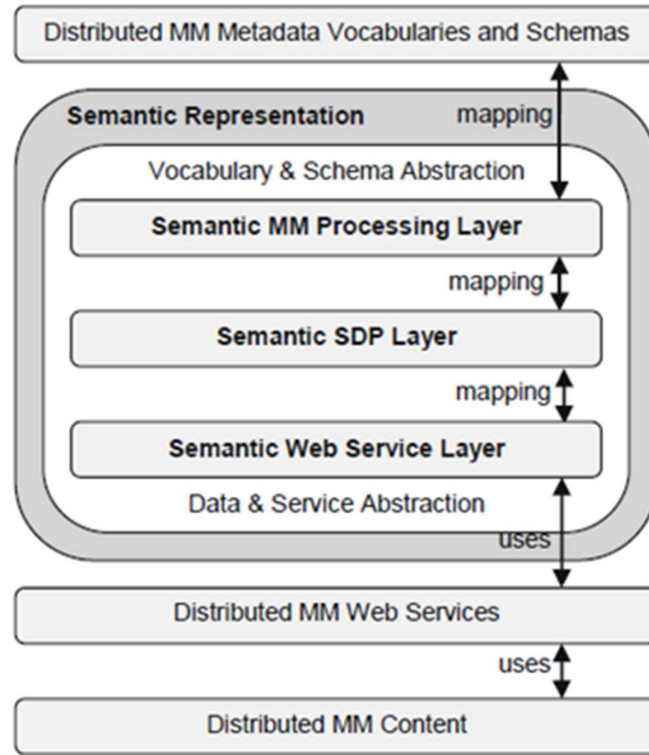


Figure 4. Resulting conceptual framework from [18]

The following tables summarizes the different contextual parameters used in the different systems described in context-aware multimedia.

Table 2. Comparison of context information in context aware multimedia

Feature/Paper	[8]	[9]	[10]	[13]	[14]
Type	Paper	Master Thesis	Paper	Paper	Paper
Target	Content	Content	Content	Content	Content
Context Information					
User Context					
Age	Yes	Yes	Yes	Yes	Identifies Roles managers for gathering different context
Identity	Yes	Yes	Yes	Yes	
Profile/Preferences		Yes	Yes	Yes	
Location	Yes	Yes	Yes	Yes	

Granularity of location	Region	Room		Region	information
User Activities	Yes	Yes		Yes	
Emotions					
Needs					
Agenda	Yes	Yes	Yes	Yes	
Consumption History	Yes	Yes	Yes	Yes	
Gender					
Purchase History					
Service Context					
Current Program					
Brand Information					
Promotion					
Content Language	Yes	Yes	Yes	Yes	
Content Format	Yes	Yes	Yes	Yes	
Content Description	Yes	Yes	Yes	Yes	
Access rights		Yes		Yes	
Location		Yes		Yes	
Current Content Viewership Rating/Actual Viewership					
Computing Context					
Network and Bandwidth	Yes	Yes	Yes	Yes	
Device Type	Yes	Yes	Yes	Yes	
Device Status	Yes		Yes		
Operating System					
Physical Context					
Surrounding Environment					

Nearby Objects					
Weather					
History Context					
Time					
Privacy	Yes	Yes		Yes	
Session Mobility					

Feature/Paper	[15]	[16]	[17]	[18]
Type	Paper	Paper	Paper	Paper
Target	Content	Content	Content	Content
Context Information				
User Context				
Age		Yes	Yes	
Identity				
Profile/Preferences		Yes	Yes (Group/Individual)	
Location			Yes	
Granularity of location			Region	
User Activities				
Emotions				
Needs				
Agenda				
Consumption History		Yes		

Gender			Yes	
Purchase History				
Service Context				
Current Program				
Brand Information				
Promotion				
Content Language				
Content Format				
Content Description		Yes		
Access rights				
Location				
Current Content Viewership Rating/Actual Viewership				
Computing Context				
Network and Bandwidth	Yes	Yes		
Device Type		Yes		
Device Status				
Operating System				
Physical Context				
Surrounding Environment				
Nearby Objects			Yes (Devices)	
Weather				
History Context				
Time				
Privacy			Authenticates Devices	

Session Mobility			Yes	

2.3. User Modeling

One of the sub-problems of this research is how to model users to be able to serve them better content and a few works focused only on this problem. Sample of these works are presented in [19-20]. [19] offers a design research approach for user modeling in the field of personalized mobile advertising. They discuss that the buyer's decision is influenced by several factors including the buyer's individual characteristics, the environment, and the merchant's marketing strategy components such as price and promotion. Context information covers aspects such as location, time, user activities, and weather. The authors of this paper use customers' user demographics (e.g., age), user preferences (e.g., preferred ambience), context (e.g., weather and location), and content (e.g., brand) information. Their approach in brief is to survey users then use the survey data to set prior probabilities on Bayesian network. Their survey was performed on 200-300 samples. The following diagram shows the proposed dimensions of personalization in a restaurant recommendation system.

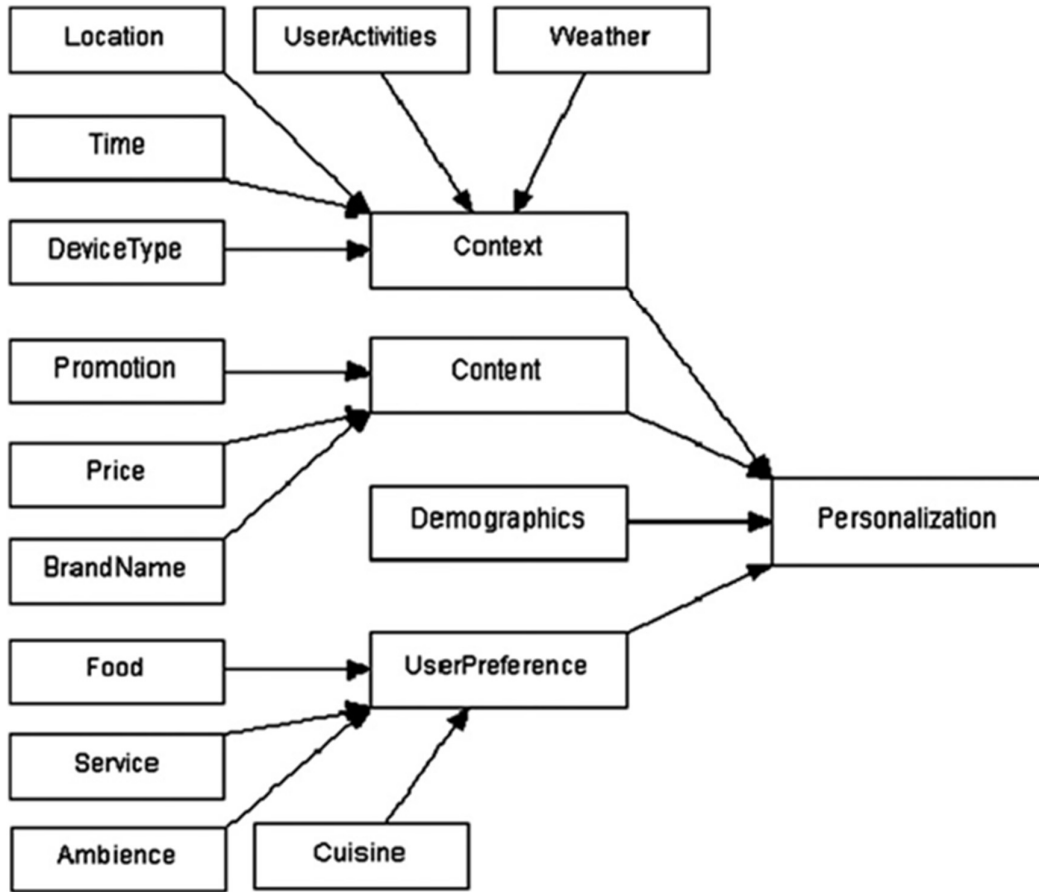


Figure 5. User modeling in [19]

The research in [20] offer a flexible user profile management for context-aware ubiquitous environments. They define context as any information that can be used to characterize the situation of an entity. The idea is to offer a middleware that offers context to multiple applications. The context information used in [20] is grouped into user profile, device profile (for an array of devices), network profile, service profile and context profile (consisting of the dynamic / volatile data). User profile contains basic information such as age, gender, address, phone, gender, profession, etc. and also user disabilities such as color perception and hearing impairments. Device profile is defined for each of an array of devices and each device is described by its hardware information: CPU speed, capacity, battery life, related peripherals, software information such as operating system name, version and vendor plus the device brand and serial number. The network profile is defined by the network type, medium of transmission, operator, service level agreement, etc. The service profile is defined by its name, version, protocols supported, ports, multimedia content, database files, billing, etc. The context profile groups the volatile data such as time, date, location, the service device and network, running

applications and perceived quality of service values. The below diagram shows the middleware architecture proposed by [20].

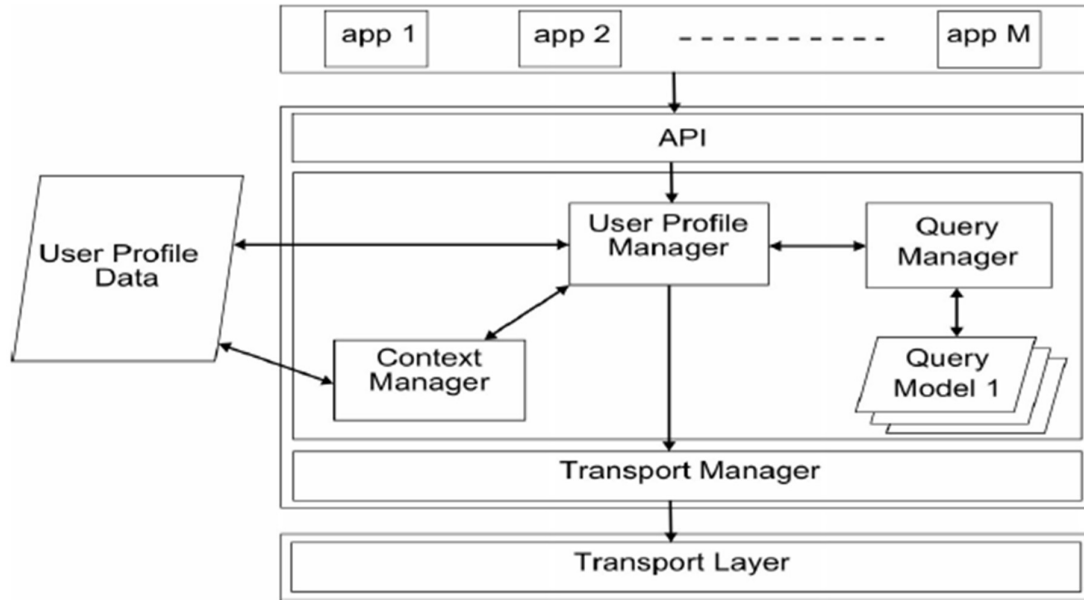


Figure 6. Middleware architecture proposed by [20]

2.4. Recommendation Engines

Another sub-problem is the choice of recommendation engine to use and several efforts were conducted in this field. A background on the different types of recommendation engines can be found in [36]. According to the author in [36] there are three main categories of recommendation engines. The first category pertains to calculating recommendations based on similarities between users and this is known as collaborative filtering technique. The second category relies on finding similarities between products (or items such as links, books, movies, etc.) and this is sometimes referred to as product filtering. The third category is called item-based filtering and it basically capitalizes on the efficiencies of product based filtering to overcome the lack of scalability issues in collaborative filtering. The basic concept is to build a similarity matrix between products or items then, to recommend a new item to a user, find the most similar items to the top rated items by the user. This narrows down the search scope significantly as well as allows for the precomputation of the similarity matrix due to the infrequent changes in products and relatively constant scale. Users are expected to grow much faster than the offered products themselves; therefore, it is more efficient to loop over products

then loop over users. After getting the most similar new items similar to the top rated items by the user, a rating score is given to each new item and then the top scores are used to identify the new products to recommend. There are different methodologies of calculating similarity measures. The simplest form is to calculate the Euclidean distances between different data points (the less, the more similar). To convert this metric to be consistent with the semantic of similarity (the more identifying the more similar), and inverse operation is performed on the Euclidean distance measure after adding one to it to avoid division by zero. Other similarity measurement techniques rely on different correlation measures such as Pearson Correlation which assumes a linear relationship between variables. It is worth mentioning that Pearson Correlation has an advantage over Euclidean distance measures as it does not require data normalization. Other distance measures can be useful in different situations such as Manhattan distance, cosine distance and others. The rating score is usually calculated after quantifying preference values. For the top similar data, the preference value in concern (based on the type of filtering technique) is weighted by the similarity score. To account for items that may have more ratings than others, the sum of the weighted ratings is normalized by the sum of similarity measures. The patent in [21] shows a system and method for recommending multimedia content to users. The purpose of the recommendation engine is to recommend content to users and recommend users to new content entering the database. The authors in [21] discuss a shortcoming in recommendation systems that rely on finding similarities between users because such systems are usually over specialized and users do not usually provide enough information. They discuss that collaborative filtering approach, which relies on similarities between users, requires a high number of active users and requires enough users to rate new content to start recommending new items entering the database. The invention described in the patent uses a hybrid approach. First it utilizes content based recommendation which relies on the user profile - more specifically user preferences - in order to find titles matching the user's preferences. The invention also uses case based recommendation which is based on recommending titles similar to those already seen and positively rated by the user. In addition, they include Bayesian recommendation which calculates, for each title, the probability of it to be preferred by the user. Then the titles with highest probabilities are recommended to the user. Finally, a combination of the recommendation list is performed from the previously mentioned recommendation methods. The step of combining those lists is based on a weighted combination of information related to the user and success of the previously recommended titles by each approach. These weights basically represent the confidence of each approach. According to [21], the utilized recommendation methods are capable of handling the arrival of

new content. The research in [22] propose a context-aware decision engine for content adaptation. Its purpose is to develop a system that is quality of service aware and targets mobile users. The prototype developed was for PDF content type but can be extended for other content types. The quality domains included color, download time, output format, and others. From an architectural perspective, the authors in [22] mention that they pre-calculate decisions for rarely changed preferences and compute only changing parameters in real time to adjust for performance. The research in [4] - previously described and therefore will not be discussed in details here- also proposes a recommendation model that is an enhancement over the collaborative filtering approach. The main contribution in [4] is they were able to quantify similarities between different contexts using genetic algorithms. Therefore, instead of relying on Pearson's coefficients alone to identify user similarities, they use the similarities between contexts produced by the genetic algorithm to adjust the Pearson coefficients between users. [11] describes the details of the recommendation engine used in UP-TO-US project discussed previously. The authors in [11] divide the context information into two groups: conditional context and situational context. Conditional context has the context information that decides whether the contents are accessible or allowed to be accessed by the user before doing the content recommendation such as user's age or lacking network capacity. Situational context is the context information that influences the user's preferences regarding two aspects: situational context (likes to watch news at 4pm) and different influence levels of different context types (it is 4pm but user is in the salon and likes to watch movies in the salon). The authors in [11] list the formulas they use to calculate the preference degree for each content feature at each content value and how they calculate the influence degree of a certain context value as well as the updating formulas for these values. Due to the realization of the performance penalty of doing all calculations in runtime, the architecture in [11] performs pre-calculations related to static content filtering to enhance the performance of the system. They also pre-calculate the preference degrees for each content feature at each content value once per day and the online phase only computes the context influence and recommendation list. In case a group of users are consuming content, the values are replaced with the average values for all users. The following diagram shows the recommendation system architecture used in UP-TO-US project.

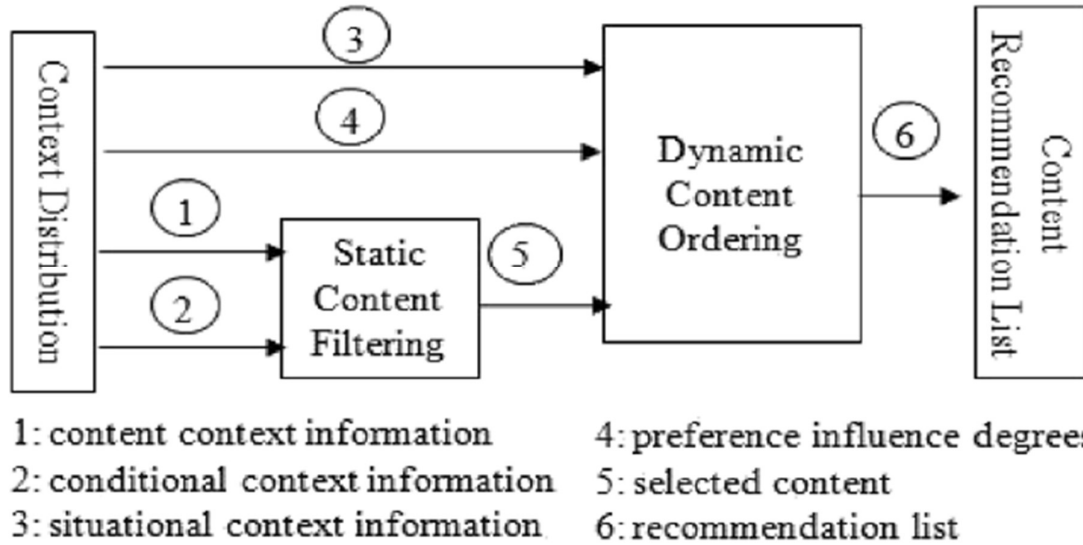


Figure 7. Recommendation engine architecture in UP-TO-US project

[23] offers a free recommender system library called MyMediaLite that is implemented in C# programming language. The library offers features such as rating prediction and item prediction. Below is a table comparison of MyMediaLite with parallel libraries.

Library	Duine	MultiLens	LensKit	Mahout	GraphLab	SUGGEST	MyMediaLite
version	4.0.0-RC1	–	0.04	0.4	v1_134	1.0	1.02
date	2009-02-17	2004-10-22	2011-07-29	2010-10-31	2011-07-29	2000-11-08	2011-08-03
actively developed	–	–	✓	✓	✓	–	✓
license	LGPL 3	GPL	LGPL 2	Apache 2.0	Apache 2.0	non-free	GPL 3
language	Java	Java	Java	Java	C++	C	C#
scalable	–	✓	✓	✓	✓	✓	✓
matrix factorization	–	–	✓	✓	✓	–	✓
rating prediction	✓	✓	✓	✓	✓	–	✓
implicit feedback	–	–	–	–	–	✓	✓
online updates	✓	–	–	✓	–	–	✓

Figure 8. Comparison of recommender system libraries adapted from [23]

[24] provides a design and implementation of user context-aware recommendation engine for mobile. Their system uses Bayesian network, fuzzy logic and rule base. The purpose of this engine is to recommend services for adaptation according to the user's current context socially and personally. Their goal of their research work is to socialize and personalize mobile. Example services provided by the recommendation engine:

- Provide the callers with the ability to communicate the high priority calls irrespective of his situation and location
- It goes to silent mode in the classroom/meeting room automatically
- It goes to the vibrating mode automatically in the library and also provides services like book search
- It provides notifications whenever required; and

- It provides context based desktop applications.

The process used by [24] can be described as taking the input from sensors which is fuzzified into linguistic terms. Fuzzified sensors are aggregated to context and stored as a vector. Then the context is compared with a set of rules and based on the matched rule, a set of actions are recommended. The research in [25] attempts to build a recommendation engine based on a psychological model. The authors propose a method to apply user characteristics to the content recommendation based on the consumption pattern derived from the user's behavior pattern. The proposed recommendation method adopts and applies the DISC model which is verified in psychology field for classifying user's behavior pattern. They then apply a decision tree on context to recommend genre and the selection of the specific content is based on the preferred attribute for each personality type. The following diagram shows an example of the schematic of the recommendation of the user's preferred multimedia content as proposed by [25].

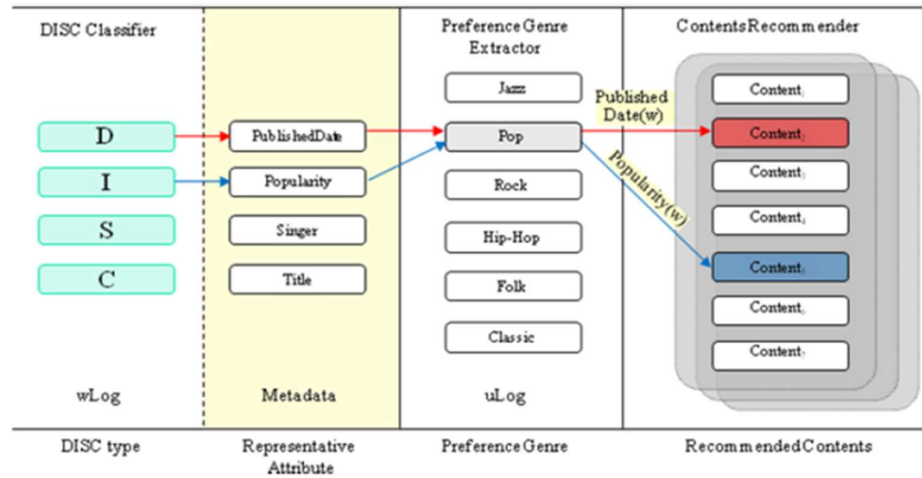


Figure 9. Adopting DISC psychological model for content recommendation [25]

2.4.1. Context-Aware Recommendation Systems (CARS)

Various efforts have been conducted to incorporate context into recommendation systems. According to [43,46,54], there are 3 different algorithmic paradigms to incorporate context into a recommendation engine and 3 methodologies for collecting context.

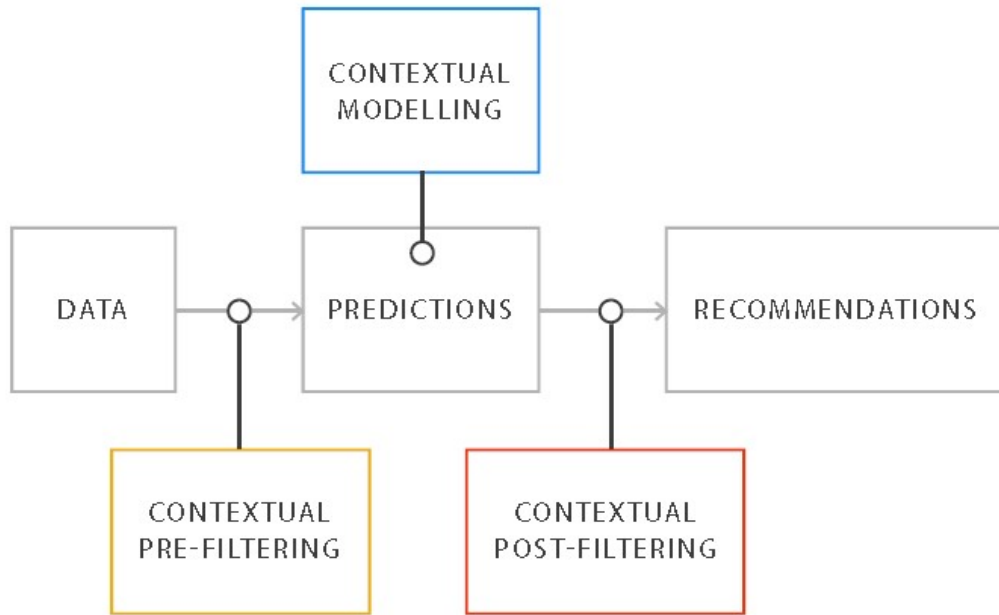


Figure 10. Paradigms of incorporating context into a recommendation engine adapted from [54]

The paradigms for including context within a recommendation engine (summarized in the above figure adapted from [54]) fall under one of the following umbrellas: pre-filtering, post-filtering, or context modelling. Pre-filtering is mainly employed to remove any content that is irrelevant to the current context. An example for our scenario is to remove gender targeted ads when the viewers are only of the opposite gender or showing ads that target children when only an adult is watching. Pre-filtering reduces the search space for the recommendation engine and, therefore, has a positive impact on performance. An example of pre-filtering approach is shown in [11] where they perform static filtering based on conditional context. The second paradigm for incorporating context into the recommendation engine is through post-filtering technique. Post-filtering is applied in mainly two situations. In case a system already has a legacy recommendation engine that needs to be extended to include context information, post-filtering is applied to filter out the irrelevant recommendations based on the current context. Post-filtering is also applied when the context information does not affect the relevance of the content in general, but rather affects the preference degree of the user to the recommended content. In this case, post-filtering is applied to re-order the recommendation list based on the preferences of the user given the current context. An example of post-filtering approach is also shown in [11] where the authors use situational context to re-sort the list of recommendations based on the user

preferences for the current context. For example, user is presented with recommendations action movies and news, but it is 4pm and user likes to watch news at 4pm. This context information should affect the order of the recommendations by presenting the news content first. The third approach for adding context awareness to recommendation engines is through modelling contexts within the recommendation engine itself. This approach mainly relies on computing similarities between contexts and weighing the Pearson coefficients between users by context similarity. Such technique is employed in [45] where the authors compute the cosine similarity between contexts then use the output as weights into the Pearson similarity between users. The output is used to identify neighborhood set of users to be consulted for recommendations. Another approach is shown in [4] which uses genetic algorithm to compute similarities between contexts and then use the output context similarities as weights for Pearson coefficients between users. The authors in [48] attempt to find latent user preferences for contexts, but instead of using the output for post-filtering as in [11], they use the latent semantic analysis to model context within the recommendation engine itself. The work in [48] tries to build a model that reflects user preferences towards certain contexts and of a given item to a context.

Methodology for gathering context has also been studied in various works [43, 54]. These can be categorized as explicit, implicit or inferred. Explicit context gathering techniques require users to specify the context information (their needs, emotions, who are they watching with, etc.). Such approach is invasive to the user experience and is kept to a minimum such as asking users to rate items. Implicit context gathering techniques use various sensors to gather context such as GPS device for location, contacting a weather service to get weather info, contacting a time service, and others. These techniques are not invasive to the user experience. The third method of context gathering is to infer context information such as estimating the current task or activity being performed by the user. This technique uses sensors to gather raw context data and attempts to analyze the raw data to produce higher level context information. For example, the location sensor can be used to infer user's district/region level location. Location sensor can also be used to infer relationships between users by proximity.

Finally, the introduction of context modelling within recommendation engines introduces a performance penalty and increases sparsity (which may lead to a reduction in recommendation precision) [49,51,52]. This is due to the fact that instead of performing computations on a users x items matrix, computations need to be performed on users x items x context matrix and each dimension can be represented by a vector in itself. The authors in

[49] attempt to reduce the execution time of recommendation engines by clustering users into segments and computing user similarities within the same segment. They also propose a solution to the cold-start problem by assigning new users to a cluster based on demographic attributes and offering the top recommended items by the users in the assigned cluster. [51] and [52] offer a solution to the high dimensionality of context aware recommendation engines by proposing dimensionality reduction techniques that also help in reducing sparsity. The authors experiment with PCA algorithm and unsupervised neural network (as an auto-encoder) to infer latent context.

2.5. Targeting Advertisements

In this section, the methodologies of targeting advertisements will be covered. This has direct influence on the way of engineering the recommendation engine itself. Since this is a multidisciplinary field, studies from the field of advertising, marketing and computer science will be included in this study. Since advertising is an industrial field, latest trends and industry practices will be taken into consideration as well. In general, targeting advertisements is aimed at enhancing four measures: attitude toward the ad, attitude toward the brand, intention to click (for online advertisements), and purchase intention.

There are different methodologies to target advertisements based on the nature of the context information at hand. This resulted in different types of targeting in the online advertising ecosystem. According to [26] the following are the different types of targeting in the realm of online (also known as digital) advertising:

1. Demographic targeting: based on user information such as age, gender, etc.
2. Geotargeting: targets users based on their location data.
3. Behavioral targeting: tracks user's actions and tries to capture patterns and trends.
4. Contextual targeting: retrieves the advertisement most relevant to the content being consumed.
5. Site-targeting: similar to contextual targeting but matches the ad against the theme or genre of the publisher. For example, a mobile manufacturer may choose to publish ads on consumer electronics sites.
6. Day-part targeting: focuses on people's work/life schedules.
7. Purchase-based targeting: similar to behavioral targeting but focuses on the purchase history of the user.

8. Retargeting: focuses on users who dropped a transaction without completing it. For example, user who expressed interest in buying a product but did not check it out. Retargeting is used to serve these users new advertisements in hope that they will complete the purchase.

In terms of the impact of the above targeting method a research shows that 65% of online shoppers pay more attention to behaviorally targeted advertisements and 39% say they pay more attention to contextual advertisements. Despite the rising interest in behavioral targeting by advertisers and agencies, it faces some opposition. Over 60% of all age groups reject tailored advertising activities due to the tracking activities required to achieve behavioral targeting [26]. Since the main concern is a privacy concern, our proposed system will address this issue on several fronts. First, all context gathering activities will be performed on the client device itself and, therefore, no images or specific location data will be sent to the back-end or stored on the server. Second, all communications with supporting cloud services will be encrypted using SSL. Third, the user will remain in control by denying access to social data information or devices (camera or location services), which will result in an inferior service in terms of ad recommendation. The article in [27] surveys the different targeting techniques for advertisements in the realm of TV and Set Top Boxes. In the realm of TV and Set Top Boxes, the aim is to identify the program(s) whose audience profile fit the target profile for the product being advertised; therefore, identifying the programs to buy advertisements in. When Set Top Boxes data is not available, surveys are conducted on sample audience to identify which programs and channels they watch in a given day or week. The result of the survey is a Nielsen rating of programs that identify viewership of each program as well as the demographic profile of the audience watching that program. Now that over 90% of US households have Set Top Boxes, the targeting problem can be treated as a supervised learning problem - that is to maximize the buyers per impression reached. The following are the different targeting methods used for Set Top Boxes. Again, the aim is to identify which programs to buy advertisements in.

1. Direct Buyer targeting: this is a simple three step process of identifying buyers, find out what TV programs buyers are watching, then target those programs watched by buyers. This method has the advantage of being highly predictive of the programs to buy in the future. It also has the disadvantage that the probability of detecting buyers is small and works for programs with very high impressions (over 1 million impressions to detect 14 buyers).

2. High Dimensional Demographic targeting: also starts with identifying buyers, then aggregate this data into 3000 variable Buyer Demographic target, then looks for TV programs matching the demographics of the target profile across the 3000 variables. It achieves the same results as direct buyer targeting but with better probability of detecting buyers.
3. Nielsen TRPs: “Nielsen TRPs are similar in concept to the High Dimensional Match method, however there are several differences including (a) size of panel, (b) number of demographics, and (c) match function.”
4. Phone response: In case the advertisement can include a phone number, utilize phone response data to track and optimize the performance of advertisements. This mechanism has the capability of tracking which stations, programs, time of day that generate the most phone calls but it is limited because not all advertisements can utilize this mechanism.

All the four methods mentioned have their usefulness based on the size of the airing (viewership) of programs. The article also includes studies showing that consumers favor free TV content with ads, but they just want fewer ads [27].

Other efforts have been conducted to enhance the targeting methods of ads. The patent in [28] focuses pushing advertisements to mobile users by recommending the most suitable category of advertisement given the time of day. The patent in [29] submitted by Facebook focuses on augmenting social networking data with user activities in a third-party system to recommend advertisements. [30] is a patent by Google which includes a method for serving relevant advertisements based on contextual targeting described above.

The article in [31] explains one of the models that can be used to measure advertising effectiveness. The model is used by Nielsen and is called the 3Rs which stand for Reach, Resonance and Reaction. Reach measures the media performance and basically answers the question of whether you got what you paid for? Resonance indicates if the right message got to the right person. Reaction measure if there was enough resonance to create a business outcome.

The paper in [32] proposes a system for enhancing the targeting of advertisements in social media by the introduction of a social endorsement. The idea is to identify customers who talked positively about a certain advertisement by means of sentiment analysis then identifying the target consumer who should see the ad. The study in [33] studies the impact of advertising targeting and advertising avoidance. It realizes that increased precision in targeting will

increase returns up to a point where users will not want to share more information for higher targeting precision, in which case, a higher precision leads to lower returns.

2.6. Quality of Service and Quality of Experience

Other important aspects to study in this research are the Quality of Service (QoS) and the Quality of Experience (QoE). QoS measures the throughput of the system, such as delay in calculating the recommended advertisements, network delays, jitter, packet loss, turn around time, etc. QoE measures other aspects as perceived by the user such as the precision of the recommendation engine or smoothness of video streaming. QoE is sometime referred to as Perceived Quality of Service (PQoS). We noticed that some researchers refer to precision as accuracy, such as in UP-TO-US project. Precision is the number of correct predictions for a specific class divided by the total number of predictions for this class, while accuracy is the total number of correct predictions divided by the total number of samples. UP-TO-US project discussed in context-aware multimedia uses three metrics to measure their performance [10]. The first two metrics are related to the quality of service and performance of the system. The first metric calculates the delay of personalized content selection and the second metric measures the delay in service initiation. The third metric measures the EPG browsing time which pertains more to the quality of experience. The estimated EPG browsing time depends on the precision probability of the recommendation engine. Precision probability is given by checking the recommended content that user expressed interest in divided by the total content recommendations delivered to that user. The following is the equation for probability precision.

$$Pa = \frac{\text{recommended} \cap \text{interested}}{\text{recommended}}$$

Figure 11. Probability precision used in UP-TO-US project

Given the probability precision, the estimated EPG browsing time can be calculated by assuming a time constant that users consume before judging content and deciding whether to view it or to switch the channel. This time constant can be multiplied by the number of recommended content the user disliked with probability $(1-Pa)^i$ where i represents the number of skipped content. The following is a complete formula on the estimated EPG browsing time given a total number of programs n .

$$\begin{aligned}
E(EBT) &= \sum_{i=1}^n (1 - p_a)^{(i-1)} p_a (i - 1) t' \\
&= \left((1 - p_a) \frac{1 - (1 - p_a)^{n-1}}{p_a} + (1 - n)(1 - p_a)^n \right) t'
\end{aligned}$$

Figure 12. Estimated Browsing time employed in UP-TO-US project

In simple terms the above summation operation can be explained as the time spent judging and skipping content before content i “ $(i-1) t'$ ” weighted by the probability of skipping all content before content i $(1-p_a)^{(i-1)}$ and the probability of expressing interest in content i (p_a) . The paper in [34] proposes a QoE-aware Internet Multimedia Subsystem (IMS) infrastructure for multimedia services. They monitor the PQoS at the terminal and define two thresholds, below which a yellow or red alarm is triggered to the Multimedia Content Management System component. The main input received from the terminal is the packet loss ratio. The thresholds for PQoS in relation to packet loss ratio are gathered using personal experiments and interviews. The authors in [35] discuss bandwidth allocation optimization for IPTV. They divide the video QoE errors into three types of errors, Edge errors, Color errors and Jerkiness errors. Edge errors are measured by detecting blurring images, edge business or block distortion. Color errors are measured by hue and saturation. Jerkiness error means that the same frame stalls for a while before proceeding to the next frame (freezed frame). The authors discuss three ways of measuring those errors. The first is the Frame Reference which uses the original frame for comparisons. The second is Reduced Reference which extracts features from the image before and after processing and performs comparison on the extracted feature. The third is No Reference which uses only processed image for evaluation and is used to measure jerkiness. The following table shows the different error types and measurement techniques as presented in [35].

Blurring	RR	Comparing edge standard deviation between OS and PS	Encoding/ Decoding
Block distortion	RR	Estimation using edge of vertical/horizontal direction and weight of around pixels	Encoding/ Decoding/ Network Transport
Color error	RR	Comparing hue and saturation between OS and PS	Encoding/ Decoding/ Network Transport
Edge busyness	RR	Comparing edge average values between OS and PS	Network Transport
Jerkiness	NR	Freezing frame extraction using edge difference between frames	Network Transport

Figure 13. Video quality indicators adapted from [35]

2.7. Quality of Context (QoC)

Due to the fact that implicit context gathering techniques rely on devices and sensors (such as location sensor) that have varying degrees of accuracy, there is a need to model and manage quality of context to better enable pervasive systems make the right decisions. According to [63], the term QoC was first coined by [57] back in 2003 and is referenced by a few works afterwards concerning the same problem domain. The authors in [57] defines QoC as “any information that describes the quality of information that is used as context information”. The quality of context information severely influences the capability of context-aware applications to adapt and the lack of information on the quality of context hinders pervasive applications from having robust performance in real-life scenarios [58].

The author in [57] primitively lists QoC parameters as the probability of correctness, trustworthiness, resolution, and up-to-dateness. Following works in the same field contribute to enriching the classification and categories of QoC parameters as well as standardizing their definition, such as [58] which realizes a confusion in the used terms and borrows from engineering and industry for instrumentation measuring to resolve the confusion. Research in the domain of QoC focuses on categorizing and defining QoC parameters, QoC sources and propose models and frameworks for designing context-aware applications that take quality of context into consideration [58-63]. Fast-forward 13 years later, in 2016, the authors in [58] present an MLContext extension for modeling QoC. In the process, they perform a classification of quality parameters and define measures to quantify them. The authors in [58] scope the Quality of Context to the quality of information that is used as context information, not to processes or devices that provide that information. Quality is simply fitness for use which may depend on various factors such as accuracy, timeliness, relevancy or precision. Defining quality as “fitness for use” makes it relative to the application domain and the degree of tolerance it can handle. For example, if an application relies on the exact user location for navigation requires a higher degree of precision and accuracy than an application that only cares about the region the user is in. The former will need access to GPS signal while the latter may only depend on 3G or 4G cellular antenna location. Quality parameters have been classified into 3 main classes by [58], mainly data acquisition, data representation, and data usage.

- Data acquisition parameters are quality parameters directly related to the sensors that gathered the information. This class includes parameters such as resolution, precision, accuracy, freshness, etc. Resolution is the fineness to which an instrument can be read or the smallest change in the underlying physical quantity that produces a response in the measurement. Precision is the degree to which repeated measurements under unchanged conditions show the same result. The accuracy of the sensor is the maximum difference that will exist between the actual value and the indicated value at the output of the sensor. Freshness refers to how recent the provided information is at the time of delivery. We must be aware that some information remains valid over time, while other information may become discredited or obsolete.
- Data representation parameters pertains to the specification of acceptable formats and units by the application as well as the understandability of the context information by the application.

- Data usage parameters are the parameters that define the context of the data acquisition itself to decide if the data has significance. Quality parameters such as trustworthiness, completeness, relevance, availability, etc. fall under this class of parameters. Trustworthiness is the extent to which information is regarded as true and credible. Depending on the application, context information that are relevant to the situation are chosen.

A summary of the classes of quality parameters is shown in the table below which is adapted from [58].

Classification of quality parameters.

Data acquisition	resolution, precision, accuracy, range, freshness, location, coverage
Data representation	units, format, understandability, aliases
Data usage	trustworthiness, completeness, relevance, comparability, availability

Figure 14. Classification of quality parameters adapted from [58]

In addition to the above categorization of quality parameters, there exists dependencies between those parameters. For example, trustworthiness parameter depends on precision, accuracy and freshness. If a context information is not consistent, not representing the actual state or has stale information it will not be fit for use by a context-aware application. Similarly, completeness relies on all required parameters by the application being available. The identification of which quality parameters to be chosen and which level of quality is required is determined by the specific application requirements. [58] continues to show the proposed MLContext to model and generate QoC-Aware and Context-Aware applications. Due to the structured and exhaustive approach used by the authors in [58], their work will be used as a pivot in comparing the other systems discussed in this section.

Between [57] and [58] multiple efforts have been developed to create modelling frameworks and applications that associate QoC with context information. The work in [59] aim at building ubiquitous QoC-aware applications through model-driven software engineering. They propose a generic and extensible design process for context-aware applications taking into account the quality of context. They also define QoC as “any information describing the quality of information that is used as context.” QoC for them is represented by a set of parameters such as accuracy, probability of correctness, trustworthiness, resolution or freshness. They also define four types of imperfection of context information. The first is when the context information is unknown or incomplete, which may hinder the application from adapting. If this is an expected scenario, then context-aware applications may have to

deal with lack of context information either by not producing a decision at all, produce a decision with the information available in case of partial incompleteness, or downgrade the service to an inferior level. The second is when the context information is ambiguous as there is a risk of having contradictory information from different context sources. In such cases the application may ignore this information or rely on the source that is more “trustworthy”. The third is when the context information is imprecise. The fourth is when the context information is erroneous and does not exactly represent the real state. In such scenarios, application developers will either have to search for an alternative source of context that is more accurate or search for alternative pieces of context-information as a substitute. Other efforts such as [60] focus on enhancing QoC awareness for IoT (Internet of Things) systems. Their efforts stem from the underlying assumption that context data are known to be imperfect and uncertain by nature. One way to limit this uncertainty is to introduce more knowledge associated with context data such as metadata describing the QoC of the context data. The authors propose adding QoC-based filtering and attribute-based privacy policy to enhance the distribution of context data for IoT systems. The QoC parameters used in their work are freshness, precision and correctness. In an earlier work by the same authors [61], they offer a dedicated Quality of Context Information Model (QoCIM) metamodel, which offers a unified solution to model heterogeneous meta-data about QoC. QoCIM facilitates exploiting and manipulating criteria in an expressive, computable and generic way. In addition, they discuss different perspectives of context management and divide it into two elements: context collection and context processing. A context collector is a software entity dealing with the acquisition of raw context data (data that have not been processed or transformed) [60]. The context collector uses the QoCIM to associate QoC metadata to the raw context data. “A context processing capsule is a functional element that performs the processing of context information into information of a higher level of abstraction. It is a consuming and producing entity. Several categories of context data manipulation can be operated by a capsule: aggregation, filtering, fusion, inference” [60]. According to the same authors, context management operations also includes analysis of the impacts on the management of QoC metadata during different manipulations such as adding and retrieving QoC parameters, updating the value of a parameter, or filtering on the value of a parameter. Then the exchange of context data is enhanced through QoC aware contracts that facilitate the expression of requirements and guarantees. The work in [62] offers a quality model for context information and a context management mechanism for inconsistency resolution. This mechanism is based on ER ontology based model with the extension of quality

measurements. They claim that adoption of context-aware applications in real-life systems is impeded due to lack of quality of context management. For measuring quality of context, three parameters were proposed. The first is delay time which “is the time interval between the time when the situation happens in real world and the time when the situation is recognized in computers” [62]. This translates to freshness parameters used in other models. The second is context correctness probability which translate to the accuracy parameter in [58] or the difference between the actual state and the reported state by the different sensors. The third is context consistency probability which implies both precision and trustworthiness in other models. The authors in [63] focus on the modeling and management of context using object-based approach. Their proposed modeling approach is based on tagging context associations with the relevant QoC measures. They include a number of observations on the nature of context information and list them as

1. Context information exhibits a range of temporal characteristics such as static context or dynamic context. Static context is context data that do not change much over time and therefore are candidate to be manually input by application users. An example of static context is user profile data. The authors comment on static context as more reliable than dynamic contexts since they are usually input by users, but are prone to become stale if users do not update that data. Dynamic context are parameters that change over time and are most conveniently captured implicitly through sensors.
2. Context information can be imperfect. Sources of imperfections lie due to the fact that context data may be incorrect (or inaccurate), it can be inconsistent (or imprecise), or context data can be incomplete. The authors offer a technique for conflict resolution between conflicting context parameters by favoring the class of context that is more reliable.
3. Context data can be represented differently. A location data can be expressed by a GPS coordinate, a street address or a region depending on the need of the application. This translates to the the class of data representation quality parameters proposed in [58].
4. Context data can be interrelated. Such relationships can be obvious such as the relationships between a user location and a device location, other dependencies can be implicit such as dependencies between context data such as inference or derivation rules. This is also similar to the context parameters dependencies modeled in [58].

The authors in [64] quantify the Quality of Context parameters to be presented in a form suitable for use within applications in a pervasive system. They present a mechanism to tailor

the Quality of Context parameters based on the needs of the application and then evaluate these parameters. The authors also claim that context information is imperfect by nature as its quality is dependent on the way it was acquired. The authors classify context quality issues as incorrect (synonym for inaccurate in [58]), inconsistent (similar to imprecise in [58]), or incomplete such as having missing information. The authors in [64] divides the domain of QoC into QoC Parameters and QoC Sources. QoC parameters are then divided into Generic parameters and Domain-specific parameters. QoC Sources are divided into Sensed sources or User Profiled sources. This classification of QoC information is summarized in the below figure which is adapted from [64].

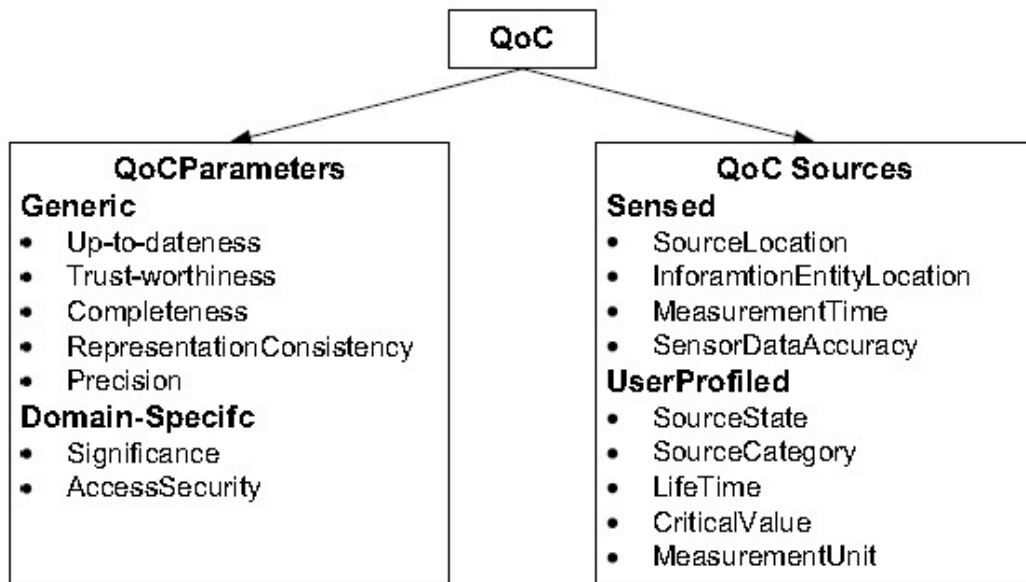


Figure 15. Classification of QoC information adapted from [64]

QoC sources are either the quantities that are sensed from the environment or the profiled configuration of the system. They describe the information about the source of context information, environment being sensed, and the information data itself. Information data can be directly sensed from the environment, inferred, or statically configured. It is worth noting the CriticalValue parameter of QoC Sources. This parameter indicates how critical is a context information to a specific application. This can be used by applications to choose not to make decisions if a context value is missing. QoC parameters are derived from QoC sources and are represented in form usable by an application. Those are splitted into Generic and Domain-Specific parameters. Generic parameters are the ones needed by most applications and Domain-Specific parameters are only important for some applications such as significance (derived from critical value) and access security. To map the structure in [58],

Up-to-datedness is the same as freshness in [58], RepresentationConsistency is the same as comparability, and trustworthiness and completeness are the same. It is not clear from the paper how the authors define precision.

2.8. Nomadism

One of the research sub-problems in the field of Context-Aware Advertising is Nomadism. A nomadic situation is the situation where a user is accessing a service from a device other than his personal device. In our context, when a user is viewing media content and advertisements from a friend's place or from a hotel - or may be a public place like a cafe, restaurant or boarding gate at the airport. This poses the question of how to identify the user and how to provide him with the necessary service on foreign devices without compromising the user's privacy or sharing his login credentials with others. Given this perspective, the research efforts in this field easily confuse nomadism with mobility. Mobility is when a user is on the move and wants to receive the same service on multiple devices he owns or through one mobile device. A nomadic user is not necessarily mobile since he can be in a hotel room watching TV or at a friend's place watching a match. In general, the field of nomadism covers both nomadic users and nomadic services. An example of nomadic services are the services hosted on a mobile device that can be accessible using multiple networks (WiFi, or 3G) depending on the available network. Since the purpose of this work is on Context-Aware Advertisements, only works related to nomadic users will be covered.

The work in [65-66] provides a relevant example of a nomadic situation where a user is visiting an exhibition guide and is receiving content recommendations through a non-personal PDA (one that is given to the visitor at the gate of the exhibition or museum) or an information Kiosk. The authors describe a nomadic system by "continuous access to information spaces independent from specific devices" [65]. Their works describe the goal and practice of a nomadic exhibition guide called "Hippie". The purpose of Hippie is to recommend articles to be visited and recommend specific content for each article based on user's interest such as analysis of the artwork or history. Hippie has been developed for a cultural environment, providing information about an art exhibition and a fair. As a basis for recommendation, the authors propose a definition for Context of Use and discuss that the more information included in the context of use, the more effective, efficient and satisfactory a user's visit will be. Their proposed context of use contains three models - two static models and one dynamic model. The static models represent the domain and the space. The domain

model contains information about the objects and classes of information to be presented. The space model contains information about the physical environment where the nomadic system is used and the locations of objects in this space. The dynamic model represents the user model which describes the knowledge of user's interests and movement. This is updated and inferred automatically by the system. The application also allows users to specify their interests to accelerate the adaptation of the recommendation engine. Feedback of the users on the output recommendations is implicitly detected by users listening or watching presentations. If a user watched or listened to the presentation till the end, this indicates a positive feedback, while if a user skipped or stopped the presentation this indicates a negative feedback. Location of the user is detected indoors by fitting infrared infrastructure in the environment space. The orientation of the user is detected by an electronic compass along with infrared receivers attached to the users and connected to the handheld device. Infrared emitters fitted on the different objects and gates send their ID's to be received by the infrared receivers attached to users to identify the user's location as well as the object of concern. The purpose of this system is to predict the user's information needs during the episode of the visit. User login or identity is not required as only a session identifier is needed to describe the episode of the visit. This session identifier is automatically started when the user is given a handheld device at the entrance of the exhibition.

Another research that supported nomadic situations is described in [13] which is part of UP-TO-US project described earlier in the domain of Context-Aware Multimedia. The authors in [13] describe content personalization during nomadism as "allowing the user to access his personalized IPTV content in a nomadic situation like in a hotel, in a friend's house or anywhere outside his domestic sphere." Their architecture contains a dedicated module called Nomadic Service Module (NSM) to support nomadic situations. This module contains replicas of the other modules which represent the home situation representing the nomadic situation to differentiation between the home status of the user and the nomadic status of the user. The NSM module communicates back and forth with the home domain of the user to update the context status and receive content recommendations. The user is identified using RFID tags and RFID readers. User's scan their RFID tag at the viewing device to identify their presence and interest to use this device and receive content recommendations on it. The architecture supports both nomadism and mobility since in a home domain, a user can move from one room to another and migrate his session to the new location by scanning his RFID tag. If the user wishes to transfer his session to a mobile device, a mobile application is used to receive input instruction from the user and transfer the user's session to the mobile device.

The approach is invasive in nature as the user has to provide his identity (using RFID) and thus providing his location rather than detecting and following the user implicitly.

Chapter 3: Proposed Approach

3.1. Contextual Parameters

The first contribution which involves the evaluation of contextual parameters was conducted by performing a literature survey in the field of Context-Aware Advertising and reporting on the effectiveness of each contextual parameter. The output of this study is shown in the literature review section of this work. The second contribution is to include a methodology for gathering the contextual parameters and designing the system in a way that facilitates recommending advertisements to an individual users and to a group of users. Contextual information will be gathered from three main channels:

1. The first type of information gathering is a static profile that the user creates upon registration.
2. The second type of information gathering is performed during runtime by accessing different devices (camera, and Location Information) and analyzing their outputs for meaningful business context. An image will be analyzed to detect age, gender, emotions and attention of viewers. Location Services API's available in devices will be used to retrieve the user's location. These API's are available in different mobile platforms and HTML5 for web development and return GPS coordinates. In case the user is indoors and no GPS signal is accessible, the API will get location data from the network (3G, WiFi, or Wired) and will still return the corresponding GPS coordinates. The returned GPS coordinates will then correspond to the nearest cell tower (if connected through 3G network), nearest edge node (if connected using wired or WiFi networks) or triangulated location (if multiple cell towers are available in the region). The returned GPS coordinates will be analyzed to detect region (district level) of the user.
3. The third type of context information is gathered by integrating with social networks (Facebook) and analyzing the interests of the viewers. In case of multiple viewers, it is expected that only one logged in user will be available so the information on interests whether from profile setup or social network integration will only be available for the logged in user. To circumvent this issue, a facial identification module is applied on the captured image to identify other registered users. This way, a user profile gets automatically detected when a registered user visits a new place without having to supply login credentials.

A prototype was developed for gathering the contextual information and making it ready for delivery to the recommendation engine. The prototype supports gathering all three types of information listed above and performing higher level analysis, normalization and fusion.

Details of the developed prototype are discussed in the Implementation section of this document.

3.2. Recommendation Engine

The recommendation engine will support real-time requests. The engine will also support different levels of personalization based on the amount of information the user allows access to the application (e.g. user denies access to the camera device). In case the user is not logged in and no device access is allowed, the system will default to showing random advertisements that target the country of the user. The recommendation engine will be composed of multiple phases:

1. The first phase assumes that each advertisement has a target audience profile associated with it (age, gender, location, emotion, etc). Such parameters can be used in direct filtering of the advertisements to reduce the input of the next phase.
2. The second phase takes user interest into consideration and performs item-based filtering to recommend an advertisement. The item-based filtering approach is chosen since it performs comparisons on the list of available items to recommend; which is expected to be much smaller than the number of users (viewers) in the system. This will allow for a better performance and faster recommendations. The update of the similarity matrix between advertisements can be done on a nightly basis or on a different cluster of servers; thus, not affecting the overall performance of the system. The recommendation engine will also take into consideration the marketing strategy of the advertiser for budget optimization. In case the advertiser has a reach strategy then each ad will be shown at most once to each viewer in a day. If the strategy is to target frequency then the system will repeatedly show the ad to a smaller target of users up to a certain limit specified by the advertisers.
3. The third phase employs emotion context to re-sort the recommendations based on the preferences of consuming the output recommendations from the second phase under the detected emotion. This is done by joining the retrieved recommendations with their emotion context latent preferences and performing a quick sort on the preference value.

In case more than one emotion is detected by multiple viewers, the highest preference emotion for each item is selected for sorting.

3.2.1. Vectorization of Advertisements

As mentioned in the Contributions section, different approaches for constructing the vector describing each advertisement will be explored. The output vector will then be used to compute the similarity matrix utilized in the item-based filtering recommendation approach. The purpose is to incorporate intrinsic properties of the advertisement along with historical transactions to circumvent the cold-start problem.

Approach 1: The first approach will attempt to compute similarities between each element of the vector for all the nominal attributes then apply a similarity measure (cosine, correlation, etc.) for the numerical attributes. Historical transactions will be split into two sets: users who liked an ad and users who did not like an ad. The similarity for these two sets will be computed using Jaccard similarity coefficient. The final similarity measure will be an average of all the similarities calculated for each attribute. This approach has the advantage of circumventing the cold-start problem (when no historical transactions are available), but it also suffers from the disadvantage of not giving additional weight to historical transactions in a way that minimizes the impact of content properties when more and more historical transactions are entered into the system. This is mainly due to the fact that historical transactions are represented by two attributes: users who liked the ad and users who disliked the ad. Accordingly, the weight of historical transactions in this equation is two out of all the available attributes describing an advertisement. To circumvent that drawback, weights can be assigned to different attributes such that we give more weight to attributes describing historical transactions than to other content properties attributes such as brand name. This approach will assign low similarity measures between content due to the cold-start problem, but still similar content will receive higher similarity coefficient than dis-similar content. By sorting similar content by similarity coefficients we can still apply a top N technique to fetch the most similar items, but it denies us the capability to set a threshold below which we can claim that there are no similar items.

Approach 2: A second approach would be to attempt to transform all attributes to numerical attributes and append historical ratings to this vector. This is based on a hypothesis that, in the initial state of the system, content properties will be more than historical transactions and will have more influence on the similarity measure. As more and more historical transactions are

added, it is expected that user ratings will have more weight in the similarity calculations than intrinsic item properties. This is yet to be proven by experimentation. This approach still suffers from a weakness of not capturing the difference between content that received too many reviews and content that received little or no reviews. This is due to the fact that both vectors have to be of equal size and therefore it works only on the intersection between the two sets of user ratings ignoring the original sizes of the different sets. Circumventing that issue can be applied by adding a numerical attribute capturing the number of ratings for each item. This will add a new dimension to each item, but still it is expected that as the historical transactions grow, the impact of item properties will be reduced (including the new dimension we have added). Another challenge that faces the second approach lies in presenting nominal attributes as numerical values. For nominal attributes that have small range of possible values such as target gender, this might not be a huge problem. For nominal attributes that can hold many values such as brand name it might pose a problem due to the fact that brand names are not limited or bounded. For such kind of attributes, it is possible to just assign a sequence number to each value and use that as the representation in the item vector. However, due to the large range of values, a brand name represented by 1 will be farther away in any distance calculation than a brand name represented by 100 than a brand name represented by 50. This defeats the semantics of similarity between brand names. Several methodologies are proposed to approach this challenge. The first approach is to ignore the brand name and only include more tractable nominal attributes describing it, such as target shopping interest, and including those in the vector describing each item. Alternatively, a similarity coefficient between brands can be computed separately and included in the vector describing each item. A third approach is to normalize the brand similarity value to either same brand (similarity = 1) or different brands (similarity = 0). The approach for constructing the vector describing each advertisement is yet to be decided based on the results of further experimentation.

3.3. High Level System Architecture

Below is a proposed system architecture.

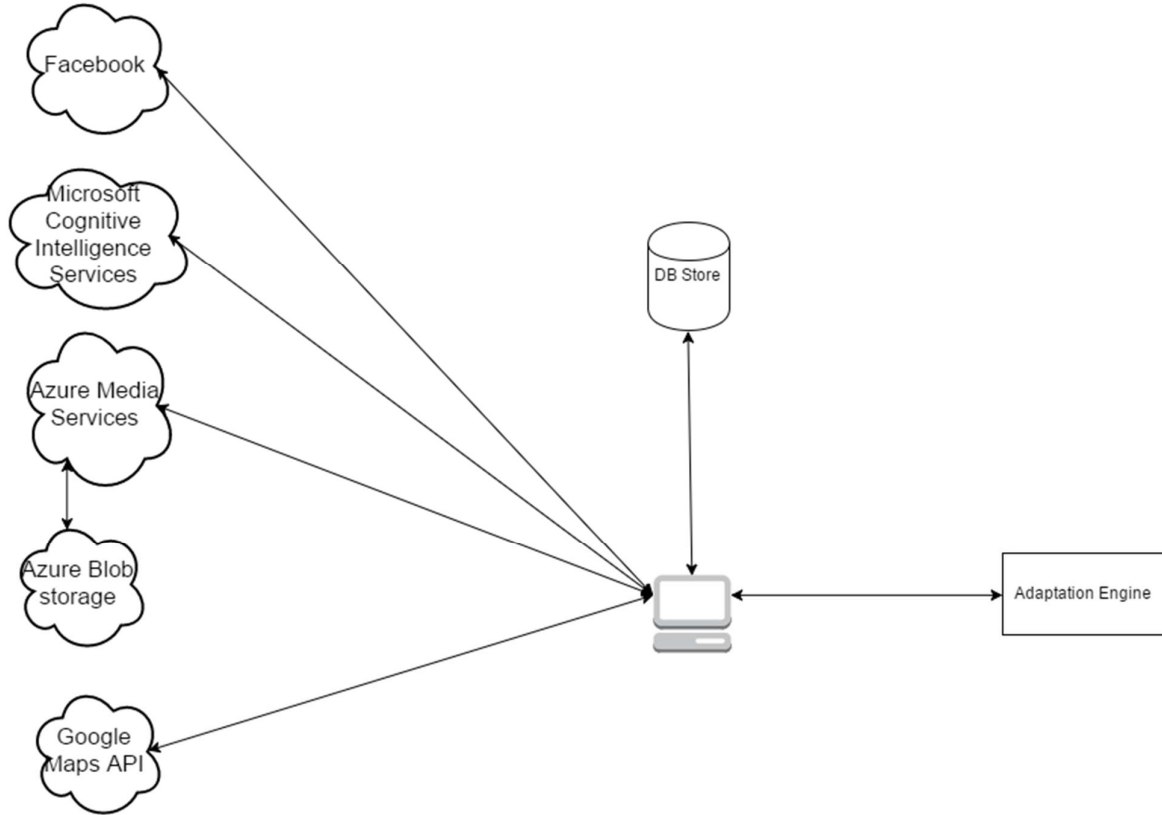


Figure 16. System high-level architecture

The above diagram shows the overall architecture of the system. The left part of the diagram shows all the cloud services utilized by the system. The video rendering engine is performed using Azure Media Services which supports dynamic packaging of content to suit the bandwidth requirements of the client. This will be used to render content and advertisements to clients. Azure Media Services utilizes Azure Blob storage service to store the video files to be rendered. The rest of the cloud services are used either as context sources or context enrichment services to formulate the context information to be sent to the recommendation engine. All communications happen between the client device and the cloud services directly over a secure connection to ensure the privacy of users. The client device also communicates with a database service to store and retrieve user's information. Upon formulating the context information, the client sends the data to an adaptation server which is responsible for delivering the right advertisements to users and sends that information back to the client device.

3.4. Class Diagram

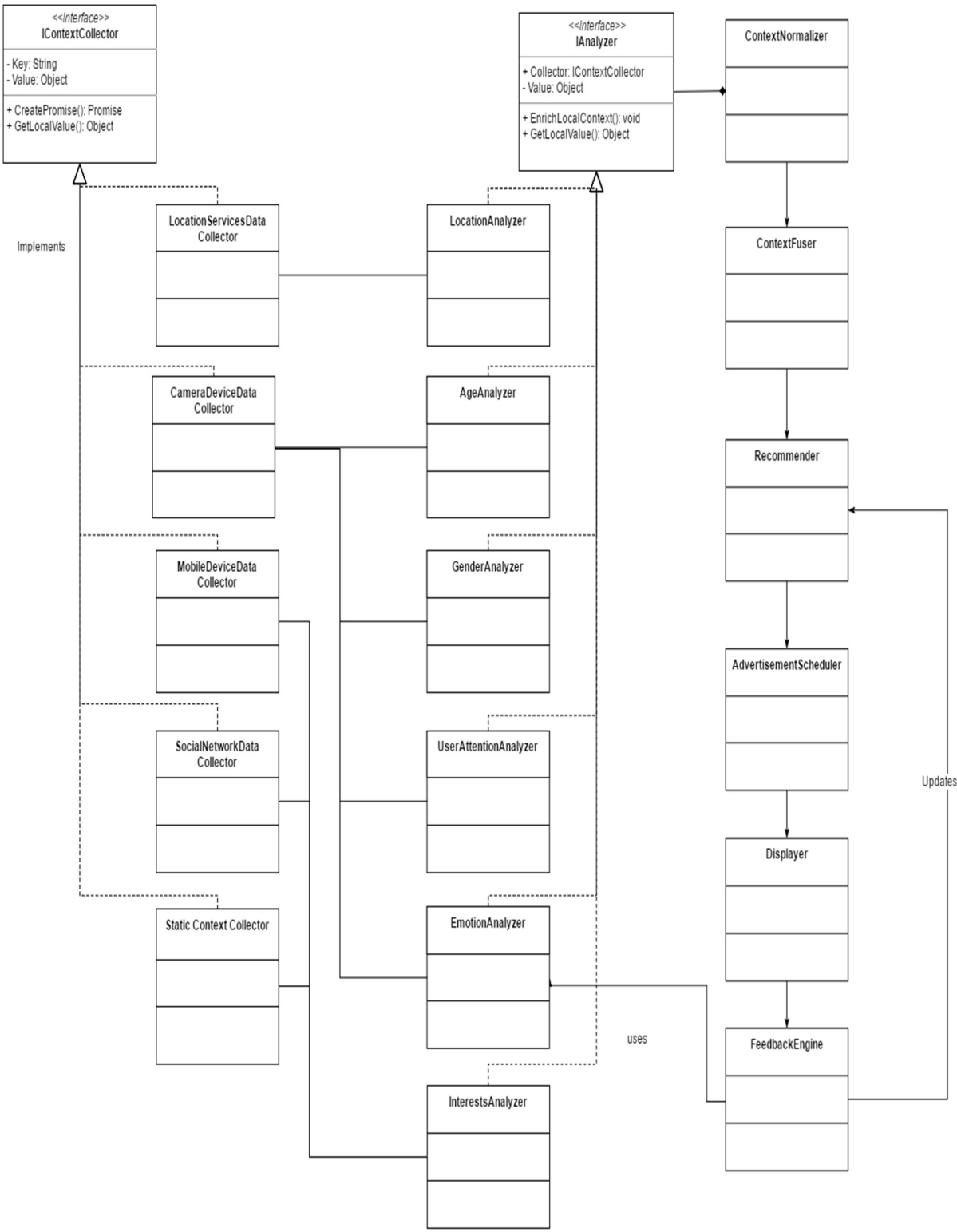


Figure 17. Class Diagram

The above shows the class diagram of the overall system. The system is composed of context collectors that implement IContextCollector interface. These collectors are mainly responsible to gather raw context data from various sources such as the camera device, location services, social networks or even static context data. The raw data are then given to one of the analyzer classes that implement IAnalyzer interface. These analyzers take the raw context data and extract higher level information from that raw data. For example, GenderAnalyzer class will take a raw image and identify the genders of the faces in these images; LocationAnalyzer class will take a GPS coordinate and extract the region where that address is located. The analyzed context information are then given to a ContextNormalizer class which aggregates transforms all data structures to a unified model. The output of the context normalizer are fused by a ContextFuser class into one data structure that is ready to be serialized and communicated over the network to the Recommender class. The Recommender class prepares a list of recommendations along with their confidence levels and priorities for each user. These recommendations are then scheduled by the AdvertisementScheduler class to identify which recommendations will be displayed and when and send them to the displayer during the triggered ad breaks. A FeedbackEngine class is used that uses the Displayer class to identify if the user watched the ad till the end or not and uses the EmotionAnalyzer class to identify if the user was satisfied with this ad or not. The gathered feedback is then fed into the recommendation engine to enhance its precision.

3.5. Interface descriptions

The system contains two main interfaces IContextCollector and IContextAnalyzer. Classes that implement IContextCollector interface are responsible for communicating with devices and external context sources to extract the raw data. Classes that implement the IContextAnalyzer interface are responsible for extracting higher level context information from the raw context data.

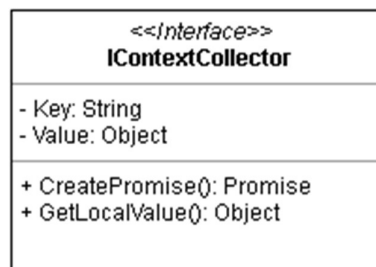


Figure 18. IContextCollector interface

IContextCollector interface has the following properties:

1. Key: a description of the raw context
2. Value: the value of the raw context

It also has the following methods:

1. CreatePromise(): a method that returns a Promise object. This method creates an asynchronous task that handles the communication with the context source(s) and sets the Key and Value properties of the object instance.
2. GetLocalValue(): method at returns an object. This is a public method that is invoked by external objects and returns the instance variable values of the instance after the completion of the promise.

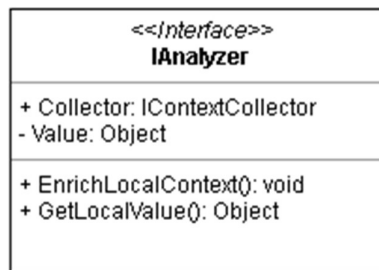


Figure 19. IAnalyzer interface

IAnalyzer interface has the following properties:

1. Collector: instance of a class that implements IContextCollector interface
2. Value: object that holds the high level context information

IAnalyzer interface has the following methods:

1. EnrichLocalContext(): this method utilizes the IContextCollector instance variable to extract raw context information, waits for the promise to complete and then utilizes external services to enrich the raw context values with higher level context information (image to faces to genders and ages, GPS coordinates to regions, etc.)
2. GetLocalValue(): method used by external objects to retrieve the value of the analyzed context information.

3.6. Communication Model

The communication model can be described in a high level by the following diagram.

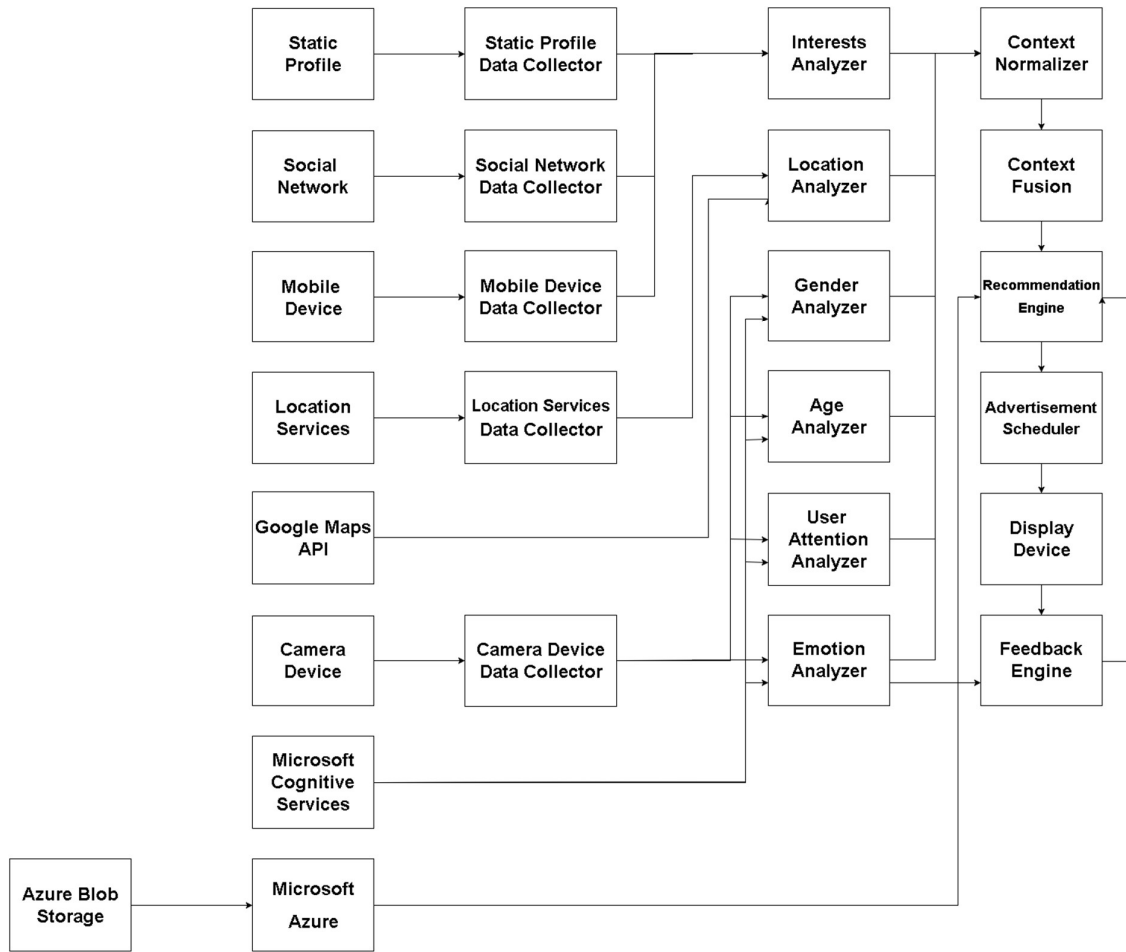


Figure 20. Communication model between system components

As shown in the diagram, different context sources and services are used either to gather raw context information or enrich the raw context data.

1. Interests context formulation: Raw data coming from the user's static profile, social network profile and mobile device can be gathered by different specialized collectors and passed on to the interests analyzer. The interests analyzer aggregates all these data and formalizes them into a usable format by marketers who can then target advertisements to users with specific set of interests.
2. Location context formulation: Location Services data collector uses the location services on the viewing device to extract the raw GPS coordinates of the user regardless of the type of network the user is utilizing. The raw location data is then passed on to the location analyzer which utilizes Google Maps API cloud service to translate the raw coordinates to high level region.
3. Camera context formulation: The Camera Device data collector captures a photo using the hardware camera device and sends the raw image to different analyzers that

utilize Microsoft Cognitive Services, each specializing in a different type of analysis (ages, genders, emotions, and attention spans).

4. Context Normalization and Fusion: Context Normalizer is used to transform all the gathered data into a unified format. The context fuser serializes this normalized data to be sent to the recommendation engine.
5. Producing recommendations: The normalized and fused context is received by the recommendation engine to compute the relevant ads to push to the user. The output of the recommendation engine is given to a scheduler which decides which ads will be displayed and when depending on user's historical viewership and to avoid showing the same most recommended advertisement repetitively. Ads are then pushed to the displaying device and a feedback engine is used to gather data that can be used to enhance the recommendation engine in the future.

The following diagram zooms in the high level architecture diagram shown above.

Specifically it lists the relevant context information along with the relevant context sources.

In addition, it zooms in on the adaptation engine to show the different internal modules. The red boxes represent out-of-scope components and the gray boxes represent future enhancements. The following subsections describe each sub-component in detail.

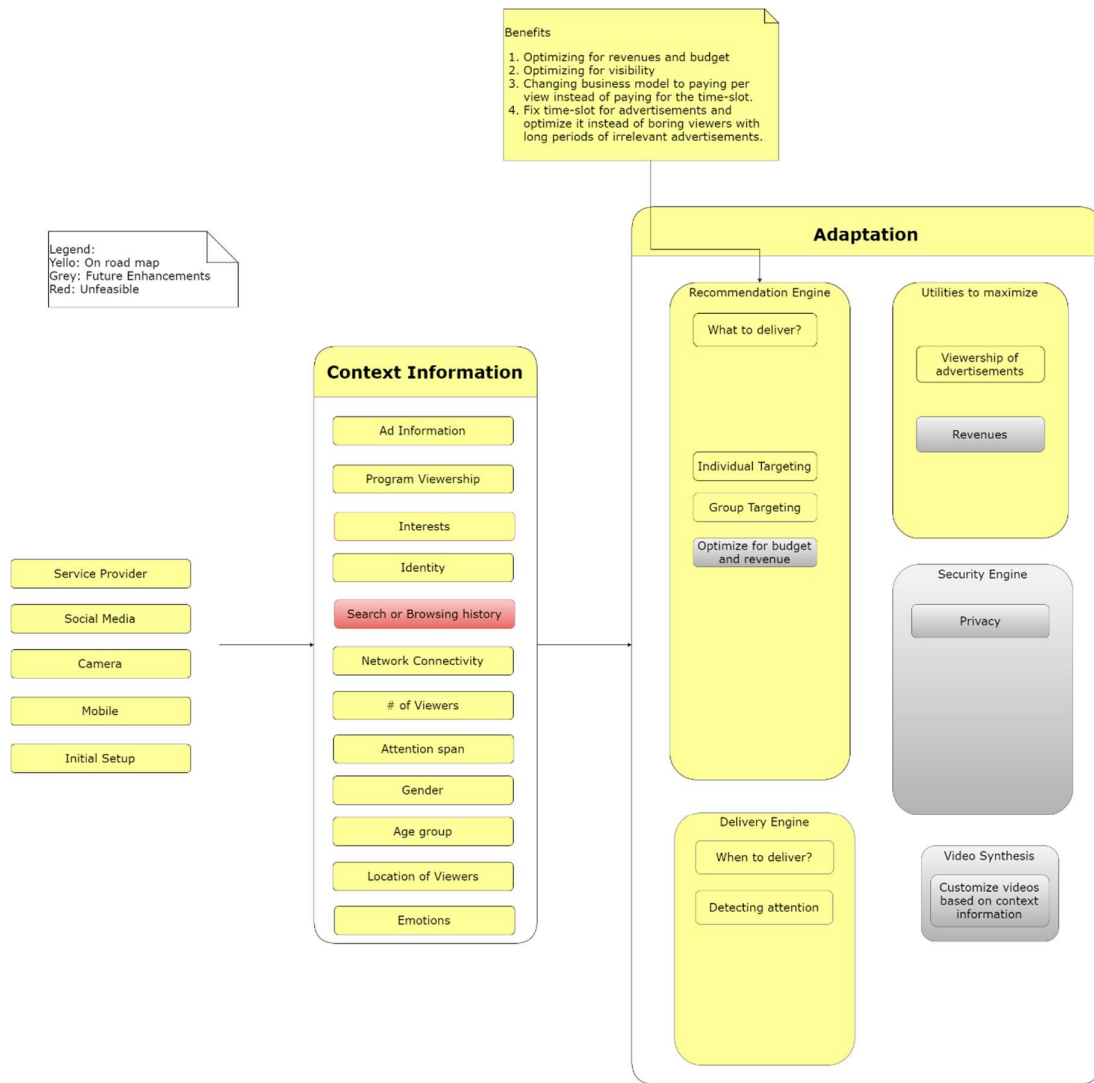


Figure 21. Detailed system components and employed information

3.7. Context Sources

The following are the proposed context sources with their descriptions and the information extracted from each context source:

1. **Service Provider:** The service provider provides the content to the user. The service provider also receives advertisement bidding requests along with their target audience and characteristics of the ad itself (eg. genre, commercial/infomercial, appealing to emotions, etc.). In this system, bidders information may include information about the target audience profile (such as age, gender, education level, interests, etc), target location, bid per viewership (how much they are willing to pay per viewership) and

total budget to limit their spending. Also, the bidding request may include budget optimization information such as targeting for maximum reach or targeting for maximum frequency. In case of targeting for maximum reach, the same advertisement will be displayed at most once to each viewer; whereas for maximum frequency, the same advertisement will be shown multiple times to the same set of target audience. In this study, due to the lack of a real service provider, bidding information will be excluded from this study and postponed to an industrial implementation.

2. **Social Media:** Social media integration will be utilized to provide system login, user identity, user attributes (such as demographics) as well as user interests
3. **Camera:** A camera device can be very useful when combined with computer vision algorithms to identify the number of current viewers, their attention, their genders and ages plus their emotions.
4. **Mobile:** A mobile device can be included to identify user identity and user emotions. In addition, the list of applications installed on the device can be an indicator of the user interests. In this study, the mobile device context will be ignored as alternative sources will be used to gather the same context. Social media will be used to gather user identity and interests instead of the mobile device. The location information will be extracted from the viewing device itself instead of a mobile device. This allows for granularity of location information up to which room in a household the user is sitting in.
5. **Initial Setup on device:** Upon registering for the service, users will be prompted to create a profile to indicate their viewership interests to recommend content accordingly and shopping interests to use as additional context information for recommending advertisements.

3.8. Adaptation Modules

3.8.1. Recommendation Engine

The recommendation engine is responsible for selecting the most relevant advertisements to send to a user taking into consideration all the relevant context sources. The engine itself may be split into several subsystems where the initial subsystem performs direct filtering on the advertisements based on input target location and demographics set by the advertisers and the second phase performs item-based filtering to generate the most relevant ads based on user's

interests. In addition, it is the job of the recommendation engine to optimize for revenues and budgets which will be available upon an industrial implementation. The proposed recommendation engine will focus on individual targeting while group targeting is left as a future enhancement.

3.8.2. Delivery Engine

The delivery engine is responsible for answering the question of when and how to deliver an ad? Currently known practice in advertisement delivery is for content providers to predefine ad breaks and sell them as advertising slots based on content viewership. A different approach can be by automatically detecting user's attention and having a machine intelligent methodology for showing ads. This may help in increasing ad viewership and optimization of number of breaks as well as the length of each break. The proposed approach is to utilize the yaw angle detected by the face detection algorithm in the context information to analyzer viewer's attention. The question of how to deliver an ad is based on the network parameters to adjust the video bitrate based on the connection speed to optimize for quality of experience. The project will utilize Azure Media Services which has a dynamic packaging component that automatically generates different versions of videos with different pixel densities. Azure Media Services then deliver the most appropriate version of the video based on the detected connection speed with the client.

3.8.3. Utilities Maximization

As a control for the above engines several utilities need to be maximized to ensure the adoption of the system:

1. Customer satisfaction: by providing relevant ads that users are interested in. This can be measured using the QoE measures described later in the Evaluation section.
2. Viewership of advertisements: increase the effectiveness of advertisement targeting and reach. This is meant to satisfy the strategy of ad bidders to either target for maximum reach or maximum frequency to target users without exceeding the preset budget. Allowing ad bidders to set their advertising strategy will help them better measure the relevant return on investment (RoI) and increase the adoption of the system by ad bidders.
3. Revenues: not all advertisers bid the same amount for the target profile, so the engine needs to optimize for revenues as well. This is meant to increase the revenues of the

service provider himself. Since advertisement bidding is out of scope of this work, it will be left for a future research work to study optimization techniques for ad scheduling towards maximum returns.

3.8.4. Security

Given the sensitivity of the context information for users, privacy becomes a main concern that need to be addressed by several measures. First, users will be given the option to opt-out from providing access to context sources to the system and thus will receive an inferior service. Second, none of the processed images and locations will be saved on the server and therefore the risk of losing information is minimal. Third, the context gathering will be performed on the client device itself not server side; thus, ruling out the possibility of sniffing data in transit.

3.8.5. Video Synthesis

Synthesizing videos aims at optimizing the return on ad breaks by showing more - shorter versions - of ads depending on the time slot available. This poses the challenge of how to edit videos on the fly while maintaining the marketing message; in other words, without losing the semantics of the video. This can be considered as a summarization problem which will be postponed to a future work as it poses a different research problem.

3.9. Client-Side Architecture

This section describes the different components of the client application and the logical flow of data between those components. Before going into detail on the client components the following sub-section describes the context information gathered from the different context sources described previously. The following subsections describes how these information are gathered and enriched before they are sent to the adaptation service.

3.9.1. Context Information

The following are the proposed information to be utilized as part of the context for advertisement recommendation:

1. Ad information: includes remaining budget, target audience and other advertisement characteristics. Such information are expected to be supplied by the service provider. Due to the exclusion of the service provider role from this study, this information will

not be included in the proposed study; however, a set of advertisements with their information will be assumed.

2. Program viewership: includes number of viewers and their demographics which are retrieved from a third party research organization and known as Nielsen ratings. Such information will be used when users deny access to personal information or hardware devices.
3. User Identity: which will be retrieved from Social Media. In this study, the proposed social network to integrate with is Facebook.
4. User Interests: composed of the pages users liked on Facebook and static data supplied during profile setup.
5. Search or browsing history: this information is used by search engines for behavioral targeting of online advertisements. Unfortunately, none of the search vendors offer an API to access such information and accordingly it will be dismissed from this study.
6. Network Connectivity: defines the bandwidth of the connection between the user and the service provider. Based on this information, different versions of the advertisements can be rendered to users to enhance their experience. In this study we propose the deployment over Azure Media Services technology which automatically detects user connection and performs dynamic packaging of the content to a different bit rate based on the detected user's connection speed.
7. Number of viewers: the number of persons sitting in front of the viewing device. This is gathered by processing the image captured from the camera device using a face detection algorithm. In this study, Microsoft Cognitive services Face detection API will be employed to detect the number of faces in an image.
8. Attention span: attention span defines whether users are actually looking to the screen or not. This can be identified by detecting the Yaw angle of the detected faces. The same face detection API will be used to detect the Yaw angle of viewers. This information can be very useful in identifying when to display an advertisement. Instead of having preconfigured ad breaks, such a feature can help in developing adaptive systems that increase the return on investment (ROI) by only showing ads when users are actually looking.
9. Gender: a demographic attribute that can be used to target advertisements. Gender can be detected using the same face detection API used in 7.
10. Age group: another demographic attribute that can be used to target advertisements. This is also detected using Microsoft Cognitive Services Face Detection API.

11. Emotion: such as happy, sad, angry, contempt, disgust, fear, surprise, or neutral. A different API also provided by Microsoft Cognitive Services will be used to detect this feature. Emotion detection can be combined with advertisement attributes to match the appropriate ad based on user's current emotion.
12. Location: defined at different granularity levels such as room within a household, GPS coordinates, street, district / region, or country levels. Advertisers will be allowed to choose the target audience for their advertisement based on their geographical region. In the proposed prototype, the GPS coordinates will be retrieved from the hardware GPS device in the viewing screen and Google Geocoding APIs will be used to find the relevant district / region for the retrieved coordinates.

3.9.2. Client Components

The client components which pertain to delivering advertisements can be thought of as different layers shown in the following diagram.



Figure 22. Client components

The bottom layer represents the context information sources which are described previously. These are mainly the hardware devices and the external social networks that are used to provide context to the system. The second layer represents the context collection part which mainly contains the set of APIs and tools used to gather information from the context sources and prepare them for enrichment. The context enrichment layer is responsible for using the raw context information and enriches it with more information such as the district level of the location, or the number of faces in the image. The main enrichments performed by the client are:

1. District Level Location: transforming the GPS coordinates to a district or region level. This process is known as reverse geocoding.
2. Face detection: Takes an image as input and detects the faces in that image. For each image a specific set of attributes is also detected such as age, gender and head pose (angle of viewing). Head pose is described by three variables (pitch, yaw and roll) which represent the tilt of the head on 3 axes. This research is interested in the yaw angle as it represents how far the user is looking to or away from the viewing screen.
3. Emotion detection: Takes an image as input and detects the emotions expressed by each face. This can be one of the following values - each represented by a confidence level:
 - a. Anger
 - b. Contempt
 - c. Disgust
 - d. Fear
 - e. Happiness
 - f. Neutral
 - g. Sadness
 - h. Surprise

These emotions are understood to be cross-culturally and universally communicated with particular facial expressions. The system compares the confidence level returned for each emotion and uses the one with highest confidence value as the emotion expressed by the user. The detection of emotions along with the confidence level of each emotion is performed through the [Emotion Detection API](#) provided by Microsoft Cognitive services.

4. Face Identification: Takes as input the detected faces from the face detection API and returns for each face the associated person identifier. This is very useful to help in roaming user profile while viewing from devices as a guest (visiting a friend's place for example). The system will utilize the returned person identifier to fetch from the database storage the relevant user profile and interests.

The context aggregation layer gathers the outputs from the context enrichment layer and prepares it for sending it to the adaptation layer.

3.9.3. Client Use Case

The following diagram shows the use case scenarios for user registration. User registration is an optional step. If a user chooses not to register and create a profile in the system, s/he will receive an inferior service as the adaptation engine will tend to serve random ads.

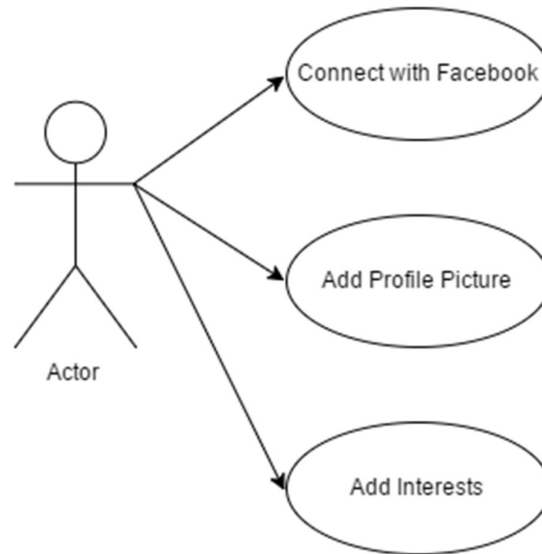


Figure 23. Client use case

When the user accesses the application, the user is prompted to log in with Facebook. This allows the system to retrieve user profile info from Social Media as well as user's interests. User profile is defined by name, age, gender, education, language, home country, etc. User interests retrieved from social media are defined by the pages liked by the user and the categories of those pages (political, entertainment, sports, etc.). If the user chooses to connect with Facebook, s/he is presented with a page to add profile picture. This step is necessary to be able to identify a roaming user later when s/he is viewing as a guest from another device using the face identification enrichment step. When the user adds a profile picture, the face identification engine is trained accordingly and a person id is created for this account. This person id is associated with the user's profile in the database for later retrieval. Next the user is prompted to specify system related interests (viewing interests and shopping interests). Viewing interests can be used for content recommendation while shopping interests can be used for advertisement recommendation.

3.9.4. Context Data Flow

The following diagram shows the data flow from context sources till the context aggregation step.

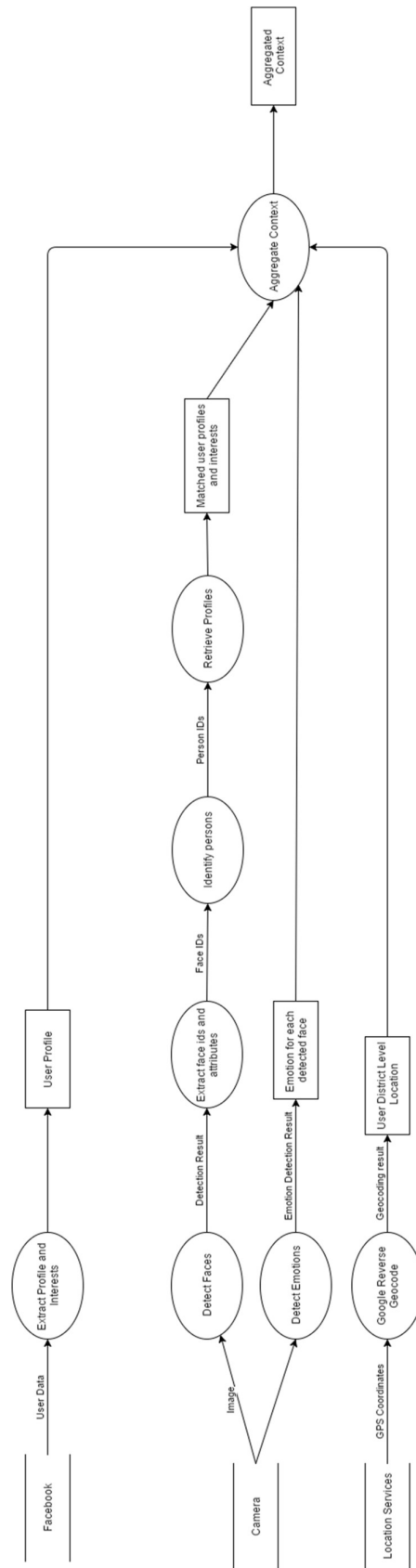


Figure 24. Client data flow

The first source of context information used is Social Media (Facebook). User logs into the system using Facebook Authentication and allows access to profile information and liked pages (interests). The client application then extracts user profile and interests' values and passes them to the aggregation component. The camera device is the second source of context information. The context collection component triggers the camera device after requesting permission from the user and captures the photo. The context collection component then passes the captured image to the context enrichment layer. In this layer the captured image is passed to the Face Detection API and Emotion Detection API provided by Microsoft Cognitive Services. The extracted emotions are passed directly to the aggregation component. The output of the face detection API is passed to the face identification API to identify the persons in the image. After identifying the persons in the image, the data is augmented with the stored profile information in the database and is passed to the aggregation component. Location information is also used as part of the user context to recommend advertisements. For this purpose, the location services available on the device are used to identify user's information. Location Services API return approximate GPS coordinate location of the user which can be calculated from the GPS signals directly (if accessible), assisted GPS (through wireless network), network signal triangulation (through 3G access), or from the nearest IP node in the network (if connected through ADSL or wired connection). The returned GPS coordinate is then passed to the context enrichment layer which uses Google Maps API to perform reverse geocoding on the input coordinates. Reverse geocoding is the process of taking GPS coordinates as input and returning a human readable address as output. Google Maps API has the capability of taking a GPS coordinate as input and returns human readable addresses at multiple granularity levels - from the street level to the country level. The system extracts the neighborhood level location and sends it to the context aggregation component. The same information flow can be explained using the sequence diagram below which reveals the participating object instances in the process.

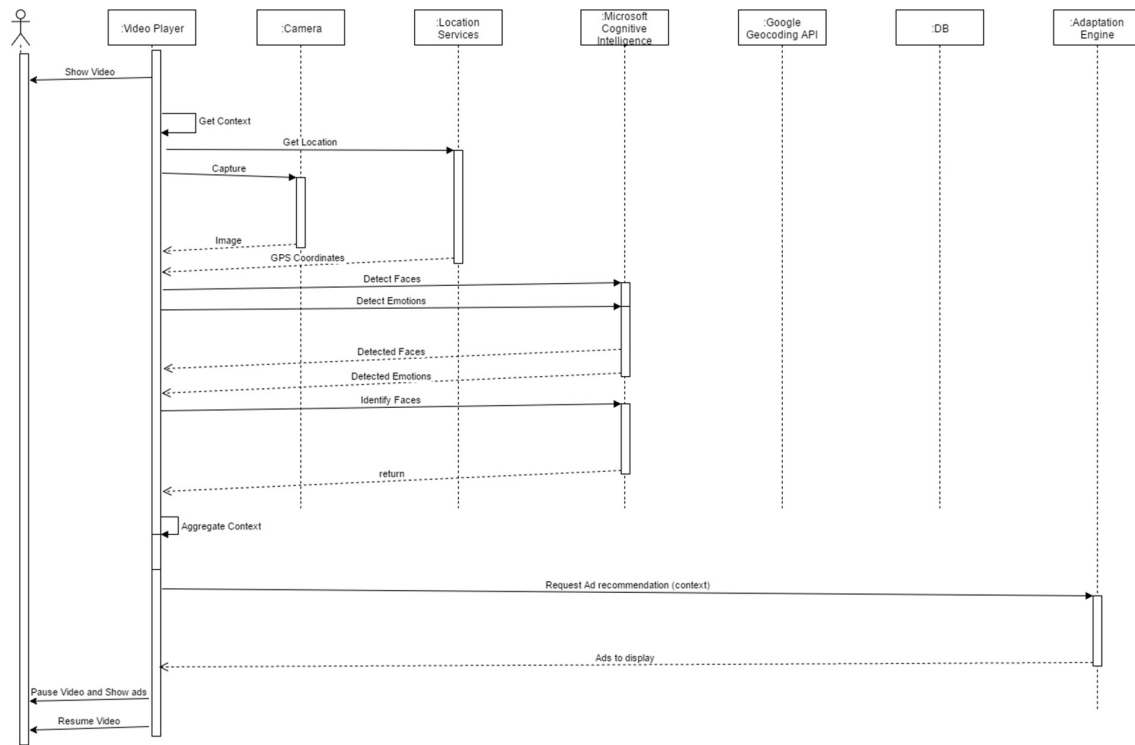


Figure 25. Sequence diagram on the client side

Chapter 4: Implementation

4.1. Client Side

A prototype for the client side components responsible for gathering context information and delivering them has been developed. The underlying technologies of choice are web technologies (HTML5 and Javascript). Web technologies have been selected due to their availability on a diverse array of devices (desktops, tablets and mobiles). The system allows users to either log in using their facebook accounts or not to log in at all if they choose, in which case the users will receive higher privacy but an inferior level of recommendation. Whether the users allow access to hardware devices or not (Location Services and Camera), the user will still receive the service but in case the user denies access to devices they will receive random advertisements which is an experience similar to that of regular TV. Users may allow access to Facebook account but not hardware devices or they may allow access to hardware devices but not facebook account. The following is a screenshot of the landing page.

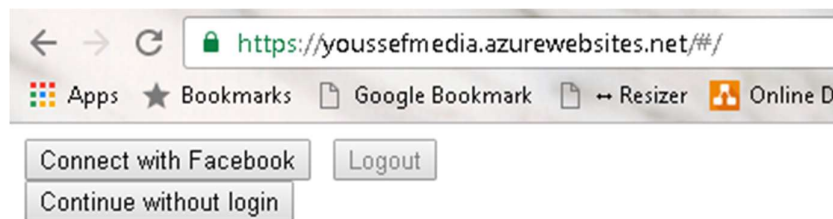


Figure 26. Sample home screen

Assuming the user chooses to log in with Facebook, s/he will be prompted with Facebook prompt to grant read access to account information and liked pages. Users will also be prompted to capture a profile picture that can be later used for face identification. As a first time setup, the users are asked to specify their interests in terms of viewership interests and shopping interests. Viewership interests can be later used to recommend content to users, while shopping interests can be used to recommend advertisements to users.

In the prototype only one episode content is presented since content recommendation is out of scope of this work. To test the context capturing techniques, a “Capture Context” button is presented which gathers the context, enriches it and displays the result in the screen. In

production, this button will be removed and instead content publishers will be given the option to either specify ad breaks manually or to allow the system to detect the best time to show an ad break based on detecting user attention. When the “Capture Context” button is clicked it triggers the workflow for accessing hardware devices and facebook account. The workflow goes as follows:

1. Capture logged in user information
2. Trigger the camera to capture an image
 - a. After capturing the image run the following 2 APIs in parallel available through Microsoft Cognitive Services
 - i. Face Detection
 1. The Face Detection API returns the following context information: face rectangle, age, gender, yaw angle
 2. The face rectangles are then passed to the Face Detection API to identify the persons represented by those faces
 3. The output of the Face Detection API is then used to retrieve profile information from database storage.
 - ii. Emotion Detection
3. Trigger the location services to retrieve the location
 - a. Use the returned GPS coordinates to identify the higher level region information using Google Maps API.

The main three steps above are performed in parallel. When all three threads return, a context object is formulated which is an aggregation of all the context information returned.

The face identification API is used to cover the user roaming scenario, the detected faces are passed to a face identification API which returns the best matched users to the supplied images. This allows for users visiting other places to be identified without having to log in and receive personalized recommendations without having to share their credentials with others. The system uses the returned information to retrieve any pre-stored user profile interests and Facebook interests.

Due to the sensitivity of the information being transferred over the wire, the web application mandates access over https and access to hardware devices won't be granted unless the user uses SSL protocol to access the application.

The following image is an extract of a logged in user context.

Logged in User:

- Name: Youssef Youssef
- Gender: male
- Birthday: 07/31/1983
- Email: youssefy@aucegypt.edu
- Hometown: Cairo, Egypt
- Location: Cairo, Egypt
- Languages:
 - Egyptian Arabic
 - English
- Viewership Interests:
 - Action
 - Adventure
 - Comedy
 - Science fiction
- Shopping Interests:
 - Autos & Vehicles
 - Beauty & Fitness
 - Computers & Electronics
 - Fashion
 - Food & Drink
 - Games
 - Hobbies & Leisure
 - Internet & Telecom
 - People & Society
 - Real Estate
 - Sports
 - Travel
- Facebook Interests:
 - EgyptVR360
 - National Geographic Egypt
 - the2fish.com
 - Dr. PI's SciFi
 - Bassem Youssef
 - عزالدين عروسة
 - مصر MBC
 - Office for Businesses
 - WINHGO

Neighbourhood: Al Matar, Qism El-Nozha, Cairo Governorate, Egypt

Figure 27. Snippet of captured user information using login

The following is an extract of a user context who chose not to log in. Note that since this user was already registered from another device, the face identification algorithm was able to reproduce the same context information (shown in Green color).

Detected Faces:

- Face 1:
 - Gender: male
 - Age: 36.4
 - Yaw: 1.5
 - Emotion: neutral
 - Name: Youssef Youssef
 - Gender: male
 - Birthday: 07/31/1983
 - Email: youssefy@aucegypt.edu
 - Hometown: Cairo, Egypt
 - Location: Cairo, Egypt
 - Languages:
 - Egyptian Arabic
 - English
 - Viewership Interests:
 - Action
 - Adventure
 - Comedy
 - Science fiction
 - Shopping Interests:
 - Autos & Vehicles
 - Beauty & Fitness
 - Computers & Electronics
 - Fashion
 - Food & Drink
 - Games
 - Hobbies & Leisure
 - Internet & Telecom
 - People & Society
 - Real Estate
 - Sports
 - Travel
 - Facebook Interests:
 - EgyptVR360
 - National Geographic Egypt
 - the2fish.com
 - Dr. PI's SciFi
 - Bassem Youssef
 - عريس وعروسة
 - مصر MBC
 - Office for Businesses
 - WINJIGO
 - Berlin sofa-bed
 - Clash Of Clans Sarcasm Society - COCSS
 - Microsoft
 - Entertainment Weekly

Neighbourhood: Al Matar, Qism El-Nozha, Cairo Governorate, Egypt

Figure 28. Snippet of captured user information without logging in

Notice the yaw angle returned by the face detection API in the above snapshot. This detected angle is used for identifying whether the user has been looking to the screen or not. An angle range between -20 and +20 indicates that the user is viewing the screen. The application keeps detecting this angle every minute and if the user is consistently looking at the screen for a configurable time span then an ad break is triggered automatically. In case multiple users are detected, then the number of users looking at the screen is counted and divided by the total number of detected users every minute. After the configurable time span ends, the average attention duration is calculated and an ad break is triggered accordingly.

4.2. Recommendation Engine

After capturing the context, the client side application sends the information to a back-end service to compute the recommended ads. In case the user is not logged in and devices are not allowed to be used then a random set of advertisements is retrieved. In case the user is logged in but no context was gathered then recommendations will be set based on the user profile and historical transactions. When a context is captured and only one user is detected and that user has logged in or was identified by the system, then recommendations for that user will be computed based on his profile and historical transactions and the output recommendations will be resorted based on the preferences of the currently detected emotion context. If only one user was detected and this user has no profile in the system, then the detected age, and gender attributes by the face detection API will be used to select targeted ads. Again, the output recommendations will also be resorted by latent preferences of the currently detected context. When multiple users are detected, the system automatically gathers the minimum and maximum ages of the detected users. The system extracts the detected genders and examines if all detected viewers are of the same gender. If users are not of the same gender then only ads that target both genders will be selected. After identifying the pool of ads that are suitable for the detected users, personal recommendations are computed based on the detected identities of each user if exists. The generated recommendations are then intersected along with the applicable pool of ads to find the common recommendations suitable for all detected users. This list of recommendations is also sorted by their preferences to the detected emotions. Since more than one emotion type can be detected, the application links each advertisement to the emotion with maximum preference value and use that value when resorting the list of recommended ads.

The following diagram shows the decision flow for selecting the mode of recommendation to follow.

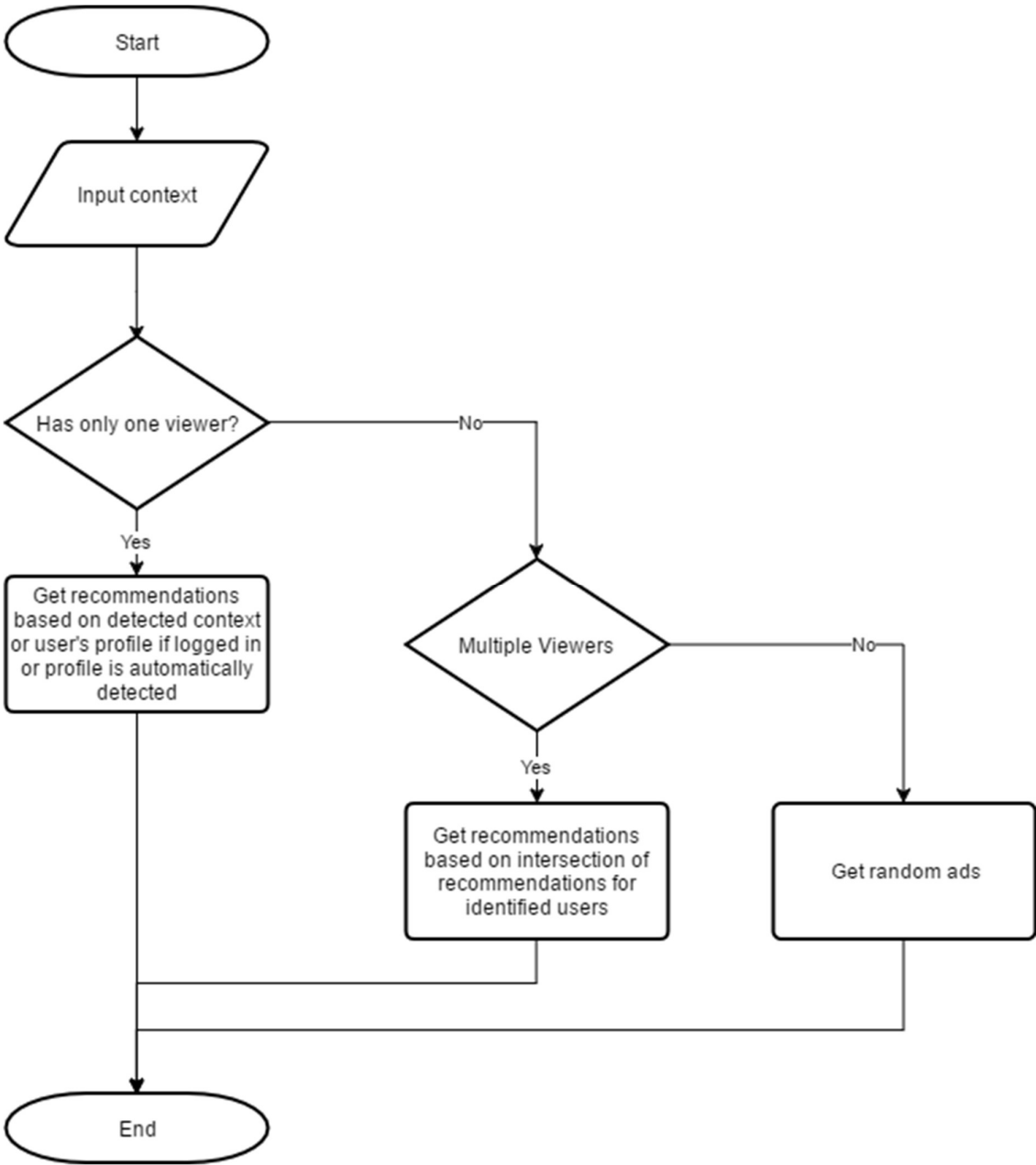


Figure 29. Recommendation engine decision flow

The process for generating recommendations for one user based on contextual information is shown in the following flowchart.

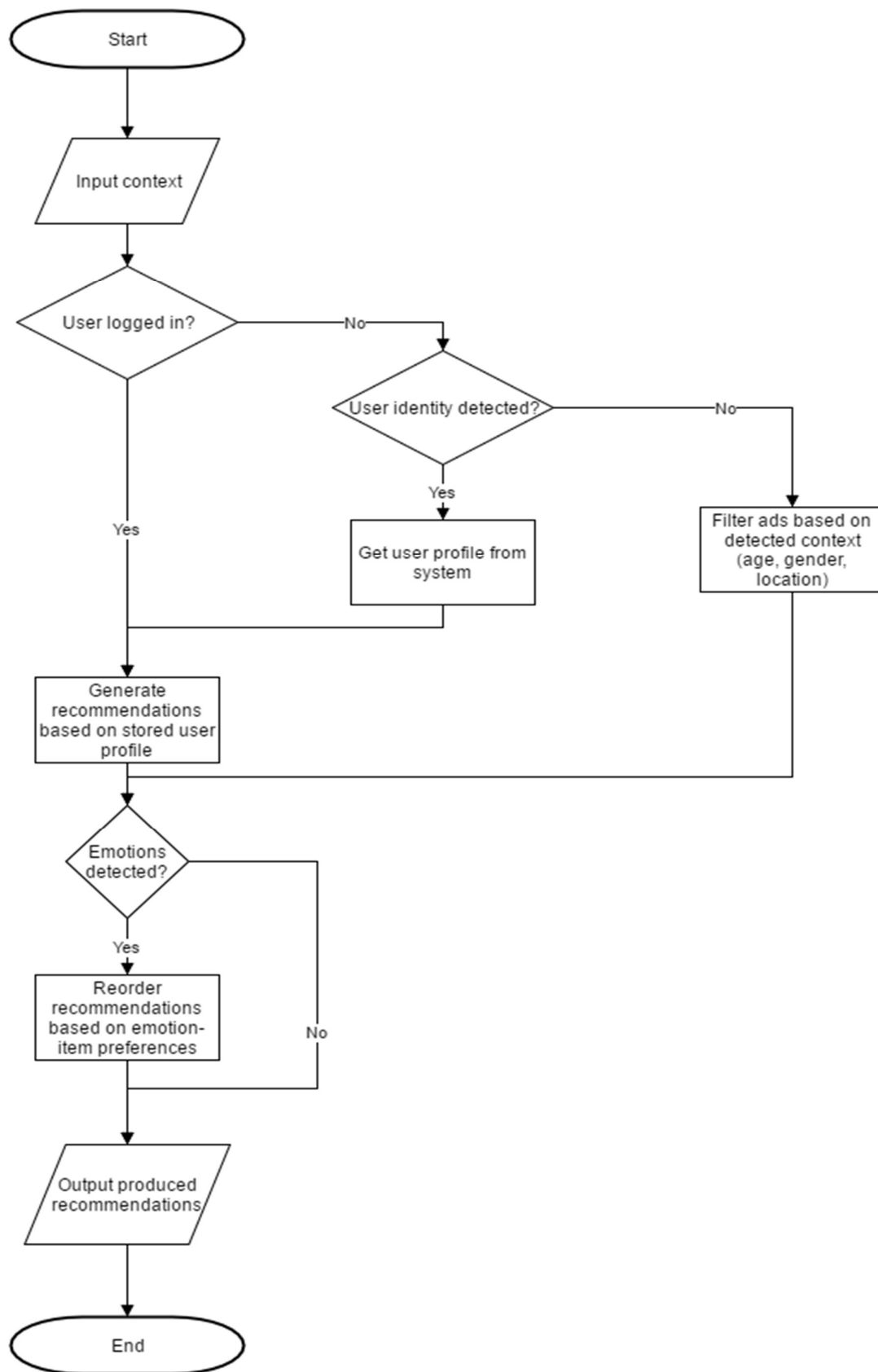


Figure 30. Context based ad recommendation for one user

When multiple viewers are detected in the same context, the previously described process is summarized in the below diagram.

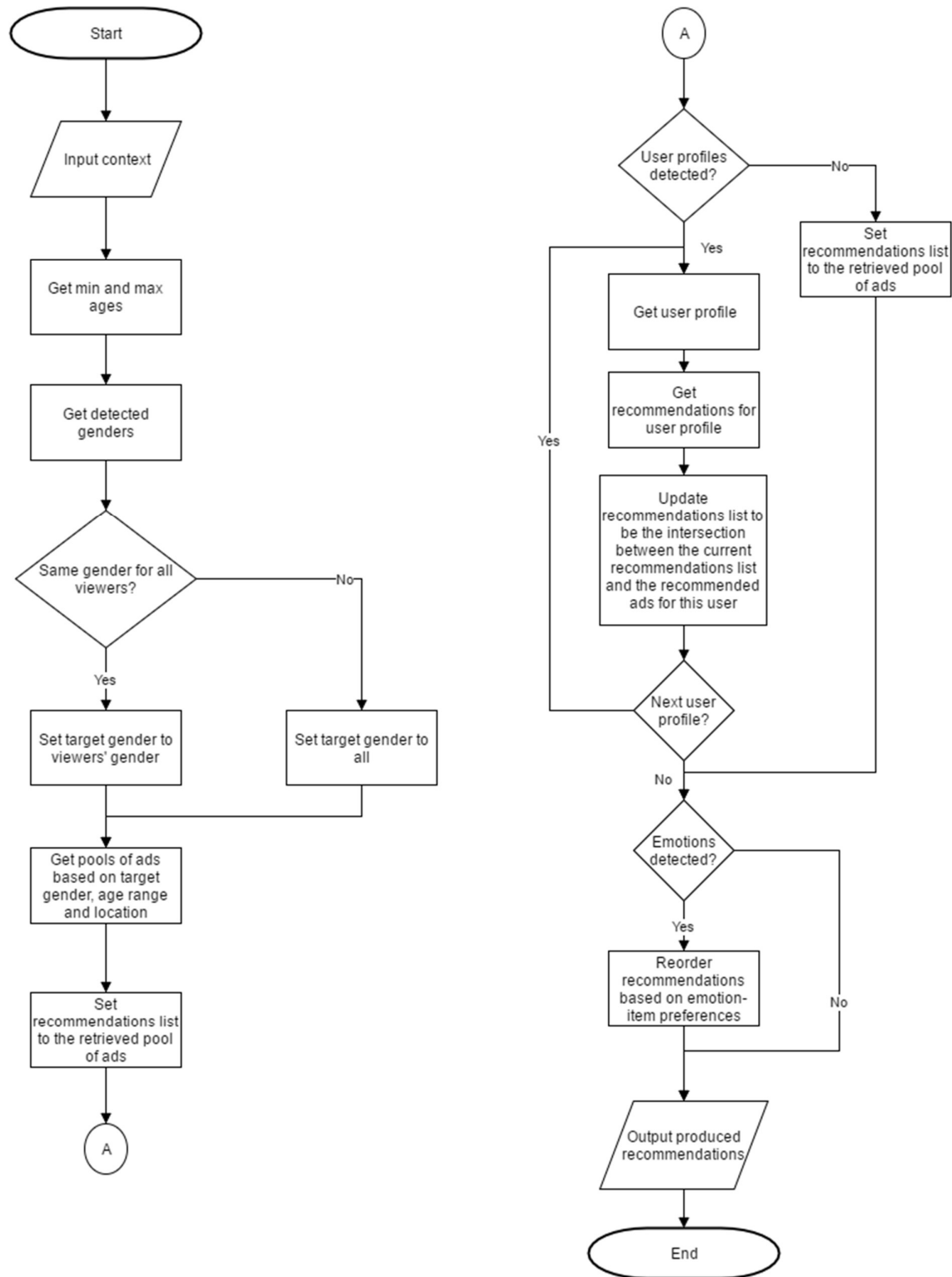


Figure 31. Ad recommendation for multiple users

When producing a recommendation set of ads to a specific user, the item-based collaborative filtering algorithm was implemented. Two variations were implemented and experimented with to identify the most suitable mode of implementation (see Experimentation section).

- The first mode of implementation takes as input the whole set of available ads and filters them by the target age, gender and location. This filtered list is then identified as the pool of ads which we will compute the predicted rating for each ad in this set. Then the system retrieves the set of historical ratings for the current user in concern. The similarity measures between the set of user rated ads and pool of ads are retrieved from the ads similarity matrix. These similarity measures for the pool of ads are then multiplied by the user rating and divided by the sum of similarity measures for all user rated ads to return the weighted rating for the new advertisement. The predicted ratings are then used to sort the recommendations in descending order before sending them back to the user. This approach can be further explained by walking through an example. Assuming a user with few historical ads and we want to predict ratings for new ads, table 3 shows the workings of the item-based filtering algorithm explained above.

Table 3. Example of predicting recommended ads to a user using item-based filtering

Historical ad	Rating (R)	New ad 1		New ad 2		New ad 3	
		Similarity (S)	R x S	Similarity (S)	R x S	Similarity (S)	R x S
Ad 1	1	0.9	0.9	0.8	0.8	0.4	0.4
Ad 2	1	0.7	0.7	0.6	0.6	0.3	0.3
Ad 3	0	0.3	0	0.2	0	0.8	0
Total		1.9	1.6	1.6	1.4	1.5	0.7
Normalized Sum(RxS)/Sum(S)		0.842		0.875		0.467	

From the calculations in table 3, we can identify that new ad 1 and new ad 2 have a higher predicted rating than new ad 3. Furthermore, the calculations show that there is a higher predicted preference for new ad 2 than new ad 1. Accordingly, the output of the recommendation engine should give a higher priority to new ad 2 to be displayed

to the user before new ad 1 and ignore new ad 3 from the recommendation. Since we have hundreds of ads in the system, our recommendation engine output returns the top N recommendations – where N is configurable in the client application.

- The second algorithm performs the same operations but filters out ads that were disliked by the user (rating =0) and focuses only on user liked ads. This this mode of implementation the user rated ads are only the ads with positive rating and the pool of ads does not contain ads that were previously disliked by the user. The second approach has the advantage of narrowing down the search space and simplifying the rating prediction operation which leads to faster execution time. Walking through the same example above, table 4 shows the same calculations but while excluding disliked ads. Notice that the record for historical ad 3 is excluded from the computations.

Table 4. Item-based filtering while excluding disliked ads

Historical ad	Rating	New ad 1	New ad 2	New ad 3
Ad 1	1	0.9	0.8	0.4
Ad 2	1	0.7	0.6	0.3
Average Rating		0.8	0.7	0.35

Notice the simpler calculations and fewer records in table 4 which leads to less execution time. The results are consistent that new ad 3 should not be recommended (or receive lowest priority). However, there have been a slight reordering of preferences where new ad 1 shows higher priority than new ad 2, unlike the first approach. Since the recommendation engine sends the top N recommended ads, this does not lead to a huge degradation in precision performance. Our experimentation results in section 6.2 show minimal precision gain by the former approach in general. Figure 32 shows the flowchart for generating predicted advertisement ratings for a specific user.

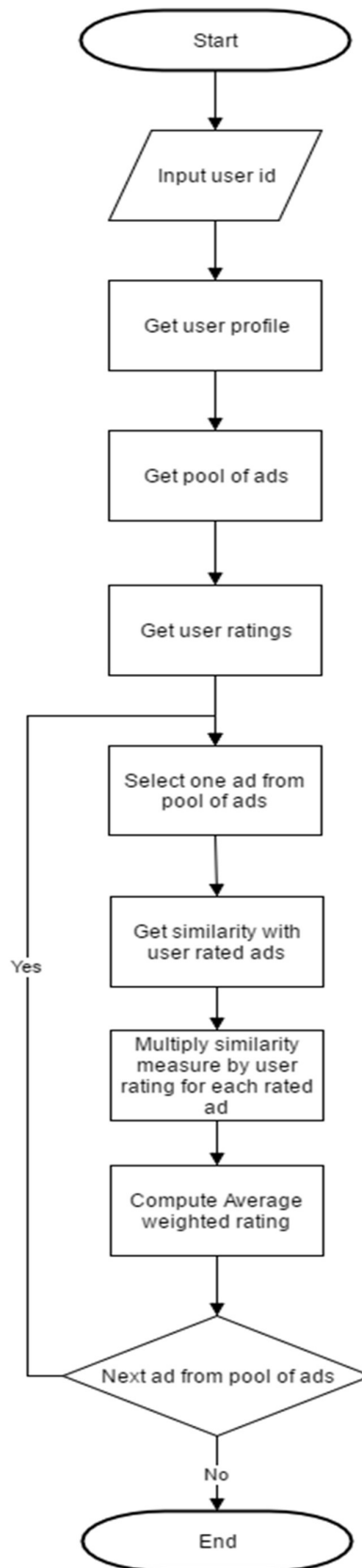


Figure 32. Predicting ad ratings for specific user

4.3. Media Rendering

The rendering engine for the videos (content and advertisements) is built on top of Azure Media Services. Azure Media Services provides storage for content, streaming and dynamic packaging. Dynamic packaging allows for adaptation of content according to the bandwidth available to enhance the quality of experience.

4.4. Storage

The back end storage utilizes a No-SQL document database (MongoDB). This allows for a big data storage as well as a dynamic schema capable of adapting to future system requirements. Several other options have been considered such as Relational Databases (such as MSSQL) and No-SQL Columnar storage (such as HBASE and Cassandra). No-SQL document data bases have been selected as they allow complex data structures with embedded objects and array attributes while maintaining the flexibility of partitioning and indexing collections and performing complex queries. Out of the different No-SQL document databases, such as MongoDB, Microsoft Azure DocumentDB, Amazon SimpleDB and others, MongoDB was chosen due to its maturity and complex querying capabilities in addition to the ability of performing map/reduce operations on its collections. The following diagram shows the main entities maintained by the system and their relationships.

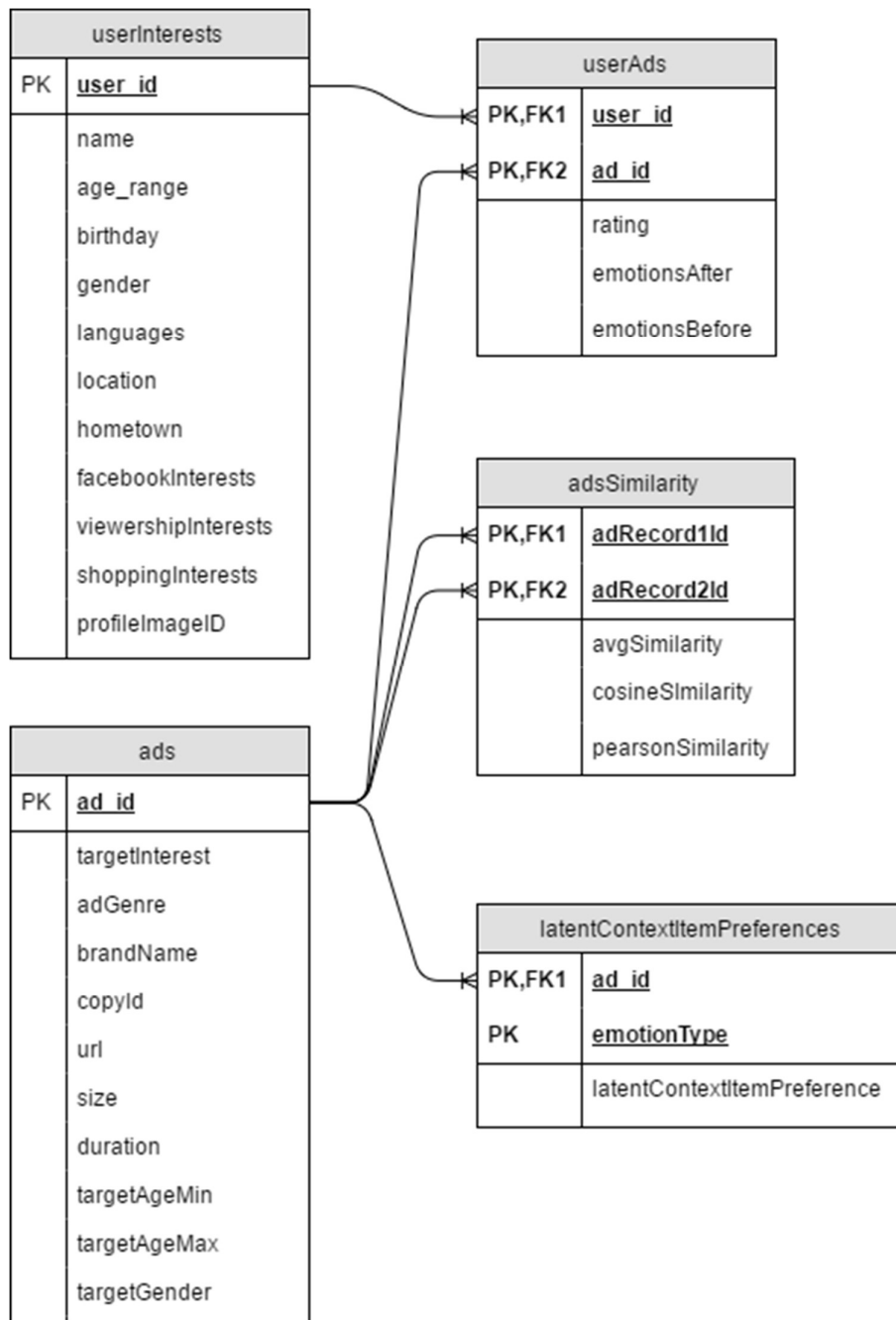


Figure 33. System ER Diagram

4.5. Back-end processes

A nightly job was created to perform 4 main tasks: create ad vectors, create ads similarity table, index the ads similarity table and computing the latent context-item preferences. The overall flow of the main tasks can be visualized in the following flow chart.

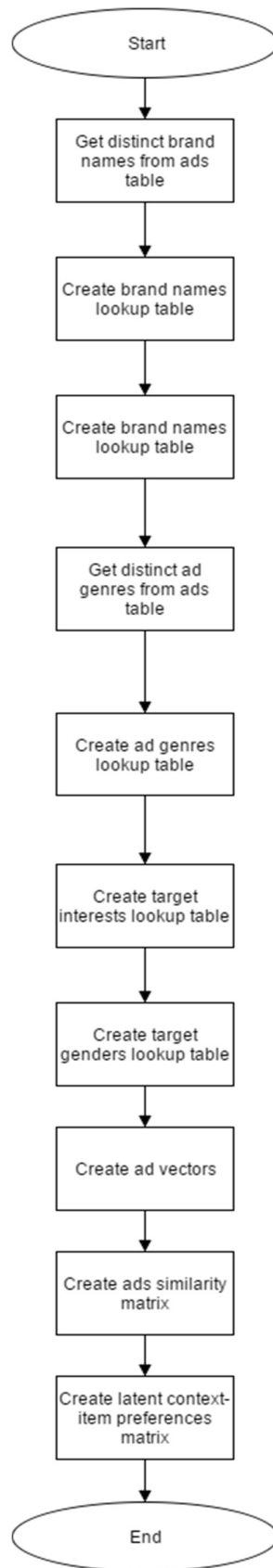


Figure 34. Summary of back-end process

4.5.1. Create ad vectors

This task is concerned with creating numerical vectors representing each ad. It first creates lookup tables for all the nominal attributes such as brands, target genders, target interests, and ad genre and replaces them with a numerical identifier in the vector representing each advertisement. It also creates a set of users who liked the ad and another set of users who disliked the ad. The output of this process is saved in a table that has the same row count of the advertisements table. The following table lists the attributes in the ad record after transformation.

Table 5. Advertisement attributes

Attribute	Type	Description
_id	String	Ad identifier
targetInterest	String	Eg. Home & Garden
adGenre	String	Eg. Drama
brandName	String	Name of product brand
size	Float	Size of file in megabytes
duration	Integer	Ad video duration in seconds
targetAgeMin	Integer	Minimum age for target audience
targetAgeMax	Integer	Maximum age for target audience (null if no upper age limit)
targetGender	String	Gender of target audience for this ad (Males, Females,

		or All)
usersLiked	Array	List of user ids who liked this advertisement (rating = 1)
usersDisliked	Array	List of user ids who did not like this advertisement (rating = 0)
brandId	Integer	A numeric identifier from brands lookup table
targetInterestId	Integer	The corresponding target interest identifier
adGenreId	Integer	The corresponding ad genre identifier
genderId	Integer	The corresponding target gender identifier

Depending on the similarity measure to use, a subset of the attributes in the above table are selected for computation. Detailed identification of the separate vectors for each similarity measure technique is detailed in appendix III. The flow for creating ad vectors can be explained by the below diagram.

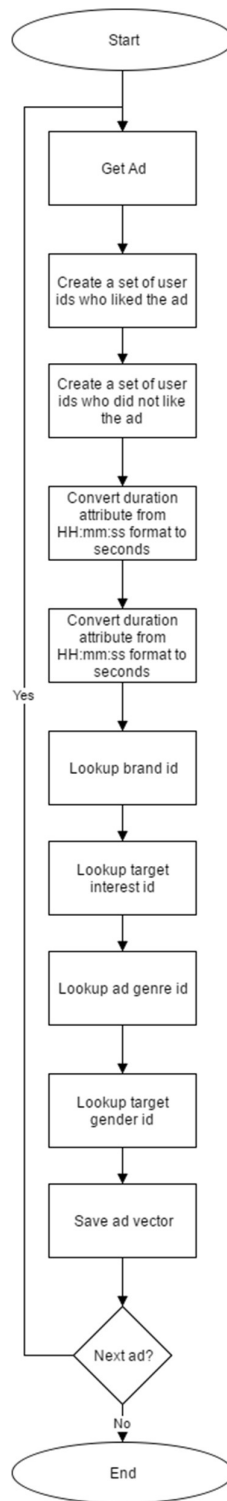


Figure 35. Flow chart for advertisement vectorization

4.5.2. Ads similarity matrix creation

In this task, two inner loops are executed over the ad vectors created in the ads vectorization task to compute the similarities between each pair of ads. Assuming N ads, the output of this

tasks is N^2 records. For each pair of ads, three similarity measures have been computed. One measure is the average similarity between ads. This measure employs the first ad vectorization technique described in the proposed approach section above. In this approach, nominal attributes are compared against each other and produce the number 1 if they are equal or 0 if they are not equal. For set attributes, users liked and users disliked, a jaccard similarity measure is computed to represent each set. For numerical attributes, the cosine similarity measure is computed. The output similarity measure for the pair of ads is the average similarity value of all their attributes. The other two similarity measures are the Pearson coefficient and cosine distance between each pair of ads. These two measures employ the second advertisement vectorization technique described in the proposed approach section above. This table is dropped and re-created at each execution. The following flow chart show the process flow for creating the ads similarity matrix.

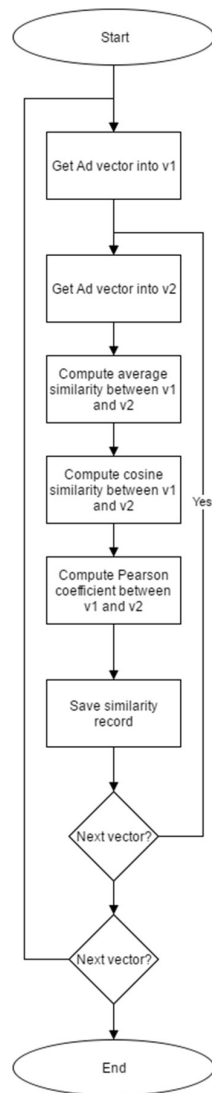


Figure 36. Flow chart for ad similarity matrix computation

The average similarity measure computation is elaborated in the following diagram.

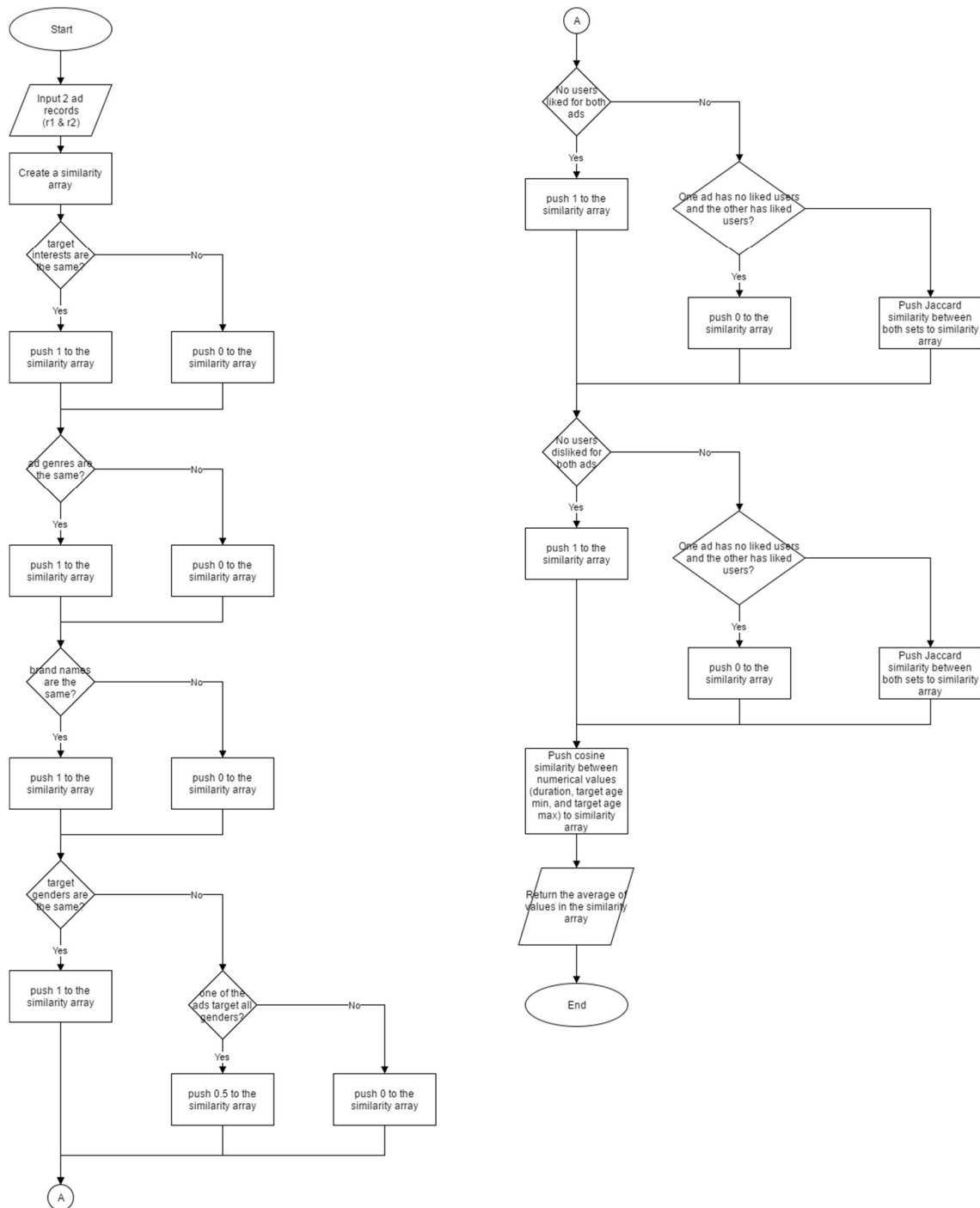


Figure 37. Average similarity measure computation

Cosine similarity measure and Pearson coefficient are computed in the same manner which is shown in the following flow chart.

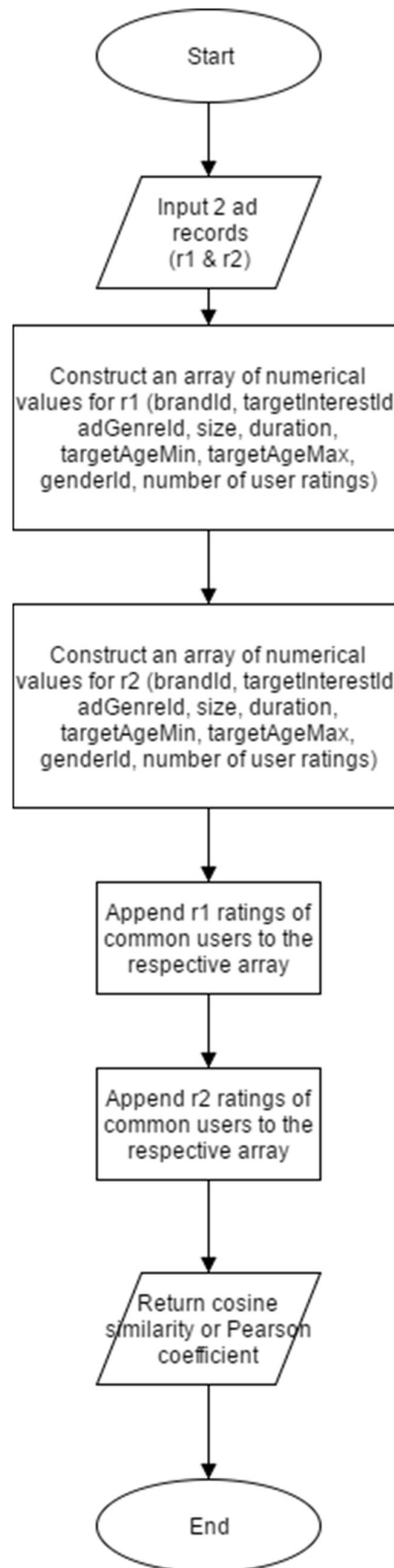


Figure 38. Cosine similarity and Pearson coefficient computation flowchart

4.5.3. Ad similarity matrix index creation

To speed up the search in run-time, the ads similarity matrix is indexed for faster querying. The created index is based on the advertisement ids, sorted ascendingly, and the similarity measure, sorted descendingly.

4.5.4. Latent context-item preferences matrix creation

To find the latent context-item preferences, a context-item matrix is created which contains the frequency by which each item has been consumed under a certain context by all users. This matrix is then normalized so that each context vector is of unit length. The resulting normalized context-item matrix is then multiplied by the item-item similarity matrix (ads similarity matrix) to produce the latent-context item preferences matrix. This approach is borrowed from the authors in [48]. The process for computing latent context-item preferences is shown in the below diagram.

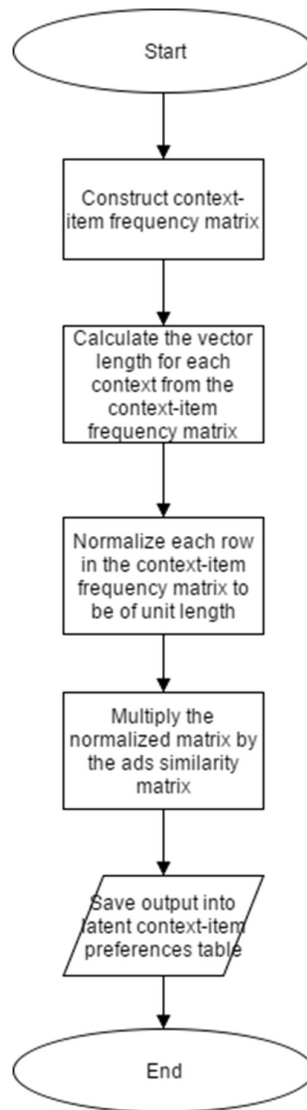


Figure 39. Latent context-item preferences computation

4.6. Data Gathering

Due to the fact that recommendation engines require historical transactional data to be able to compute a similarity matrix on it (whether similarity between users or similarity between items), a sample web application was developed where users can watch the ads and provide feedback on whether they liked or disliked an ad. A corpus of 400+ ads was gathered and is presented to users to evaluate. To simulate the fact that not all users watch all ads, only a small random sample of ads is presented to users (10 samples). Upon completion of the first batch of ads, users are offered to evaluate more ads. For each ad, the user identifies the various emotions under which s/he would like to view the ad. Users are also asked to specify their viewership and shopping interests, similar to what they would do if they are creating a

profile in the main application. To be able to link different responses to the same user and avoid showing the same advertisement for evaluation, a Facebook sign in is required to link multiple responses to the same user. The same application was used for two purposes:

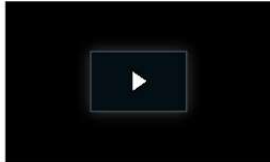
1. The application is used to gather a corpus for training and computing the ads similarity matrix and latent context-item preferences
2. During experimentation, the same application served as simulating historical transactions for users as well as evaluating the percentage precision of random generation of advertisements. This served as a benchmark to compare the effectiveness of our application.

The output is stored in a No-SQL database. In the future, stricter measures will be implemented such as encrypting stored data at rest to ensure user's privacy. The below is a screenshot from the data gathering web application.

- ☐ Fashion
- ☐ Finance
- ☒ Food & Drink
- ☐ Games
- ☐ Hobbies & Leisure
- ☒ Home & Garden
- ☒ Internet & Telecom
- ☐ Jobs & Education
- ☐ Law & Government
- ☐ Online Communities
- ☒ People & Society
- ☐ Pets & Animals
- ☒ Real Estate
- ☐ Sports
- ☒ Travel

For each advertisement below, please specify whether you liked or disliked the ad.

Brand Name: Dettol



Do you like the ad?

☐ I liked ☐ I didn't like

Under which emotions would you like to see this ad (select all that apply):

☐ Happiness ☐ Sadness ☐ Neutral

Brand Name: Al Tayeb



Do you like the ad?

☐ I liked ☐ I didn't like

Under which emotions would you like to see this ad (select all that apply):

☐ Happiness ☐ Sadness ☐ Neutral

Brand Name: Al Hanem Ghie



Do you like the ad?

☐ I liked ☐ I didn't like

Under which emotions would you like to see this ad (select all that apply):

☐ Happiness ☐ Sadness ☐ Neutral

Brand Name: Ceramica Cleopatra



Do you like the ad?

☐ I liked ☐ I didn't like

Under which emotions would you like to see this ad (select all that apply):

☐ Happiness ☐ Sadness ☐ Neutral

Brand Name: Magdy Ya3koup



Do you like the ad?

☐ I liked ☐ I didn't like

Under which emotions would you like to see this ad (select all that apply):

☐ Happiness ☐ Sadness ☐ Neutral

Brand Name: Dreem



Do you like the ad?

☐ I liked ☐ I didn't like

Under which emotions would you like to see this ad (select all that apply):

☐ Happiness ☐ Sadness ☐ Neutral

Brand Name: Life Park



Do you like the ad?

☐ I liked ☐ I didn't like

Under which emotions would you like to see this ad (select all that apply):

☐ Happiness ☐ Sadness ☐ Neutral

Brand Name: 57357 Hospital



Do you like the ad?

☐ I liked ☐ I didn't like

Under which emotions would you like to see this ad (select all that apply):

☐ Happiness ☐ Sadness ☐ Neutral

Figure 40. Snapshot of the data gathering application

Chapter 5: Evaluation and Experimentation

A set of experiments have been designed to decide on the approach to follow in the implementation and to evaluate the quality of service and quality of experience of the implemented system.

5.1. Evaluation Criteria

The goal of the system is to supply users with relevant ads in a timely manner. The timeliness of the application indicates the service level agreement or quality of service (QoS). The relevance of the advertisements indicates the precision of the recommendation engine itself and pertains to the quality of experience (QoE). QoS and QoE can be measured independently of one another as they measure different objectives (timeliness versus. relevance respectively) and each has different metrics to use. In addition, it is also necessary to quantify the quality of context (QoC). Since our recommendation engine is based on contextual information, poor quality of context may lead to inaccurate recommendations. The following are the evaluation criteria that are proposed to be used:

1. Quality of Service (QoS): measure the amount of time taken to gather context, recommend advertisements and deliver the advertisements. Detailed description of the metrics used to measure the QoS aspects of the system can be found in the Experimentation section.
2. Quality of Experience (QoE): To measure the effectiveness of the recommendation engine, feedback from the application will be gathered based on the ads that were watched till their end. This implies that users liked the ad. If users chose to skip an ad, then it means that users disliked the ad. A log of each presented advertisement to each user will be tracked. This way the QoE can be measured by dividing the completely watched ads by the total presented ads and averaged over all users (described by equation 1 below). This feedback mechanism will allow the recommendation engine to continuously improve on its performance. In the future a confidence measure on how much a user liked the advertisement can be added by taking into consideration the attention span of the user. Tracking the yaw angle of viewers over a time span will inform the application how much time the viewer was actually looking at the advertisement. The ratio between attentive time span and total advertisement duration

will provide a confidence measure on how much the user liked the ad (equation 2).

The following equations describe the used measures in detail.

$$QoE \text{ (without confidence)} = \frac{\text{recommended} \cap \text{watch till the end}}{\text{recommended}} \quad \text{eq(1)}$$

$$\text{rating (with confidence)} = \text{watched till the end} * \frac{\text{attention duration}}{\text{total ad duration}}, \text{ where watched till the end} \in \{0,1\} \quad \text{eq(2)}$$

3. Quality of Context (QoS): for all the context sources and enrichment services introduce a probability of error to the system. We will study the performance of individual components that contribute to the collection and enrichment of contextual information and report on their performance results. Description of the metrics used for each component is detailed in the Experimentation section.

5.2. Metrics

In this project, three metrics are defined to measure the different aspects of the system.

1. Turnaround time: for all tasks triggered on the client side, the average turnaround time will be measured and reported in milliseconds. Turnaround time is defined by the time taken from the moment a request is triggered till the return of the result. This measurements includes network communication time and server execution time. Turnaround time measures the quality of service exhibited by the system.
2. Execution time: for all the tasks that do not include a network latency overhead, the average execution time will be measured in milliseconds.
3. Percentage precision: the precision measure will be used to evaluate the quality of experience exhibited by the system. Precision is calculated by the number of ads recommended by the system and liked by the users divided by the total number of recommendations produced by the system. Percentage precision is computed by multiplying 100 to the precision measure.

$$\text{Percentage Precision} = \frac{\text{recommended} \cap \text{interested}}{\text{recommended}} \times 100 \quad \text{eq(3)}$$

5.3. Recommendation engine modeling

There are two implementation decision choices that needed to be performed regarding the recommendation engine itself. The first decision pertains to whether we should include disliked ads (ads with rating=0) while predicting ratings for new ads or exclude them from the computation. If we include disliked ads in the modeling, it gives the system a richer set of

information when predicting ratings for new advertisements - as the more similar a new ad is to a disliked ad, the less likely it will receive a high predicted rating. However, including disliked ads in the computations increases the search space (and, therefore, the execution time) and may require the user to enter more ratings to adapt to the user's interest. For example, if a user likes multiple advertisement copies for a certain brand but did not like a specific advertisement copy, this disliked copy will continue to receive high predicted ratings and may show up in the top recommendations because it is very similar to a lot of ads the user liked. The other approach is to filter out disliked ads from the calculations, focusing only on advertisements liked by the user and filtering out previously disliked ads from the pool of ads to predict a rating for. This way, advertisements that are more similar to ads liked by the user will receive a high predicted rating while ads that are dissimilar will receive a lower predicted rating. The execution time of the second approach is lower as the search space is reduced and calculations can be simplified. This is due to the fact that we do not need to multiply rating by similarity measure as we are only focusing on liked ads (where rating = 1). For the purpose of identifying which approach to follow, a set of experiments will be executed given the data collected by the data gathering app (described previously) as a corpus for training and evaluating the two approaches. The decided upon approach will be further evaluated through user surveys. User surveys are still needed as the corpus suffers from sparsity, this sparsity resulted from the fact that the data gathering app presented 10 ads for feedback out of 421 ads in total. This results in a high possibility that the recommended advertisement may be a correct recommendation from the user perspective, but it was not presented to him before in the data gathering app. For this reason, the decided upon approach will be further evaluated using a human experiment.

The second decision that needed to be made is the methodology for advertisement vectorization and similarity measure to use. How to represent the advertisement as a numerical vector, and then perform the appropriate calculation to compute the similarity between one advertisement and the other. The methods we propose in this work is to incorporate historical transactions with content intrinsic properties to form one representative vector of an advertisement that is not vulnerable to the cold start problem. We have implemented three different similarity measurement techniques and two advertisement vectorization techniques. The two vectorization techniques are:

1. Numerical vectorization: this technique replaces all nominal attributes (such as brand name, target gender, etc.) with numerical values. Then it appends the ratings given by

the intersecting set of users to the vector. This approach enables the employment of commonly used similarity measurement techniques such as cosine similarity and Pearson coefficient.

2. Set vectorization: this technique applies a set similarity to each nominal attribute between two ads. For example, if the brand name is the same between two ads, then similarity for brands will be equal to 1; and 0 otherwise. The similarity measure for each attribute is appended to a vector. For the set of user ratings, the Jaccard similarity of users who liked the ad is computed as well as the Jaccard similarity of users who disliked the ad and both measures are appended to the same vector. For the remaining numerical attributes, a cosine similarity measure is computed for all numerical attributes and is appended to the same vector.

The three similarity measures are:

1. Cosine similarity: this measures the cosine of the angle between two vectors - the closer to 1 the more similar are the ads being compared. This measure utilizes the numerical vectorization technique described above.
2. Pearson coefficient: this measures the linear dependence between two vectors and is computed by dividing the covariance between the two vectors by the standard deviation of each vector. This measure also utilizes the numerical vectorization technique.
3. Average similarity: this measure utilizes the set vectorization technique described above and is computed by finding the average similarity of all the attributes of the two ads.

5.3.1. Experiment setup

To be able to identify the approach to follow (whether to model disliked ads or not and which similarity measure to use) three sets of experiments will be conducted. The three sets of experiments represent how the precision of the system behaves with regard to the number of recommendations presented to each user; that is, when the system has low chances of targeting the user, does it produce an accurate recommendation or not and, at the same time, how does it behave when given more opportunities to target the user. In the first set of experiments, a simulation of recommending one advertisement to each user will be conducted and evaluated against the set of ads rated positively the user through the data gathering app. In the second set of experiments, a simulation of recommending 5 ads to each user will be

conducted. In the third set of experiments, a simulation of recommending 10 ads to each user will be conducted. In each set of experiment, the two recommendation modeling methodologies will be tested (whether to include disliked ads or not). For each methodology, the three similarity measurements will be tested as well to understand if one similarity measure behaves differently depending on the recommendation modelling technique employed. This leads to a total of 18 experiments to be conducted (3 recommendation set sizes x 2 modeling methodologies x 3 similarity measures). In each of the 18 experiments, the resulting percentage precision will be computed to identify the most suitable combination to follow and further test in a human survey. A summary of the list of experiments configurations are listed in the table below.

Table 6. List of experiments for deciding on the recommendation engine implementation methodology

Recommendation set size	Recommendation model methodology	Similarity measure
1 recommendation	Filter out disliked ads	Cosine similarity
		Pearson coefficient
		Average similarity
	Model disliked ads	Cosine similarity
		Pearson coefficient
		Average similarity
5 recommendations	Filter out disliked ads	Cosine similarity
		Pearson coefficient
		Average similarity
	Model disliked ads	Cosine similarity
		Pearson coefficient
		Average similarity

10 recommendations	Filter out disliked ads	Cosine similarity
		Pearson coefficient
		Average similarity
	Model disliked ads	Cosine similarity
		Pearson coefficient
		Average similarity

5.4. Back-end performance

We also need to measure the back-end performance time and how does it scale depending on the number of advertisements available. Since item-based filtering is employed, then the computation times are dependent on the number of products or items available not the number of users in the system [36]. In the back-end we have two main processes. The first process calculates the similarity matrix between ads. For each advertisement, it computes how similar it is to all other advertisements. If we have N ads, this computation results in an $N \times N$ matrix that is stored in the employed NoSQL database - since we have a total of 421 ads in the corpus, this leads to a 421×421 matrix. To be able to identify the trend of the execution time for this process, the execution time for this process will be measured for a set of 8 different values of $N \in \{50, 100, 150, 200, 250, 300, 350, 400\}$.

The second back-end process is concerned with calculating the latent context-item preferences matrix. This matrix maps the different context values to different items and each cell contains a measure of how preferable it is to watch that item under a certain context. We will use this matrix for post-filtering the recommended set of advertisements based on the current emotion context. We will focus on 3 emotions values (happy, sad and neutral). Accordingly, the output matrix of this process will be a $3 \times N$ matrix - where N is the number of ads. The computation of this matrix is a three step process. The first step is to compute how many times each item has been consumed under a certain context (producing a $3 \times N$ matrix). The second step is to normalize the output of the first step so each row representing a context vector will have a magnitude of 1. The third step is to multiply the normalized matrix

by the ads similarity matrix to produce the latent-context item matrix. In essence, the third step weighs the preference of consuming this item under a certain context by how were similar items consumed under the same context. This approach is one of the approaches that were applied by the authors in [48]. Since this process involves multiplying by the ads similarity matrix, it will also be evaluated on the same different sizes.

5.5. Front-end performance

To evaluate the quality of service of the system, the execution time for all the various components running on the client side will be computed. For each component, 15 different measurements will be taken and an average will be computed to accommodate for the variance in network latencies. The various components are listed below:

1. Location extraction: we will measure the total turnaround time for detecting the user location (as GPS coordinates) and enriching it to a region level using the reverse geocoding API.
2. Photo capturing: how much time does the system take to capture a photo for the viewer(s) to be later processed for context detection. This will be measured as execution time on the client's device in milliseconds.
3. Extracting facial context: this measures the turnaround time to detect faces in the captured photo and identifying the gender, age and yaw angle of each face.
4. Identifying persons: this measures the turnaround time to identify the persons presented in the detected faces. This helps the system identify nomadic users.
5. Emotion detection: this measures the turnaround time to detect emotions of detected viewers.
6. Context-based recommendation generation: after the context is gathered, we need to measure how much time does it take to generate recommendations based on the input context.
7. Login-based recommendation generation: for logged in users, how much time does it take to generate recommendations based on pre-stored user profile.

5.6. Human survey

A survey was developed for evaluating the drawn conclusions from the recommendation modelling experiments. This experiment will evaluate the application's capability to produce the right recommendations to users. In addition, the survey will be used to measure the

effectiveness of employing emotion context. The survey will include 10 users who are asked to supply some ratings through the previously developed data gathering application. The precision of recommendations supplied by the data gathering application are recorded. For each user, the application will show a recommended ad based on his historical transactions and profile then ask the user on whether he liked the ad or not. It will then show a happy video and capture the user's context. It will then present an advertisement to the user followed by 3 questions:

1. Did you like the ad? (yes or no)
2. Was the advertisement relevant to your current mood? (yes or no)
3. Under which emotions do you recommend others to watch the advertisement? (happy, sad or neutral)

The third question will be used as a validation technique to filter out contradicting answers of the same user (e.g. watching an ad in a happy mood and recommending it for only a sad mood). Then the application will show a sad video, capture the user's context, show an ad based on the new detected context then present the same 3 questions above to the user. The output of the survey will be a percentage precision for recommendations based on login credentials and a percentage precision for recommendations based on context detected.

Figure 41 is a snapshot of the application used for the human survey. To test the effectiveness of targeted ads, a variance of the application was developed which utilizes user demographic information and recommends a set of random ads that target the same viewer profile – without incorporating the user's historical consumption. The result of this experiment was also used to compare the effectiveness of our system against a typical ad targeting system.



Figure 41. Snapshot of web experiment application.

5.7. Quality of context

In addition to the precision of the recommendation engine, we also need to measure the quality of context gathered as it is used as input to our recommendation engine. For the various context parameters gathered, each will be evaluated using its own metric. The list of metrics used for each contextual parameter is presented in the following table.

Table 7. Metrics used for contextual parameters

Contextual Parameter	Description	Measurement technique
Location context	Was location at the district/region level captured correctly?	Percentage of times the location was captured correctly
User age	Was the user age captured correctly?	Mean absolute error

User age-range	Was the user age-range captured correctly?	Percentage of times the age-range was identified correctly
User gender	Was the user gender identified correctly?	Percentage of times the gender was recognized correctly
Emotion context	Was the user emotion captured correctly?	Percentage of times the emotion was identified correctly
Happy emotion recall	Identifying the capability of the API to identify a happy emotion	Percentage accuracy
Sad emotion recall	Identifying the capability of the API to identify a sad emotion	Percentage accuracy

As presented in the table above, most items are evaluated using the percentage accuracy metric. The only exception is for the user age which is measured using the Mean Absolute Error (MAE) metric. MAE is calculated by summing the absolute errors and dividing by the number of samples. User age-range is also captured since advertisements are targeted to an age range not a specific age. Detected ages of users will be placed in bands to compute the percentage accuracy of the API in placing users in the correct age range. Capability of capturing emotion context is measured from three perspectives. The first is the overall accuracy of the API. The other two metrics pertain to the recall measure for each emotion type. This shows the capability of the application to - for example - identify a sad emotion when it is faced with one.

Chapter 6: Results and discussions

All experiments have been conducted. A corpus of 421 ads have been collected for evaluation from a media agency. These are all public ads that were displayed during one month of the year 2016 on satellite TV. This corpus was used for training and evaluation of our system. The following subsections explain the results of the individual contributions.

6.1. Contextual parameters

As explained in the related works section, we have listed the appropriate contextual parameters that contribute to better advertisement recommendation. The list can be found in tables 1 and 2 in sections 2.1 and 2.2 respectively. Out of all the applicable parameters, our system focused on user context parameters. We also created an application that is capable of capturing the selected parameters from various contextual sources (social network and camera device). A compiled list of all the parameters can be found in table 8. The bold items in user context category are the ones included in our system.

Table 8. List of applicable contextual parameters

Context Category	Context Attributes			
User Context	Age	Activities	Gender	Agenda
	Identity	Emotions	Consumption history	Location
	Profile/Preferences	Needs		
Service Context	Current program	Brand information	Location	Current content viewership rating/Actual viewership
	Promotion	Content description	Access rights	
	Content language	Content format		
Computing Context	Network and bandwidth	Device type	Device status	Operating system
Physical Context	Surrounding environment	Nearby objects	Weather	

History Context	Time			
-----------------	------	--	--	--

6.2. Recommendation model approach

To simulate historical transactions, the data gathering application described previously in the implementation section was used to gather feedback from users on random ads. For evaluating the recommendation modeling approach and similarity measure to use, the experiment has been executed by gathering ad ratings for 40 users. The approaches were tested through simulations of generating recommendations for each user then checking if it was in the list of positively rated ads or not.

Table 9. Experiments for implementation methodology results

Experiment	Measure	Precision	Experiment	Measure	Precision
Precision of sending one recommendation while excluding disliked ads	Cosine Similarity	2.56%	Precision of sending one recommendation while modeling disliked ads	Cosine Similarity	2.56%
	Pearson Coefficient	2.56%		Pearson Coefficient	2.56%
	Average Similarity	30.77%		Average Similarity	33.33%
Precision of sending 5 recommendations while excluding disliked ads	Cosine Similarity	2.05%	Precision of sending 5 recommendations while modeling disliked ads	Cosine Similarity	2.05%
	Pearson Coefficient	2.05%		Pearson Coefficient	2.05%
	Average Similarity	15.38%		Average Similarity	15.38%
Precision of sending 10	Cosine Similarity	2.31%	Precision of sending 10	Cosine Similarity	2.31%

recommendations while excluding disliked ads	Pearson Coefficient	1.54%	recommendations while modeling disliked ads	Pearson Coefficient	1.54%
	Average Similarity	10.51%		Average Similarity	10.51%

The above table lists the results of calculating the percentage precision under different configurations. From the results, we can conclude that filtering out disliked ads does not impact the precision of the recommendation engine in general. We can also realize that the best similarity measure is the Average Similarity we propose which employs the set vectorization technique. The percentages are low in general because users rated an average of 10 out of 421 ads and the recommendation set was based on all 421 ads. This leads to the possibility that the user might like a recommendation but it was not presented in the data-gathering app before. This highlights the need to perform the human experiment described above to further test the effectiveness of the system in general; however, the results of this experiment were necessary to decide on the recommendation model implementation approach for our system.

6.3. QoS results of the solution

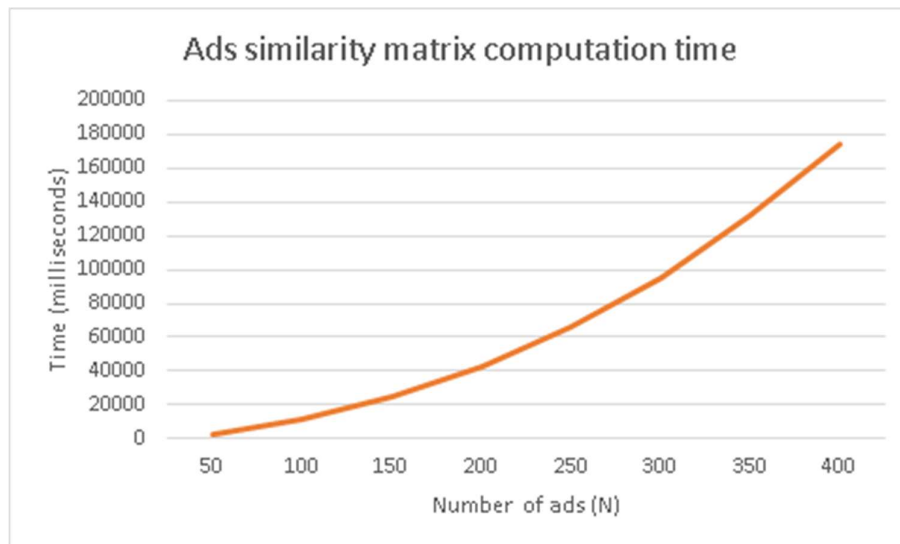


Figure 42. Ads similarity matrix creation time

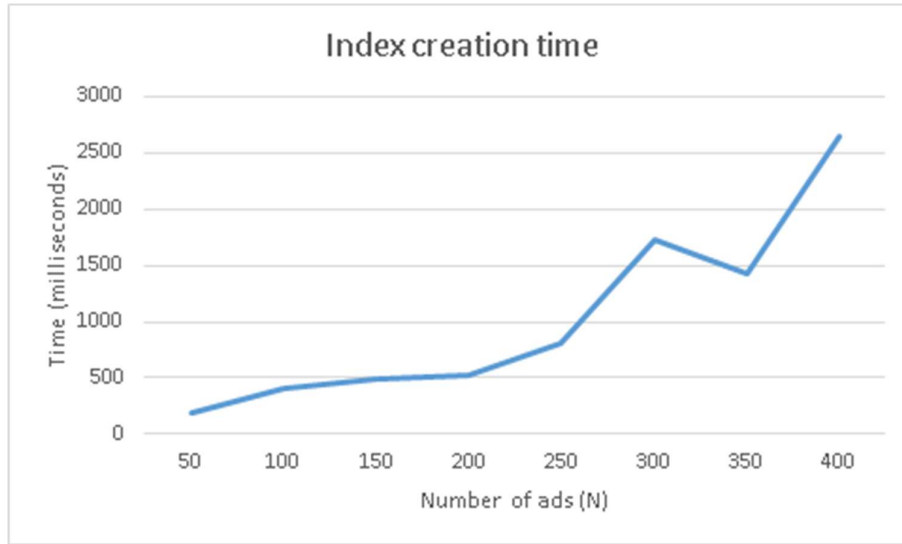


Figure 43. Ads similarity matrix index creation time

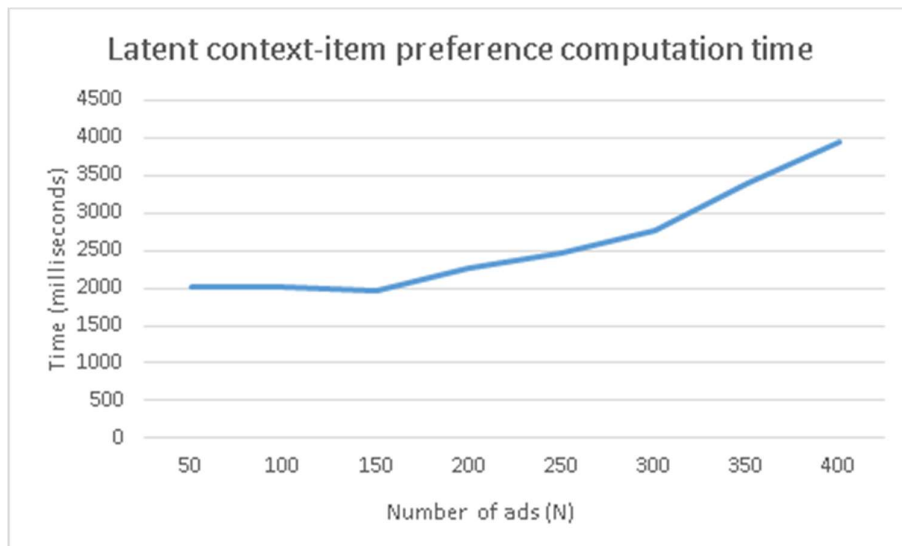


Figure 44. Latent context-item preferences computation time

The performance of the back-end processes have been measured and can be summarized in the above charts. The back end processes were executed on a 4-core virtual machine with 7GB RAM. No parallelism or partitioning has been performed on the database level. Due to the fact that the similarity matrix is of the size $N \times N$, it is expected that the performance of the matrix creation would be N^2 and the chart is consistent with our expectations. Despite the fact that this performance can be enhanced, we do not find this a major concern as 400 ads is a real dataset (ads that were actually displayed throughout a month) and took around 3 minutes to compute. In addition, this is an offline process that does not affect the system in real-time.

Index creation time seems to follow a polynomial pattern as well. Latent context-item preference computation time grows linearly with the number of ads and it took around 4 seconds to compute this matrix for 400 ads.

Table 10. QoS - client components execution times

Component	Description	Average execution time (milliseconds)
Detect Location	Detecting the current GPS coordinates of the user	179
Enrich Location	Transforming the GPS coordinates to a region	218
Capture Photo	Triggering the camera device and capturing a photo of the viewers.	3107
Detect Emotion	Using the captured photo to detect emotions.	1637
Detect Faces	Using the captured photo to detect faces and demographic attributes (age, gender, angle of view)	3071
Identify Persons	Takes as input the list of detected faces and returns person ids	1322
Context Gathering	Total time for all the above tasks. This is not a summation as the above tasks are performed in parallel.	7361
Recommendation based on context information turnaround time	Time taken by the system to generate recommendations based on the detected context.	674

Recommendation based on logged in user information turnaround time	Assuming recommendations are generated based on user profile who denied access to all devices.	444
--	--	-----

The front-end application is a web application that is published on a shared infrastructure on the cloud. The execution time and turnaround time for components running in real-time are summarized in the table above. We can see it takes around half a second to generate recommendations in real-time. Context gathering does consume the majority of the time with a critical path existing in the photo capturing, followed by face detection then identifying persons, which on average takes around 7 seconds. Context gathering and recommendation generation happens as a background thread (implicitly) on the viewing device while the user is watching his/her content; and therefore, the user experience is not affected nor interrupted.

6.4. QoE results of the solution

The user survey was conducted on a sample of 9 users, 5 females and 4 males. The ages of the sample users ranged from 25 to 70. A summary of the results of the experiment are presented in following table.

Table 11. QoE - Precision of different recommendation methodologies

Measure	Percentage Precision
Random ads	41.11%
Targeted ads	51.39%
Login-based recommendation	69.70%
Happy context-based recommendation	77.78%
Sad context-based recommendation	50%
Overall context based recommendation	69.23%
P (ad suitable for current mood liked ad)	100%

We realize that 9 users is a low number for this experiment. To be able to gain more data points, each user rated an average of 7 advertisements in each scenario (7 random ads, 7 targeted ads and 7 login-based recommended ads). This led to having at least 60 ratings for each advertisement generation technique scenario. As described in the Data Gathering application and Human Survey sections, the ratings of the users while simulating historical transactions were recorded and the percentage precision was computed. When sending random ads to users, 41.11% of recommendations were actually liked by the users. To filter out the effect of ad targeting from ad recommendation, another experiment was done which pushes targeted ads to users. Targeted ads are ones that are pushed based on user demographics (age, gender, etc.) without taking historical transactions into consideration. Our results showed that targeted ads are better than random ads and exhibit a percentage precision of 51.39%. The results show that the system was capable of presenting the right ad recommendations to users based only on their profile and historical transactions (login-based recommendation) with a 77.78% precision. This is a significant improvement over random ads generation which is the current experience for satellite TV and IPTV systems. This also shows a significant improvement over pushing targeted ads that rely solely on user demographic attributes. When introducing emotion context, the system was capable of improving its precision in a happy context scenario but degraded its performance in the sad context scenario. This is because the training data gathered by the data-gathering app was not balanced for both scenarios. In general, the overall system performance did not get affected much by introducing context. However, the results show that a context-based recommendations are always suitable for the users' current mood – $P(\text{ad suitable for current mood} \mid \text{liked ad})$. One respondent was recommended a coffee ad by the application within the sad context scenario and said “a coffee or chocolate ad are most suitable to lift my mood when I am sad”. For the disliked ads recommended based on context, users showed that they did not like the ad in general and would not want to view it even under a different mood.

6.5. QoC results of the solution

The user survey conducted was also used to calculate metrics related to the quality of context. Measurements for the different contextual parameters were taken while users were performing the survey. The following table lists the different metrics defined that pertain to the quality of context and their results.

Table 12. QoC - Experiment results

Contextual Parameter	Measurement technique	Result
Location context	Percentage of times the location was captured correctly	100%
User age	Mean absolute error	4.42 years
User age-range	Percentage of times the age-range was identified correctly	88.89%
User gender	Percentage of times the gender was recognized correctly	100%
Emotion context	Percentage of times the emotion was identified correctly	72.22%
Happy emotion recall	Percentage accuracy	100%
Sad emotion recall	Percentage accuracy	44.44%

Age ranges were split between 15-45, 46-65, and 66+. These ranges identify different purchasing groups. We can see from the above results that the application was capable of identifying the correct age range 89% of the time with MAE 4.42 years. Location context (region or district level) has been correctly identified consistently. User gender is also identified consistently for all users. The emotion detection API showed capability in capturing evident emotions. Since expression of happiness consistently exhibits certain curvature in the lips or showing of teeth, it was correctly identified 100% of the time. However, sad emotions were not always expressed with evident manipulation of facial muscles, so users who did not inverse the curvature of their lips when expressing sad emotion were not captured. This is explained by the low recall for sad emotions - 44%. This analysis

can be confirmed by experimenting with sample public images from the API [page](#) itself. Overall, the Emotion detection API exhibited a percentage accuracy of 72%.

The following image shows a sample of a correctly identified happy face.



Figure 45. Correctly identified happy face.

The image below shows a correct sample of a neutral face.



Figure 46. Correctly identified neutral face

Here is an image showing a sad face that was correctly identified.



Figure 47. Correctly identified sad face

Below is a sample sad face that was not identified correctly by the emotion detection API, rather it was classified as neutral.



Figure 48. Unidentified sad face

We can see from the above images that the emotion detection API requires evident manipulation of the facial muscles to be able to capture the emotion correctly. Happy emotions are usually easily expressed by users, however it is not always correct that users express sadness in such an extreme manner. Our analysis conclude that we can rely on the captured context in case a happy or sad emotions are detected, but should be ignored otherwise.

6.6. Other results

While the recommendation model produces the list of recommendations to show with priorities on which advertisement to show first, either based on user interests or emotion context relevance, it does not answer the question of when to show an ad. The

implementation of our system gathered the yaw angle of viewers at frequent intervals and only displays ads when users are consistently looking at the screen. The exact definition of consistency is left of business requirements, but in general it is a threshold by which we decide that the viewer is giving attention to the content being displayed. This feature was tested and worked correctly when setting yaw angle thresholds between -20 and +20 degrees.

The capability of detecting a user profile using face identification was also employed and tested. The tests ran correctly for all 9 users in the survey, but since this is a relatively a small number of users for this problem domain, further testing is required on a larger scale. This helps supporting users in nomadic situations.

The implementation section described how we support multiple viewers watching the same screen. Due to limitations on the number of different viewers, few tests were run to test this feature and the results are positive. However, further experimentation is needed with more different faces, and varying number of viewers. All experiments were run on a laptop device with front-facing camera with narrow angle of view. This led to the constraints that users are sitting very close to the screen and a maximum of two viewers can be detected at a time due to the narrow angle of view of the camera. Further tests need to be run on a simulated environment where users are sitting farther from the screen.

Chapter 7: Conclusion

Through this course of work, we have studied the various sub-problems related to this domain to survey the applicable approaches in each sub-problem. We have also identified a set of user context information that we used as input to our recommendation modelling approach. We were also able to build a system that is capable of gathering the identified contextual information with high reliability and reasonable performance. When it comes to the recommendation model itself, we experimented with various approaches and identified the most suitable approach for this problem domain. Various quality aspects of this domain were identified and measured to prove the viability of our systems.

We have built a system that is capable of generating advertisement recommendations to users with adequate precision. We were also able to incorporate contextual parameters to further enhance the relevance of ad recommendations to users. The results prove that we can target ads to users by a 70% precision, this is a significant improvement over randomly sending ads to users which has a 41% precision. When it comes to emotion context, the experimental results prove that recommended ads are always suitable to the user's current mood when employing emotion context. We have also measured the quality of context which proved to be reliable except for capturing the sad emotion that had percentage recall 44%.

Appendices

Appendix I - API References

In this section we list the set of used API's and links to their specifications. These are the API's that are used for context raw data gathering and context enrichment.

Table 13. API references

API	Description	URL
Emotion API	Takes as input an image and returns a set of emotion probabilities for each detected face	https://www.microsoft.com/cognitive-services/en-us/emotion-api
Face API	Takes an image as input and returns a set of detected faces as well as the detected attributes for each face (age, gender, head pose and others)	https://www.microsoft.com/cognitive-services/en-us/face-api
Face identification API	Takes a set of detected faces as input and returns a person ID for each face	https://westus.dev.cognitive.microsoft.com/docs/services/563879b61984550e40cbbe8d/operations/563879b61984550f30395239
Reverse Geocoding	Takes GPS coordinates as input and returns a human readable address	https://developers.google.com/maps/documentation/geocoding/intro#ReverseGeocoding
Camera device	Triggers the camera device to capture a video	https://developer.mozilla.org/en-US/docs/Web/API/MediaDevices/getUserMedia

Location services	Returns the detected GPS coordinates of the viewing device	https://developer.mozilla.org/en-US/docs/Web/API/Geolocation/getCurrentPosition
-------------------	--	---

Appendix II - Detailed experimental readings

The following tables lists the detailed questionnaire results for each subject. The first table shows readings related to login based precision and quality of context gathered

Table 14. Detailed experimental readings for quality of context

ID	True Gender	Quality of context			
		Correct Location	Correct age range?	Age Absolute Error	Correct gender?
1	Female	Yes	Yes	5.1	Yes
2	Male	Yes	Yes	2.7	Yes
3	Female	Yes	Yes	5.6	Yes
4	Male	Yes	No	4	Yes
5	Female	Yes	Yes	2.3	Yes
6	Female	Yes	Yes	5.8	Yes
7	Male	Yes	Yes	2.4	Yes
8	Male	Yes	Yes	11.9	Yes
9	Female	Yes	Yes	0	Yes

The next table shows the detailed experimental results for random ads recommendation, targeted ads recommendations and login based recommendations

Table 15. Detailed readings for different recommendation techniques

ID	Random ads		Targeted ads		Login-based	
	Liked	Total rated	Liked	Total rated	Liked	Total rated
1	4	10	4	7	6	7

2	5	17	3	7	5	10
3	4	9	4	7	6	15
4	2	7	3	7	0	0
5	7	10	6	10	1	1
6	4	9	5	10	8	9
7	3	9	4	6	11	13
8	4	9	4	9	1	1
9	4	10	4	9	8	10

The next table shows the results of the experiments for happy context scenario.

Table 16. Experimental results for happy emotion context scenario

ID	Was happy emotion captured correctly?	Did you like the ad?	Was the ad suitable for your current mood?	Under which emotions do you recommend to show this ad?
1	Yes	No		Don't like to see the ad at all
2	Yes	Yes	Yes	Happy, Neutral
3	Yes	Yes	Yes	Happy, Sad, Neutral
4	Yes	No		Don't like to see the ad at all
5	Yes	Yes	Yes	Happy, Sad, Neutral
6	Yes	Yes	Yes	Happy, Neutral
7	Yes	Yes	Yes	Happy
8	Yes	Yes	Yes	Happy, Sad

9	Yes	Yes	Yes	Happy, Sad, Neutral
---	-----	-----	-----	------------------------

The next table shows the same results but under the sad context scenario.

Table 17. Experimental results for sad emotion context scenario

ID	Was sad emotion captured correctly?	Did you like the ad?	Was the ad suitable for your current mood?	Under which emotions do you recommend to show this ad?
1	Yes	Yes	Yes	Happy, sad, neutral
2	No			No recommendation tested
3	No			No recommendation tested
4	No			No recommendation tested
5	Yes	No		Don't like to see the ad at all
6	Yes	Yes		Happy, sad, neutral
7	Yes	No		Don't like to see the ad at all
8	No			No

				recommendation tested
9	No			No recommendation tested

Appendix III - Detailed advertisement vectors representation

Similarity vector attributes that employ set vectorization can be explained by the list below. The resulting similarity measure between two advertisements is the average similarity of all their underlying attributes.

Table 18. Detailed advertisement vector attributes using average similarity measure

Attribute	Type	Similarity measure
Target interest	nominal	1 if both ads have the same value, 0 otherwise
Ad genre	nominal	1 if both ads have the same value, 0 otherwise
Brand name	nominal	1 if both ads have the same value, 0 otherwise
Target gender	nominal	1 if both ads have the same value, 0.5 if one of the ads targets all genders, 0 otherwise
Users liked	Array	Compute jaccard similarity between two sets
Users disliked	Array	Compute jaccard similarity between two sets
Duration in seconds	integer	Compute cosine similarity between both integer vectors
Target minimum age	integer	
Target maximum age	integer	
Final similarity	float	Find the average value of all the above similarities

The calculations can be explained through the below example of two ad records for which we need to compute their average similarity based on their set vectorization representation.

Table 19. Example of set similarity computation using set vectorization

Attribute	Ad 1	Ad 2	Similarity
Target interest	Electronics	Fashion	0
Ad genre	Comedy	Drama	0
Brand name	Samsung	Boss	0

Target gender	Males	Males	1
Users liked	[User1,User2]	[User3,User1]	0.333
Users disliked	[User3,User5]	[User5,User4]	0.333
Duration in seconds	30	30	Cosine similarity = 1
Target minimum age	18	18	
Target maximum age	45	45	
Final similarity	= average of the 7 similarity measures above		0.381

The following table shows the vector representation in case cosine or Pearson similarity measures are used.

Table 20. Advertisement vector representation in case of Cosine Similarity or Pearson Coefficient measures

Attribute	Type
Brand ID	integer
Target interest ID	integer
Target gender ID	integer
Ad genre ID	integer
Size in megabytes	float
Duration in seconds	integer
Target minimum age	integer
Target maximum age	integer
Number of user ratings	integer
Common user 1 rating	float
....	float

Common user N rating; where N is the number of users who rated both ads	float
---	-------

Notice that the above approaches integrate content intrinsic properties with historical transactions. This way when a new ad is introduced to the system, the set vectorization will assign 0 similarity for users liked set and users disliked set attributes and will compute the average similarity based only on the content intrinsic properties. The numerical vectorization technique will also exclude common user ratings and will only rely on the content intrinsic properties. As a result, this technique is not vulnerable to the cold-start problem.

Appendix IV – Context representation

Below is a listing of the JSON object that represents the contextual data captured by the client application. This is the object that is sent to the back-end recommendation engine to be used as input calculations. In case the user did not log in, the `loggedInUser` object will be omitted.

```
{
  "loggedInUser": {
    "id": "134500391",
    "name": "Youssef Youssef",
    "age_range": {
      "min": 21
    },
    "birthday": "07/31/1983",
    "email": "youssefy@aucegypt.edu",
    "education": [],
    "gender": "male",
    "hometown": {
      "id": "115351105145884",
      "name": "Cairo, Egypt"
    },
    "first_name": "Youssef",
    "languages": [
      {
        "id": "103755242996777",
        "name": "Egyptian Arabic"
      },
      {
        "id": "106059522759137",
        "name": "English"
      }
    ],
    "last_name": "Youssef",
    "locale": "en_US",
    "location": {
      "id": "115351105145884",
      "name": "Cairo, Egypt"
    },
    "isLoggedIn": true,
    "facebookInterests": [ ...],
  },
}
```



```

"viewshipInterests": {
  "Action": true,
  "Adventure": true,
  "Comedy": true,
  "Crime": false,
  "Drama": false,
  "Fantasy": false,
  "Historical": false,
  "Horror": false,
  "Romance": false,
  "Science fiction": true
},
"shoppingInterests": {
  "Arts & Entertainment": false,
  "Autos & Vehicles": true,
  "Beauty & Fitness": true,
  "Books & Literature": false,
  "Business & Industrial": false,
  "Computers & Electronics": true,
  "Fashion": true,
  "Finance": false,
  "Food & Drink": true,
  "Games": true,
  "Hobbies & Leisure": true,
  "Home & Garden": false,
  "Internet & Telecom": true,
  "Jobs & Education": false,
  "Law & Government": false,
  "Online Communities": false,
  "People & Society": true,
  "Pets & Animals": false,
  "Real Estate": true,
  "Sports": true,
  "Travel": true
}
},
"cameraContext": [
  {
    "faceId": "af53183f-c023-4281-90f0-ce102af827f3",
    "faceRectangle": {
      "top": 163,

```

```

    "left": 268,
    "width": 182,
    "height": 182
  },
  "faceAttributes": {
    "headPose": {
      "pitch": 0,
      "roll": -9.5,
      "yaw": -3.6
    },
    "gender": "male",
    "age": 36.6
  },
  "profileImageID": "f0cb6a70-0b3c-4235-be88-502d429a1f54",
  "viewshipInterests": {
    "Action": true,
    "Adventure": true,
    "Comedy": true,
    "Crime": false,
    "Drama": false,
    "Fantasy": false,
    "Historical": false,
    "Horror": false,
    "Romance": false,
    "Science fiction": true
  },
  "shoppingInterests": {
    "Arts & Entertainment": false,
    "Autos & Vehicles": true,
    "Beauty & Fitness": true,
    "Books & Literature": false,
    "Business & Industrial": false,
    "Computers & Electronics": true,
    "Fashion": true,
    "Finance": false,
    "Food & Drink": true,
    "Games": true,
    "Hobbies & Leisure": true,
    "Home & Garden": false,
    "Internet & Telecom": true,
    "Jobs & Education": false,

```

```

    "Law & Government": false,
    "Online Communities": false,
    "People & Society": true,
    "Pets & Animals": false,
    "Real Estate": true,
    "Sports": true,
    "Travel": true
  },
  "facebookInterests": [ ... ],
  "name": "Youssef Youssef",
  "gender": "male",
  "birthday": "07/31/1983",
  "email": "youssefy@aucegypt.edu",
  "hometown": {
    "id": "115351105145884",
    "name": "Cairo, Egypt"
  },
  "location": {
    "id": "115351105145884",
    "name": "Cairo, Egypt"
  },
  "languages": [
    {
      "id": "103755242996777",
      "name": "Egyptian Arabic"
    },
    {
      "id": "106059522759137",
      "name": "English"
    }
  ]
},
"emotionContext": [
  "happiness"
],
"locationContext": "Al Matar, Qism El-Nozha, Cairo Governorate, Egypt"
}

```


References

- [1] Kodialam, Murali, et al. "Online scheduling of targeted advertisements for IPTV." *INFOCOM, 2010 Proceedings IEEE*. IEEE, 2010.
- [2] Cankaya, Hakki. "System and method for providing context-aware local advertising in IPTV systems." U.S. Patent Application 11/311,838.
- [3] Xu, David Jingjun, Stephen Shaoyi Liao, and Qiudan Li. "Combining empirical experimentation and modeling techniques: A design research approach for personalized mobile advertising applications." *Decision Support Systems* 44.3 (2008): 710-724.
- [4] Dao, Tuan Hung, Seung Ryul Jeong, and Hyunchul Ahn. "A novel recommendation model of location-based advertising: Context-Aware Collaborative Filtering using GA approach." *Expert Systems with Applications* 39.3 (2012): 3731-3739.
- [5] Hristova, Nataliya, and Gregory MP O'Hare. "Ad-me: wireless advertising adapted to the user location, device and emotions." *System Sciences, 2004. Proceedings of the 37th Annual Hawaii International Conference on*. IEEE, 2004.
- [6] "Spark". <http://spark.apache.org/>
- [7] "Weka". <http://www.cs.waikato.ac.nz/ml/weka/>
- [8] Zemek, Radim, Songbo Song, and Hassnaa Moustafa. "Context-Awareness for IPTV Services Personalization." *Media Networks: Architectures, Applications, and Standards* (2012): 401.
- [9] Songbo, S. O. N. G. *Advanced personalization of IPTV services*. Diss. Institut National des Télécommunications, 2012.
- [10] Song, Songbo, Hassnaa Moustafa, and Hossam Afifi. "IPTV Services Personalization Using Context-Awareness." *Informatica: An International Journal of Computing and Informatics* 36.1 (2012): 13-20.
- [11] Song, Songbo, Hassnaa Moustafa, and Hossam Afifi. "Advanced IPTV services personalization through context-aware content recommendation." *Multimedia, IEEE Transactions on* 14.6 (2012): 1528-1537.
- [12] Song, Songbo, Hassnaa Moustafa, and Hossam Afifi. "Personalized TV service through employing context-awareness in IPTV/IMS architecture." *Future Multimedia Networking*. Springer Berlin Heidelberg, 2010. 75-86.

- [13] Dabrowski, Marek, et al. "A context-aware architecture for IPTV services personalization." *Journal of Internet Services and Information Security (JISIS)* 3.1/2 (2013): 49-70.
- [14] Doolin, Kevin, et al. "Context-aware multimedia services in a pervasive environment: the Daidalos approach." *Proceedings of the 2008 Ambi-Sys workshop on Software Organisation and MonIToring of Ambient Systems*. ICST (Institute for Computer Sciences, Social-Informatics and Telecommunications Engineering), 2008.
- [15] Arnaud, Julien, et al. "Adaptive IPTV services based on a novel IP Multimedia Subsystem." *Multimedia Tools and Applications* 55.2 (2011): 333-352.
- [16] Muntean, Cristina Hava, and G. Muntean. "Framework for interactive personalised IPTV for entertainment." *W3C WWW Conference, Edinburgh, Scotland, UK, May*. 2006.
- [17] Mas, Ignacio, et al. "IPTV session mobility." *Communications and Networking in China, 2008. ChinaCom 2008. Third International Conference on*. IEEE, 2008.
- [18] Dietze, Stefan, and John B. Domingue. "Towards context-aware multimedia processing through semantic web services." *Proceedings of the seventh european conference on European interactive television conference*. ACM, 2009.
- [19] Xu, David Jingjun, Stephen Shaoyi Liao, and Qiudan Li. "Combining empirical experimentation and modeling techniques: A design research approach for personalized mobile advertising applications." *Decision support systems* 44.3 (2008): 710-724.
- [20] Chellouche, Soraya Ait, Julien Arnaud, and Daniel Négru. "Flexible user profile management for context-aware ubiquitous environments." *Consumer Communications and Networking Conference (CCNC), 2010 7th IEEE*. IEEE, 2010.
- [21] Verdaguer, Xavier, et al. "Recommendation system and method for multimedia content." U.S. Patent Application No. 11/974,561.
- [22] Lum, Wai Yip, and Francis CM Lau. "A context-aware decision engine for content adaptation." *IEEE Pervasive computing* 1.3 (2002): 41-49.
- [23] Gantner, Zeno, et al. "MyMediaLite: A free recommender system library." *Proceedings of the fifth ACM conference on Recommender systems*. ACM, 2011.
- [24] Thyagaraju, G. S., and Umakanth P. Kulkarni. "Design and implementation of user context aware recommendation engine for mobile using Bayesian network, fuzzy logic and rule base." *International Journal of Pervasive Computing and Communications* 8.2 (2012): 133-157.

- [25] Park, Won-Ik, Sanggil Kang, and Young-Kuk Kim. "A personalized multimedia contents recommendation using a psychological model." *Computer Science and Information Systems* 9.1 (2012): 1-21.
- [26] Peled, Jill, and Anneken Tappe. "Impact of Targeting." Apr. 2010: n. pag. Rpt. in *ARF - Knowledge at Hand*. N.p.: n.p., n.d. Web.
<<http://www.warc.com/Content/ContentViewer.aspx?ID=4f0b4bad-7fb2-4b38-8ae2-e35f9f44fde4>>.
- [27] Kitts, Brendan. "How to Target TV Ads: A Buyer's Guide to Set Top Box Targeting Algorithms." Proc. of ARF Experiential Learning: Re:Think Conference. N.p., 2013. Web. 3 June 2016.
<<https://www.warc.com/Content/ContentViewer.aspx?MasterContentRef=75a91202-6498-4b2e-a1f7-64ba2fb187c4&q=How+to+target+TV+ads%3a+A+buyers+guide+to+set+top+box+targeting+algorithms&CID=A98967&PUB=ARF-KIF>>.
- [28] Sohn, Sang Mok. "ADVERTISEMENT PLATFORM DEVICE AND OPERATING METHOD THEREOF." U.S. Patent No. 20,160,063,555. 3 Mar. 2016.
- [29] Linden, Lee Charles, and Benjamin Lewis. "Expanded tracking and advertising targeting of social networking users." U.S. Patent Application No. 14/460,219.
- [30] Dean, Jeffrey A., Georges R. Harik, and Paul Bucheit. "Methods and apparatus for serving relevant advertisements." U.S. Patent No. 7,716,161. 11 May 2010.
- [31] McConnell, Ted. "The Programmatic Primer: Measurement in the Online Advertising Ecosystem." *Warc Exclusive* July 2015: n. pag. Web. 3 June 2016.
<<https://www.warc.com/Content/ContentViewer.aspx?MasterContentRef=075f82ac-8e42-4c83-98a2-35eff64cec48&q=Measurement+in+the+online+advertising+ecosystem&CID=A101825&PUB=WARC-EXCLUSIVE>>.
- [32] Li, Yung-Ming, Lienfa Lin, and Shih-Wen Chiu. "Enhancing Targeted Advertising with Social Context Endorsement." *International Journal of Electronic Commerce* 19.1 (2014): 99-128.
- [33] Johnson, Justin P. "Targeted advertising and advertising avoidance." *The RAND Journal of Economics* 44.1 (2013): 128-144.
- [34] Koumaras, Harilaos, et al. "A QoE-aware IMS infrastructure for multimedia services." *ICUMT*. 2011.

- [35] Kim, Jinsul. "Bandwidth Allocation Optimization for IPTV QoS/QoE Prediction & Measurement." (2008).
- [36] Segaran, Toby. *Programming collective intelligence: building smart web 2.0 applications*. "O'Reilly Media, Inc.", 2007.
- [37] Huang, Jiajin, Ning Zhong, and Yiyu Yao. "A UNIFIED FRAMEWORK OF TARGETED MARKETING USING CUSTOMER PREFERENCES." *Computational Intelligence* 30.3 (2014): 451-472.
- [38] Chen, Guanling, and David Kotz. *A survey of context-aware mobile computing research*. Vol. 1. No. 2.1. Technical Report TR2000-381, Dept. of Computer Science, Dartmouth College, 2000.
- [39] Demiris, Thanos. "Context revisited: a brief survey of research in context aware multimedia systems." *Proceedings of the 3rd international conference on Mobile multimedia communications*. ICST (Institute for Computer Sciences, Social-Informatics and Telecommunications Engineering), 2007.
- [40] Hwang, San-Yih, and Wan-Shiou Yang. "Discovering Generalized Profile-Association Rules for the Targeted Advertising of New Products." *INFORMS journal on Computing* 20.1 (2008): 34-45.
- [41] da Silva, Fábio Santos, Luiz Gustavo Pacola Alves, and Graça Bressan. "PersonalTVware: A proposal of architecture to support the context-aware personalized recommendation of TV programs." *European Interactive TV Conference (EuroITV 2009)*, Leuven, Belgium. 2009.
- [42] Stuart, Elnora W., Terence A. Shimp, and Randall W. Engle. "Classical conditioning of consumer attitudes: Four experiments in an advertising context." *Journal of consumer research* (1987): 334-349.
- [43] Adomavicius, Gediminas, and Alexander Tuzhilin. "Context-aware recommender systems." *Recommender systems handbook*. Springer US, 2011. 217-253.
- [44] Verbert, Katrien, et al. "Context-aware recommender systems for learning: a survey and future challenges." *IEEE Transactions on Learning Technologies* 5.4 (2012): 318-335.
- [45] Linda, Sonal, and Kamal K. Bharadwaj. "A Fuzzy Trust Enhanced Collaborative Filtering for Effective Context-Aware Recommender Systems." *Proceedings of First International Conference on Information and Communication Technology for Intelligent Systems: Volume 2*. Springer International Publishing, 2016.

- [46] Zheng, Yong, Bamshad Mobasher, and Robin Burke. "Emotions in context-aware recommender systems." *Emotions and Personality in Personalized Services*. Springer International Publishing, 2016. 311-326.
- [47] Abbas, Assad, Limin Zhang, and Samee U. Khan. "A survey on context-aware recommender systems based on computational intelligence techniques." *Computing* 97.7 (2015): 667-690.
- [48] M. F. Alhamid, M. Rawashdeh, H. Dong, M. A. Hossain and A. E. Saddik, "Exploring Latent Preferences for Context-Aware Personalized Recommendation Systems," in *IEEE Transactions on Human-Machine Systems*, vol. 46, no. 4, pp. 615-623, Aug. 2016.
- [49] Datta, Suman, et al. "SCARS: A scalable context-aware recommendation system." *Computer, Communication, Control and Information Technology (C3IT), 2015 Third International Conference on*. IEEE, 2015.
- [50] Hong, Liang, et al. "Context-Aware Recommendation Using Role-Based Trust Network." *ACM Transactions on Knowledge Discovery from Data (TKDD)* 10.2 (2015): 13.
- [51] Unger, Moshe. "Latent Context-Aware Recommender Systems." *Proceedings of the 9th ACM Conference on Recommender Systems*. ACM, 2015.
- [52] Unger, Moshe, et al. "Towards latent context-aware recommendation systems." *Knowledge-Based Systems* 104 (2016): 165-178.
- [53] Serrano-Guerrero, Jesus, et al. "A google wave-based fuzzy recommender system to disseminate information in University Digital Libraries 2.0." *Information Sciences* 181.9 (2011): 1503-1516.
- [54] Steeg, F. van. "Context-aware recommender systems." (2015).
- [55] Hyung, Ziwon, Kibeom Lee, and Kyogu Lee. "Music recommendation using text analysis on song requests to radio stations." *Expert Systems with Applications* 41.5 (2014): 2608-2618.
- [56] Alhamid, Mohammed F., et al. "Towards context-sensitive collaborative media recommender system." *Multimedia Tools and Applications* 74.24 (2015): 11399-11428.
- [57] Buchholz T , Schiffers M . Quality of context: What it is and why we need it. In: *Proceedings of the 10th workshop of the openview university association: OVUAÓ3*; 2003 .
- [58] Hoyos, José R., et al. "A model-driven approach for quality of context in pervasive systems." *Computers & Electrical Engineering* 55 (2016): 39-58.

- [59] Chabridon, Sophie, et al. "Building ubiquitous QoC-aware applications through model-driven software engineering." *Science of Computer Programming* 78.10 (2013): 1912-1929.
- [60] Lim, Léon, et al. "Enhancing context data distribution for the internet of things using qoc-awareness and attribute-based access control." *Annals of Telecommunications* 71.3-4 (2016): 121-132.
- [61] Marie, Pierrick, et al. "QoCIM: a meta-model for quality of context." *International and Interdisciplinary Conference on Modeling and Using Context*. Springer Berlin Heidelberg, 2013.
- [62] Bu, Yingyi, et al. "Managing quality of context in pervasive computing." *Quality Software, 2006. QSIC 2006. Sixth International Conference on*. IEEE, 2006.
- [63] Henriksen, Karen, Jadwiga Indulska, and Andry Rakotonirainy. "Modeling context information in pervasive computing systems." *International Conference on Pervasive Computing*. Springer Berlin Heidelberg, 2002.
- [64] Manzoor, Atif, Hong-Linh Truong, and Schahram Dustdar. "On the evaluation of quality of context." *European Conference on Smart Sensing and Context*. Springer Berlin Heidelberg, 2008.
- [65] Oppermann, Reinhard, and Marcus Specht. "A context-sensitive nomadic exhibition guide." *International Symposium on Handheld and Ubiquitous Computing*. Springer Berlin Heidelberg, 2000.
- [66] Oppermann, Reinhard, and Marcus Specht. "A nomadic information system for adaptive exhibition guidance." *Archives and museum informatics* 13.2 (1999): 127-138.
- [67] Henriksen, Karen, and Jadwiga Indulska. "Developing context-aware pervasive computing applications: Models and approach." *Pervasive and mobile computing* 2.1 (2006): 37-64.
- [68] Sawers, Mariam Armia Keriakos. *Context-aware goal-oriented business process modeling*. Diss. The American University in Cairo, 2012.
- [69] Hamza, Mostafa. *Feature-based Generation of Pervasive Systems' Architectures Utilizing Software Product Line Concepts*. Diss. The American University in Cairo, 2011.
- [70] Kleinrock, Leonard. "Nomadic computing—an opportunity." *ACM SIGCOMM Computer Communication Review* 25.1 (1995): 36-40.
- [71] Lee, Kun Chang, Namho Chung, and Jeongeun Byun. "Understanding continued ubiquitous decision support system usage behaviour." *Telematics and Informatics* 32.4 (2015): 921-929.

- [72] Pawar, Pravin, et al. "Performance evaluation of the context-aware handover mechanism for the nomadic mobile services in remote patient monitoring." *Computer Communications* 31.16 (2008): 3831-3842.
- [73] Aminzadeh, Nazanin, Zohreh Sanaei, and Siti Hafizah Ab Hamid. "Mobile storage augmentation in mobile cloud computing: Taxonomy, approaches, and open issues." *Simulation Modelling Practice and Theory* 50 (2015): 96-108.
- [74] Ghiani, Giuseppe, et al. "Evaluating context-aware user interface migration in multi-device environments." *Journal of Ambient Intelligence and Humanized Computing* 6.2 (2015): 259-277.
- [75] Hong, Jong-yi, Eui-ho Suh, and Sung-Jin Kim. "Context-aware systems: A literature review and classification." *Expert Systems with Applications* 36.4 (2009): 8509-8522.