



Graduation Project

Prepared by:

Meznah Faleh Alrasheed	436201211
Rand Abdulaziz Aldhwayan	436200370
Reem Saleh Algumaizi	436200324
Rola Saud Aldhwayan	436200047
Shahad Majed Bin Dakhel	436200024

Supervised by:
Dr. Rana abaalkhail

Second Semester 1439/1440
Fall 2018-2019

STUDENT DECLARATION

Student Declaration: Student acknowledges that she has read the policy pertaining to plagiarism and understands that the penalty for such an act could result in disciplinary action.

Project Title: VIBES

Term of the Project: 2nd Semester

Academic Year: 1439-1440 Hijri, 2018-2019 Gregorian

Student's Name		Student's Email	Student's University ID Number	Student's Signature
1	<i>Rola saud Al.Dhwayan</i>	436200047@student.ksu.edu.sa	436200047	
2	<i>Rand abdulaziz Al.Dhwayan</i>	436200370@student.ksu.edu.sa	436200370	
3	<i>Reem saleh Al.Gumaizi</i>	436200324@student.ksu.edu.sa	436200324	
4	<i>Shahad majed Bin.Dakhel</i>	436200024@student.ksu.edu.sa	436200024	
5	<i>Meznah faleh Al.Rasheed</i>	436201211@student.ksu.edu.sa	436201211	

ABSTRACT (IN ARABIC)

لوحظ في السنوات الأخيرة مع التقدم التقني والثورة المعلوماتية انتشار موقع التواصل الاجتماعي، وازدياد أعداد مستخدميها حول العالم [1]، وتحديداً في المملكة العربية السعودية، ونتيجة لهذا التقدم أصبح موقع تويي منصة للتعبير عما يجول في خواطر المجتمع السعودي حول مواضيع كثيرة، وقد أصبح نتيجة لذلك مصدراً غنياً بالمشاعر والأراء يساعد في الكثير من الدراسات وتحليل البيانات والتي يمكن استغلالها في أمور كثيرة منها عكس مشاعر السعوديين تجاه نقطة معينة.

مع انتشار هوس زيارة المطاعم استفاد كلّاً من روادها وعامة الناس من التطبيق، وذلك بمشاركة آرائهم وانطباعاتهم تجاهها، استناداً على ذلك كان توجب علينا البحث المستمر عن دراسات تتعلق بتحليل المشاعر والبيانات عن طريق الاستعانة بأداة تطبيق "تويتر".

إلا أنه ما زال هناك افتقاراً للوجود دراسات تقوم بتحليل مشاعر زوار المطعم، من هنا تولدت وانطلقت فكرة (فليز) وهو تطبيق يعرض مطاعم بعض مدن المملكة بناءً على عملية تحليل تغريدات وتعليقات الشعب السعودي، مع الأخذ بعين اختلاف اللهجات، ثم إظهار نتيجة ذلك عن طريق عرض تقييم المطعم بشكل يمثل تصنيف مشاعر الزوار تجاهه إلى إيجابية (منبر، سعيد، راضي) وسلبية (غاضب، نادم، مسناه) بشكل نسيبي، حيث تتغير نسبة كل مطعم حسب الشعور العام لزوارها.

ABSTRACT (IN ENGLISH)

In recent years, the technical progress and the information revolution have witnessed the spread of social networking sites and increased the number of users around the world/[1], especially in Saudi Arabia. As a result of this progress, Twitter has become a platform for expressing the thoughts of Saudi society in many topics. Therefore, a rich source of feelings and opinions helps in many studies and data analysis that can be exploited in many ways, including the reflection of the feelings of Saudis towards a certain issue. With the spread of restaurant attendance obsession, both its visitors and the public benefited from the application, sharing their views and impressions. Correspondingly, we had to constantly search for studies on emotion and data analysis, either by using Twitter as a tool.

However, there is still absence of studies that analyze the feelings of restaurant visitors, hence the idea of (VIBES), an application that presents restaurants of some cities in the Kingdom based on the analysis of Saudis, taking into consideration the different dialects in the Kingdom. As a result, the restaurants rating of visitors' attitudes towards it is positive (impressed, happy, satisfied) and negative (angry, regret, and dissatisfied). The ratio of each restaurant varies according to the general feeling of its visitors.

TABLE OF CONTENT

Chapter 1 INTRODUCTION	12
1.1 INTRODUCTION	13
1.2 RESEARCH PROBLEM.....	14
1.3 PROJECT GOALS AND OBJECTIVES	14
1.4 THE SOLUTION.....	14
1.5 PROJECT SCOPE	14
1.6 HARDWARE AND SOFTWARE TOOLS	15
1.7 WORK BREAKDOWN STRUCTURE.....	16
Chapter 2 INFORMATION GATHERING	18
2.1 INFORMATION GATHERING TECHNIQUES	19
2.1.1 LITERATURE REVIEW	19
2.1.2 RELATED APPLICATIONS.....	21
2.1.3 COMPARISON BETWEEN THE APPLICATIONS CHARACTERISTICS WITH VIBES	25
2.1.4 INTERVIEWS	26
2.1.5 QUESTIONNAIRE	27
2.2 CONCLUSION AND OUTCOMES	28
Chapter 3 SYSTEM ANALYSIS	29
3.1 DEVELOPMENT METHODOLOGY.....	30
3.2 USER CHARACTERISTICS.....	30
3.3 USER REQUIREMENTS	31
3.3.1 USER FUNCTIONAL REQUIREMENTS	31
3.4 SYSTEM REQUIREMENTS.....	35
3.4.1 SYSTEM FUNCTIONAL REQUIREMENTS	35
3.5 NON-FUNCTIONAL REQUIREMENTS	35
3.6 SYSTEM ANALYSIS MODELS	37
3.6.1 USE CASE DIAGRAM	37
3.4.2 DETAILED USE CASE DESCRIPTIONS	38
3.6.3 SEQUENTIAL DIAGRAMS	64
3.6.4 DESIGN CLASS DIAGRAM	75
Chapter 4 SYSTEM DESIGN	76
4.1 SYSTEM ARCHITECTURE	77
4.2 DATA DESIGN.....	78
4.2.1 ENTITY RELATIONSHIP DIAGRAM	78
4.2.2 DATABASE SCHEMA	79
4.2.3 DATA DICTIONARY	80
4.3 USER INTERFACE DESIGN	83
Chapter 5 SYSTEM IMPLEMENTATION	88
5.1 REQUIRED HARDWARE & SOFTWARE	89
5.1.1 HARDWARE	89
5.1.2 SOFTWARE.....	89
5.2 SYSTEM IMPLEMENTATION.....	90
5.2.1 IMPLEMENTATION PROCESS	90
5.3 SYSTEM INTEGRATION	93
5.4 SAMPLE CODE.....	93
Chapter 6 SYSTEM TESTING	98
6.1 UNIT TESTING	99
6.2 INTEGRATION TESTING.....	100

6.3 PERFORMANCE TESTING	103
6.4 USER ACCEPTANCE TESTING	104
6.4.1 CONCLUSION.....	107
6.5 TEST CASES	107
6.5.1 REGISTRATION – VALID ENTRIES TEST.....	107
6.5.2 REGISTRATION – INVALID ENTRIES TEST.....	108
6.5.3 LOGIN – VALID ENTRIES TEST	108
6.5.4 LOGIN – INVALID ENTRIER TEST	109
6.5.5 CUSTOMER LOCATION ACCESSIBILITY TEST	109
6.5.6 CUSTOMER LOCATION INACCESSIBILITY TEST	110
6.5.7 TRENDING RESTAURANTS AVAILABILITY TEST	110
6.5.8 RESTAURANTS CATEGORIES AVAILABILITY TEST	111
6.5.9 RESTAURANTS CATEGORIES SELECT ABILITY TEST.....	111
6.5.10 EMOTION LIST AVAILABILITY TEST	112
6.5.11 EMOTION SELECT ABILITY TEST.....	112
6.5.12 ANALYSIS TEST - EXISTING TWEETS	113
6.5.13 ANALYSIS TEST - NOT EXISTING TWEETS	113
6.5.14 SEARCH FEATURE TEST:	114
6.5.15 LIST OF RESTAURANTS AVAILABILITY TEST	114
6.5.16 SPECIFIC RESTAURANT SELECT ABILITY TEST	115
6.5.17 RESTAURANT INFORMATION AVAILABILITY TEST	115
6.5.18 ADD COMMENT FEATURE TEST	116
6.5.19 UPLOAD PICTURE FEATURE TEST	116
6.5.20 UPDATE USER INFORMATION WITH VALID ENTRIES TEST	117
6.5.21 UPDATE USER INFORMATION WITH INVALID ENTRIES TEST	117
6.5.22 FAVORITE LIST AVAILABILITY TEST	118
6.5.23 ADD RESTAURANT TO FAVORITE LIST TEST	118
6.5.24 REMOVE RESTAURANT FROM FAVORITE LIST TEST	119
6.5.25 LOGOUT ABILITY TEST	119
Chapter 7 SYSTEM DEMONSTRATION	120
7.1 SYSTEM SCREEN FLOW	121
7.2 SYSTEM SCREENS SNAPSHOTS	122
Chapter 8 CONCLUSION.....	129
8.1 SUMMARY	130
8.2 IMPACT OF THE PROJECT	130
8.3 LIMITATIONS AND FUTURE WORK	130
8.4 LESSONS LEARNED	130
8.5 ACKNOWLEDGMENTS	131
REFERENCES.....	132
APPENDICES.....	135

LIST OF TABLES

Chapter 1 | INTRODUCTION

Table1.1: SW, HW used	15
Table1.2: Work breakdown structure	16

Chapter 2 | INFORMATION GATHERING

Table2.1: Comparison between the applications characteristics with VIBES.....	25
---	----

Chapter 3 | SYSTEM ANALYSIS

Table3.1: View branches information	38
Table3.2: View restaurant categorize	39
Table3.3: View emotions list.....	40
Table3.4: Analyzing	41
Table3.5: Retrieve comments	42
Table3.6: Retrieve Tweets	43
Table3.7: Select branch	44
Table3.8: Determine branch location.....	45
Table3.9: Search for specific restaurant	46
Table3.10: View branches	47
Table3.11: Select restaurant categories	48
Table3.12: Determine customer location.....	49
Table3.13: Select emotions.....	50
Table3.14: Login.....	51
Table3.15: Edit profile.....	52
Table3.16: Registration.....	53
Table3.17: Add branch to list	54
Table3.18: View favorite list	55
Table3.19: Remove branch from list	56
Table3.20: Add comments.....	57
Table3.21: View comments	58
Table3.22: Delete comments	59
Table3.23: Upload pictures.....	60
Table3.24: View pictures.....	61
Table3.25: Delete pictures	62
Table3.26: Logout.....	63

Chapter 4 | SYSTEM DESIGN

Table4.1: Data Dictionary.....	80
Table4.2: Relationship Dictionary	81
Table4.3: Attributes Dictionary	82

Chapter 5 | SYSTEM IMPLEMENTATION

Table5.1:Required hardware.....	89
Table5.2: Required software.....	89

Chapter 6 | SYSTEM TESTING

Table6.1: Unit testing.	99
Table6.2: Integration testing.	100
Table6.3: 1 st user acceptance testing.	104
Table6.4: 2 nd user acceptance testing....	105
Table6.5: 3 rd user acceptance testing.	105
Table6.6: 4 th user acceptance testing.	106
Table6.7: 5 th user acceptance testing.	106
Table6.8: Registration with valid entries test case.	107
Table6.9: Registration with invalid entries test case.	108
Table6.10: Login with valid entries test case.	108
Table6.11: Login with invalid entries test case.	109
Table6.12: Customer location accessibility test case.	109
Table6.13: Customer location inaccessibility test case.	110
Table6.14: Trending restaurant test case.	110
Table6.15: Restaurants categories availability test case.	111
Table6.16: Restaurant categories select ability test case.	111
Table6.17: Emotion list availability test case.	112
Table6.18: Emotion select ability test case.....	112
Table6.19: Analysis with existing tweets test case.	113
Table6.20: Analysis with not existing tweets.	113
Table6.21: Search feature test case.....	114
Table6.22: List of restaurents availability test case.....	114
Table6.23: Specific restaurabt select ability test case.	115
Table6.24: Restaurant information availability test case.....	115
Table6.25: Add comment feature test case.	116
Table6.26: Upload picture feature test case.....	116
Table6.27: Update user information with valid entries test case.	117
Table6.28: Update user information with invalid entries test case.	117
Table6.29: Favorite list availabitiy test case.....	118
Table6.30: Add restaurant to favorite list feature test case.	118
Table6.31: Remove restaurant from favorite list feature test case.	119
Table6.32: Logout ability test case.	119

LIST OF FIGURES

Chapter 1 INTRODUCTION	
Figure1.1: TimeLine	17
Chapter 2 INFORMATION GATHERING	
Figure2.1: Screenshot for FoodMood Application.....	21
Figure2.2: Screenshot for Foursquare Application.....	22
Figure2.3: Screenshot for ParallelDots Website.....	22
Figure2.4: Screenshot for Tweet Sentiment Visualization Website.....	23
Figure2.5: Screenshot for FoodMedicine Application.	23
Figure2.6: Interview question1	26
Figure2.7: Interview question2.....	26
Figure2.8: Interview question3	26
Figure2.9: Questionnaire gender	27
Figure2.10: Questionnaire range of age.....	27
Figure2.11: Questionnaire question 1	27
Figure2.12: Questionnaire question 2.....	27
Chapter 3 SYSTEM ANALYSIS	
Figure3.1: Agile Methodology	30
Figure3.2: A simple illustrate for functional and nonfunctional requirements	36
Figure3.3: Use case diagram.....	37
Figure3.4: View restaurant's categories Sequential Diagrams	64
Figure3.5: View emotion list Sequential Diagrams.....	64
Figure3.6: Retrieve comment Sequential Diagrams	65
Figure3.7: Retrieve tweets Sequential Diagrams.....	65
Figure3.8: Select branch & View branch information Sequential Diagrams	66
Figure3.9: Determine branch location Sequential Diagrams.....	66
Figure3.10: Search for specific restaurant Sequential Diagrams.....	67
Figure3.11: View branches Sequential Diagrams	67
Figure3.12: Select restaurant categories Sequential Diagrams.....	67
Figure3.13: Determine customer location Sequential Diagrams	68
Figure3.14: Select emotions Sequential Diagrams	68
Figure3.15: Login Sequential Diagrams.....	69
Figure3.16: Edit profile Sequential Diagrams	69
Figure3.17: Registration Sequence Diagram	70
Figure3.18: Add branch to list Sequential Diagrams.....	70
Figure3.19: View favorite list Sequential Diagrams	71
Figure3.20: Remove branch from list Sequential Diagrams	71
Figure3.21: Add comments Sequential Diagrams	71
Figure3.22: View comments Sequential Diagrams	72
Figure3.23: Delete comments Sequential Diagrams	72
Figure3.24: Upload pictures Sequential Diagrams	72
Figure3.25: View pictures Sequential Diagrams	73

Figure3.26: Delete pictures Sequential Diagrams	73
Figure3.27: Logout	74
Figure3.28: Design class diagram	75
Chapter 4 SYSTEM DESIGN	
Figure4.1: Three Tier Architecture	77
Figure4.2: Entity relationship diagram	78
Figure4.3: Database schema	79
Figure4.4: Start screen	83
Figure4.5: Registration Screen	83
Figure4.6: Login Screen	83
Figure4.7: Customer Location	84
Figure4.8: Home screen	84
Figure4.9: Restaurants screen	84
Figure4.10: Information screen	85
Figure4.11: Comment screen	85
Figure4.12: Pictures screen	85
Figure4.13: Restaurants for specific emotion screen	86
Figure4.14: Sentiment analysis rate screen	86
Figure4.15: Edit profile screen	86
Figure4.16: Favorites list	87
Figure4.17: Search restaurants	87
Figure4.18: About VIBES	87
Chapter 5 SYSTEM IMPLEMENTATION	
Fiquer5.1: Therri tier	90
Fiquer5.2: Sample of lexicon	91
Fiquer5.3: Restaurant object	92
Fiquer5.4: User object	92
Fiquer5.5: Favorite list object	93
Fiquer5.6: Connection with twitter API	93
Fiquer5.7: Connection with Foursquare API	94
Fiquer5.8: Determine current user location	94
Fiquer5.9: Tweets analysis	95
Fiquer5.10: Tweets pre-processing	96
Fiquer5.11: Classify Restaurants under specific emotion	97
Fiquer5.12: Add specific restaurant to favorite list	97
Chapter 6 SYSTEM TESTING	
Figure6.1: Disk performance	103
Figure6.2: Memory performance	103
Figure6.3: CPU performance	103
Chapter 7 SYSTEM DEMONSTRATION	
Figure7.1: VIBES site map "system screen flow"	121
Figure7.2: VIBES splash screen	122
Figure7.3: Login screen	122
Figure7.4: Sign up screen	122
Figure7.5: Determine current location screen	123

Figure7.6: VIBES home screen.....	123
Figure7.7: Restaurants screen.....	123
Figure7.8: Restaurants under under "منبهـ" emotion.....	124
Figure7.9: Restaurants under under "سعـد" emotion.....	124
Figure7.10: Restaurants under under "راضـي" emotion.....	124
Figure7.11: Search bar.....	125
Figure7.12: Restaurant information screen1.....	125
Figure7.13: Restaurant information scre	125
Figure7.14: Restaurant information screen3.....	126
Figure7.15: Emotion analysis screen.....	126
Figure7.16: Favorite list screen.	126
Figure7.17: Favorite list under "منبهـ" emotion.....	127
Figure7.18: Favorite list under "سعـد" emotion.....	127
Figure7.19: Favorite list under "راضـي" emotion.....	127
Figure7.20: User profile screen.....	128
Figure7.21: Edit profile screen.....	128
Figure7.22: About VIBES screen.	128

Chapter 1 | INTRODUCTION



In this chapter, we represent an overview of our VIBES application project. We start with an introduction that provides summarized description. Then, we state the main problem and propose a solution for it. After that, we determine the goals and objectives that we aim to achieve at the end of this project. We also talk about the scope of project, the project software and hardware tools that will be used. At the end, we display the timeline and the work breakdown structure (WBS) to manage our tasks.

1.1 INTRODUCTION

The Real Business of Restaurants is not food only— it is Emotion [2]

Many studies recognize that emotions play a major role in customer (dis)satisfaction and their subsequent judgments, and more particularly, in customer (dis)satisfaction with service encounters [3]. Recent studies have also an emotions—not just cognition— influence judgment, decision-making, and even post-purchase behavior's [3]. Customers often express their opinions about restaurants based on many different attributes, including location, food quality, service, price and previous experience [4]. Customer's satisfaction is an important factor in marketing that receives huge attention because their satisfaction tends to represent the driving force for restaurant/brand success. Accordingly, if restaurants want to increase their benefits, they have to focus on the true feelings of their customers. In addition, customers' positive emotion will positively influence the number of visitors to the restaurant, and the actual amount of money spent in it [5]. This means "emotional eating" is a specific kind of eating and a specific kind of emotion [6]. Foremost restaurant work optimizes many of the characteristics of service work associated with emotional labor. waiters not only have to create a face of friendly, happy service, but they are also expected to feel happy as they are serving each customer [5].

Social web data like twitter has a large amount of data that people post, so it has become important to work on efficient intelligent systems that can do data refinement, analysis of tasks [7]. Twitter is the most popular social network service that people express their feelings/impressions in. We conduct a survey that aims at both males and females that consists of students and employees in a certain range age group - from 20-35- to analyze their attitude.

In our survey, when we asked the question "Would you like to know people's emotions and opinions about restaurants in some cities in Saudi Arabia before visiting them?" 90.2% of the answers were yes. Based on that, we as a graduation project team, decided to build up an application called "VIBES" that aims to analyze people's emotions based on their tweets, comments in google maps and emoji used. The universe (community) for this study is all adults living in some cities in Saudi Arabia.

1.2 RESEARCH PROBLEM

People sometimes face problems when they want to visit a restaurant and they have no idea about its ambience, as some restaurants can be of low quality or of awful customer services. Moreover, some restaurants have negative atmosphere. This means they are incompatible with their expectations.

1.3 PROJECT GOALS AND OBJECTIVES

The main goal of this study is to provide a user-friendly application that shows restaurants with positive and negative emotions around the user's location or by searching for specific restaurants.

1. Helping restaurants to raise quality and improve their performance.
2. Summarize people's emotions from different sources in one application.
3. Raise the spirit of competition between restaurants.
4. Detect as much as possible Saudis' tweets and comments about restaurants, which are visited, and implement an analysis to classify their emotions.
5. Help the application users to detect the suitable restaurants easily and quickly.
6. Help people to be aware of positive emotions and be more outgoing.

1.4 THE SOLUTION

The suggested solution for the described problem is represented in our project. It is an application that aims to provide restaurants search features based on people's emotions. Providing some (positive) emotions toward the restaurants; which enables other people to find nearby restaurants based on their emotions or by searching for a specific restaurant. Moreover, it allows the customers to share pictures and express their opinions by comments. This solution can be achieved by analyzing people's tweets, comments and emojis - that they used to express their impression or feeling about the restaurants through one of the two online services: Twitter and Google maps application.

1.5 PROJECT SCOPE

We are building an application that utilizes Twitter API and Google maps API to collect current Arabic tweets or comments' text and emojis. To analyze them in some cities in Saudi Arabia (Riyadh, Jeddah, Dammam), which express emotions of people towards restaurants. Our focus will be on the positive emotions such as (Happy, Enjoyable, Exciting, etc.) because these emotions enable users to visit satisfying restaurants. Also, we will take into consideration negative emotion because these emotions will help people to get more honest information about the restaurants. Our project targets users who are using iOS system and residents of some cities in the Kingdom. We chose iOS-operating system to rely on the comparison between iOS and android by apply the same application on them. The comparison focused on two aspects, first they look at lines of code, and iOS better with an average difference of 38%. Second, they attempted to quantify the number of hours spent in coding; the average Android app takes approximately 28% more time to develop than iOS [8].

1.6 HARDWARE AND SOFTWARE TOOLS

Table1.1: SW, HW used

SW	HW
Firebase DataBase	Laptop
Twitter API	iPhone
Google map API	
Adobe XD	
sketchbook	
Draw.io	
000webhost	
Grammarly	
Google Drive	
Office Timeline	
Microsoft office	
Adobe photoshop	

1.7 WORK BREAKDOWN STRUCTURE

Table1.2: Work breakdown structure

Phase	Description of the tasks	Tasks
1. Proposal	Gathering information to support the application VIBES, that needed to develop, design and complete an integrated application containing the required information mentioned in these documents.	1.1 Introduction 1.2 Problem 1.3 Goals and Objectives 1.4 The Solution 1.5 Project Scope 1.6 Timeline 1.7 Roles and responsibilities
2. Introduction		2.1 Introduction 2.2 Problem 2.3 Goals and Objectives 2.4 The Solution 2.5 Project Scope 2.6 Hardware and Software 2.7 Work breakdown structure
3. Information Gathering	Provide detailed information through: <ul style="list-style-type: none"> • Literature review. • Implement a questionnaire that supports user feedback. • Search for similar applications. 	3.1 Information gathering techniques 3.1.1 Literature review 3.1.2 Related applications 3.1.3 Comparison between the applications characteristics with VIBES 3.1.4 Interviews 3.1.5 questionnaire 3.2 conclusion and outcomes
4. System Analysis	<ul style="list-style-type: none"> • Determine functional and non-functional requirements. • Drawing models for system analysis, understanding and modeling system requirements. 	4.1 Development methodology. 4.2 User characteristics 4.3 User and system requirements 4.3.1 Functional requirements 4.3.2 Non-functional requirements 4.4 System analysis models 4.4.1 Use case diagram 4.4.2 Detailed use case descriptions 4.4.3 Sequential diagrams 4.4.4 Design Class Diagram
5. System Design	System structuring which will give an idea about the components. It also includes interfaces design.	5.1 System architecture 5.2 Data design 5.2.1 Entity relationship diagram 5.2.2 Database schema 5.2.3 Data dictionary 5.3 User interface design
6. Conclusion	Summarize the application and the importance of advanced technology as well as the importance of the opinions and emotions of people on restaurant evaluation	6.1 Summary 6.2 Impact of the project 6.3 Limitations and future work 6.4 Acknowledgments

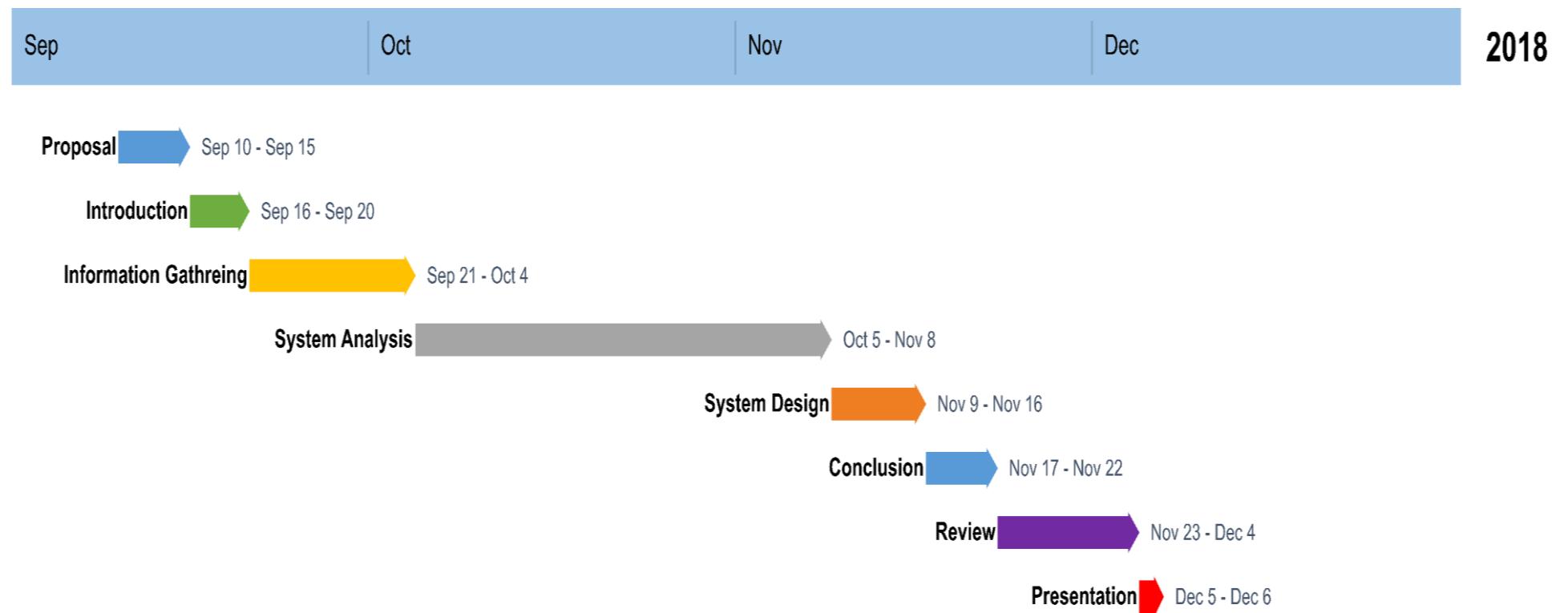


Figure1.1: TimeLine

Chapter 2 | INFORMATION GATHERING



This chapter determines the information gathering techniques that we used in VIBES project. We covered several topics. First, we started with literature review, which presents some existing studies and similar applications, which related to our project. Then we provided a comparison that compared the existing project features with our provided features. After that, we presented an interview that we conducted to measure the restaurants pioneers who are interested in visiting and evaluating different restaurants. Then, we went over the questionnaire that we conducted for this project and discussed the results to obtain useful statistics that support our project. At the end, we summarized the chapter by displaying the main outcomes and findings of our information gathering process.

2.1 INFORMATION GATHERING TECHNIQUES

This section represents three different techniques for gathering project information that includes:

- Literature Review
- Interview
- Questionnaire

2.1.1 LITERATURE REVIEW

Sentiment Analysis (SA) analyzes online texts and statements that describe writers' opinions towards a particular object [9]. In order to evaluate sentiments in sentiment analysis, we needed to detect terms, called opinion bearing words or simply opinion-words, which are the base for sentiment analysis to perform the evaluation. Emotions detection from text provides a strong and expressive opinion evaluation over conventional approaches. Emotion usefulness comes due to its importance in the academic community, in industry, and in a variety of applications such as production, politics, marketing, education, and many more. The main idea is to extract opinions from reviews and to evaluate their sentiments in terms of emotions such as joy, surprise, anger, or fear, to name a few, instead of simply using the conventional positive and negative sentiment evaluation [9]. Social networking sites like Twitter, Facebook, google+ and Instagram are rapidly gaining popularity as they allow people to share and express their views about issues, have discussion with different communities, or post messages across the world. There has been a lot of work in the field of sentiment analysis of twitter data. This project focuses mainly on sentiment analysis of twitter and google maps data, which are helpful to analyze the information in the tweets and comments where opinions are highly structured, and are either positive or negative, or neutral in some cases [10]. In the context of social sciences, research has focused on emoji as a means of expressing emotions on mobile platforms. Emoji usage in social media has increased because its embedding's learned from short descriptions that summarize collection of tweets [11]. Twitter provide a rich repository of images and captions that are associated with users' sentiments [12].

In recent years, a lot of work has been done in the field of “Sentiment Analysis on Twitter” by a number of researchers. In its early stage it was intended for binary classification which assigns opinions or reviews to bipolar classes such as positive or negative only [10].

In the work presented [13] Pak and Paroubek, used a Twitter API to collect a corpus of English tweets and a dataset of three classes was formed: positive, negative and a set of objective texts (no sentiments). They used emoticons (e.g. ‘:-)’ and ‘:D’ for happy, ‘:-(|’ and ‘= (’ for sad) as queries to retrieve a large amount of unlabeled data under the assumption that these tweets are representative of all happy and sad tweets. Tweets were only collected from accounts of popular newspapers and magazines, such as the New York Times and the Washington Post.

Duwairi et al [14] in this paper they deal with Arabic Sentiment Analysis. They developed a framework that make it possible to analyze Twitter comments or "Tweets" as having positive, negative or neutral sentiments. This can be applied in a wide range of applications ranging from politics to marketing. This framework has many novel aspects such as handling Arabic dialects, Arabize and emoticons. Also, crowdsourcing was utilized to collect a large dataset of tweets.

In the work presented by [15] M. Alsulami and Mahmood, the study considers the Ministry of Education as a case study to design a sentiment analysis model to examine tweets related to services provided by the ministry. The aim of this work is to inform decision-makers, the issues related to the performance of the ministry by exploring the users' opinions about their services. In particular, one of the most recent hashtags related to the Ministry of Education is examined which is #New_University_System. The sentiment analysis model aims to categorize the users' attitudes about the new university system that has been launched recently. It will classify the Arabic tweets regarding the posted hashtag into three categories: Positive, Negative, and Neutral. Also, the study investigates the users' expectations about the new service. The results will help decision makers to find the gap between users' needs (reflected by their opinions and expectations) and what the government's expectations in applying the new university system.

In this study Islam MR et al [16] uses sentiment analysis to analyze social network data for user's feelings and sentiments to investigate their moods and attitudes when they are communicating via these online tools. They aim to perform depression analysis on Facebook data collected from an online public source. To investigate the effect of depression detection, they propose machine learning technique as an efficient and scalable method.

The dual aim of Hernández-Fernández DA et al [17] research is, firstly, to analyze the physiological and unconscious emotional response of consumers to a new technological product and, secondly, link this emotional response to consumer conscious verbal reports of positive and negative product perceptions. From a practical point of view, the approach employed in the case study on the product Google Glass provides empirical evidence useful in the decision-making process for this and other technological enterprises launching a new product.

El-Beltagy et al [18] investigated an ML-based sentiment analysis model integrated with a set of features, many of which were extracted using an Arabic sentiment lexicon. They considered the number of segments within the text, the length of the text and emoticons. Their results showed that the presented features demonstrated increased accuracy across six of the seven datasets. They applied their system to Egyptian, Saudi, Levantine and MSA social media datasets. In their research, they claimed to present state-of-the-art Arabic sentiment analysis by developing a model that performed better than all existing Arabic opinion mining systems with publicly available datasets.

Simeon Rebora [19] focus on Wattpad, a social reading platform on which people can add comments in the margins of books. Analyzing these comments enables the comparison between specific parts of the text and the effects they have on readers. They introduce a simplified experiment with the sentiment analysis software Syuzhet. By comparing the “emotional arcs” produced in parallel by text and comments, they evaluate the approach and show the substantial differences between the intrinsic emotional valence of the text and the effects it produces.

2.1.2 RELATED APPLICATIONS



FoodMood

FOODMOOD Application have designed their search engine to make your search experience quick and easy! Dishes will be suggested to you one at a time. also learns from your response (yum or yuk) in real-time and navigates through thousands of opportunities out there while you sit down and relax. FOODMOOD works out your food mood and provides recommendations for restaurants you'll love. Also, find them around with instant access to expert reviews [20].

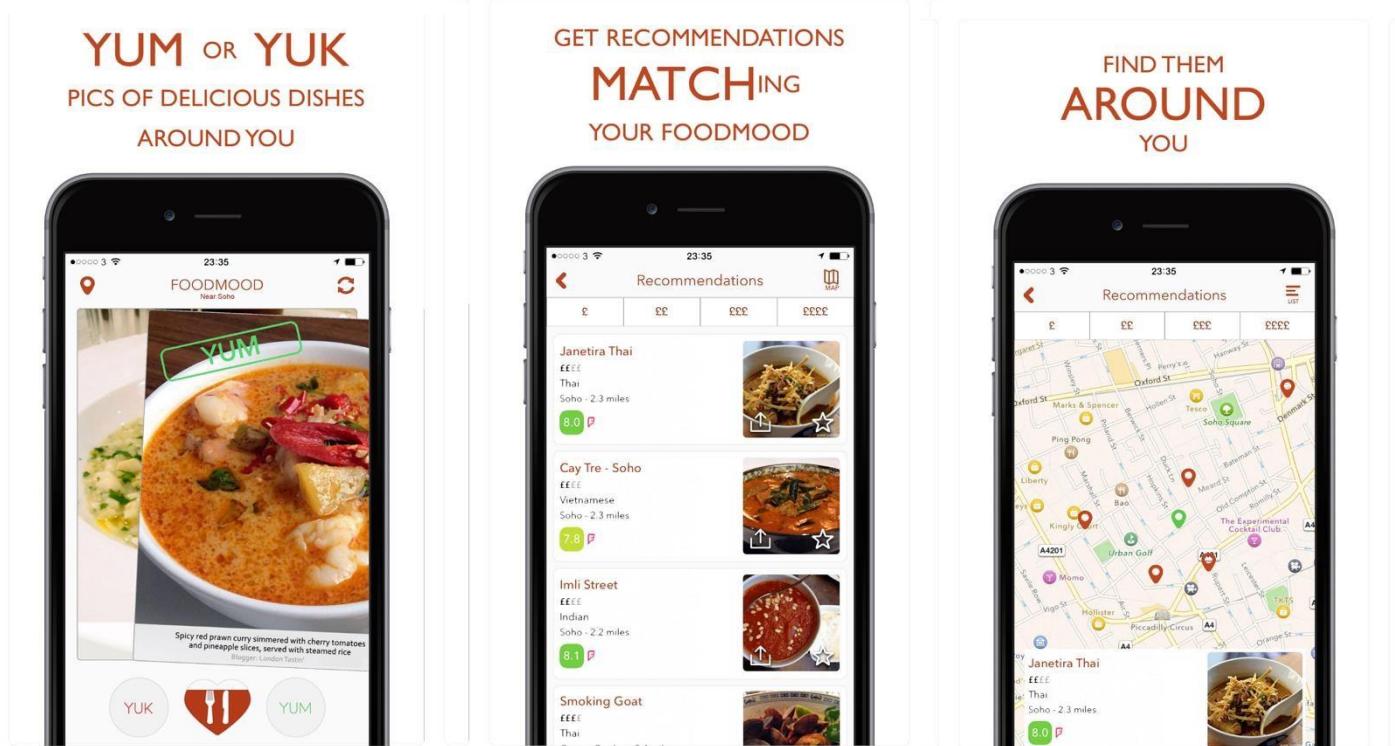


Figure2.1: Screenshot for FoodMood Application.



Foursquare

Is a local search-and-discovery service mobile app., which provides search results for its users. The app provides personalized recommendations for places to go to near a user's current location-based on users' "previous browsing history, purchases, or check-in history" [21]. It lets the user search for restaurants and other places of interest in their surrounding area. It is also possible to search other areas by entering the name of a remote location. The app displays personalized recommendations based on the time of day, displaying breakfast places in the morning, dinner places in the evening etc. Recommendations are personalized based on users "Tastes" and their venue ratings and according to their friend's reviews.

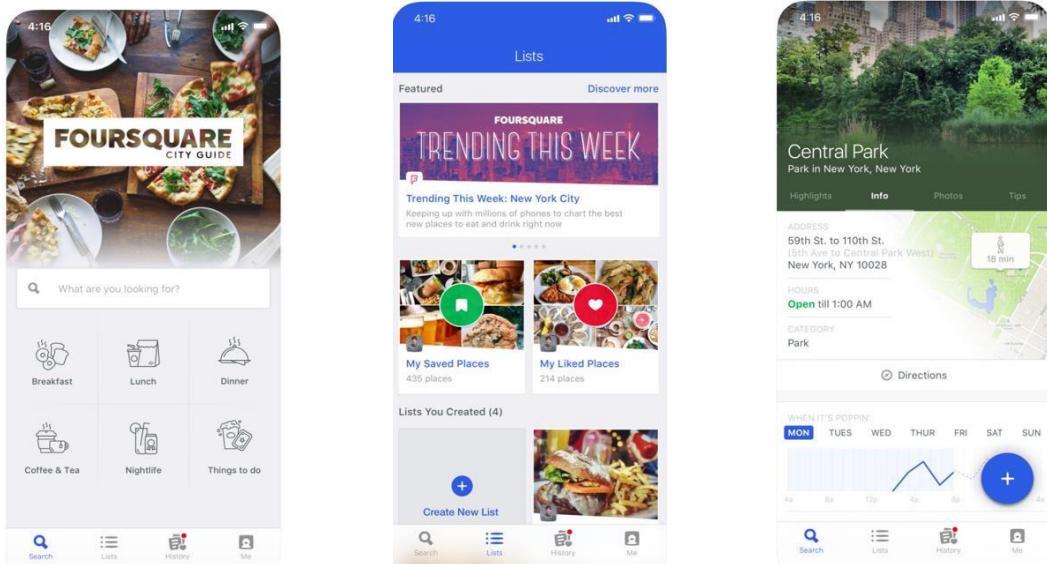


Figure2.2: Screenshot for Foursquare Application.



ParallelDots

Is a website using Emotion Detection API, it is similar to (sentiment API) it can accurately detect the emotion from any textual data. People voice out their opinion, feedback and reviews on social media, blogs and forums. Marketers and customer support can leverage the power of Emotion Detection to read and analyze emotions attached with the textual data [22].

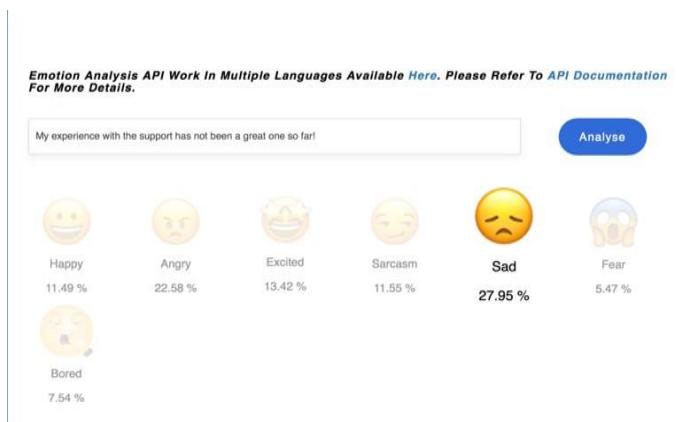


Figure2.3: Screenshot for ParallelDots Website



Tweet Sentiment Visualization

Is a website that analyze emotions from tweets through Twitter API using sentiment dictionary. it searches each tweet for words in the dictionary, then combines the words' pleasure and arousal ratings to estimate sentiment for the entire tweet. It allows the user to type a specific keyword to help analyze people's emotions toward them by showing the recent tweets that contain the keyword and pulled them from Twitter and visualize it on Sentiment tab that contains positive/negative emotions as blue/orange circles. Blue words are evaluated as-is. Orange words are evaluated as though they are negated. Example of the visual features assigned to a circle in the figure to represent its tweet's estimated sentiment: color—blue for unpleasant, green for pleasant [23].



Figure2.4: Screenshot for Tweet Sentiment Visualization Website.



FoodMedicine

Users in Food Medicine app able to select a mood or general situation you are experiencing and get a list of foods you can look out for the next time you go to the grocery store based on your mood.

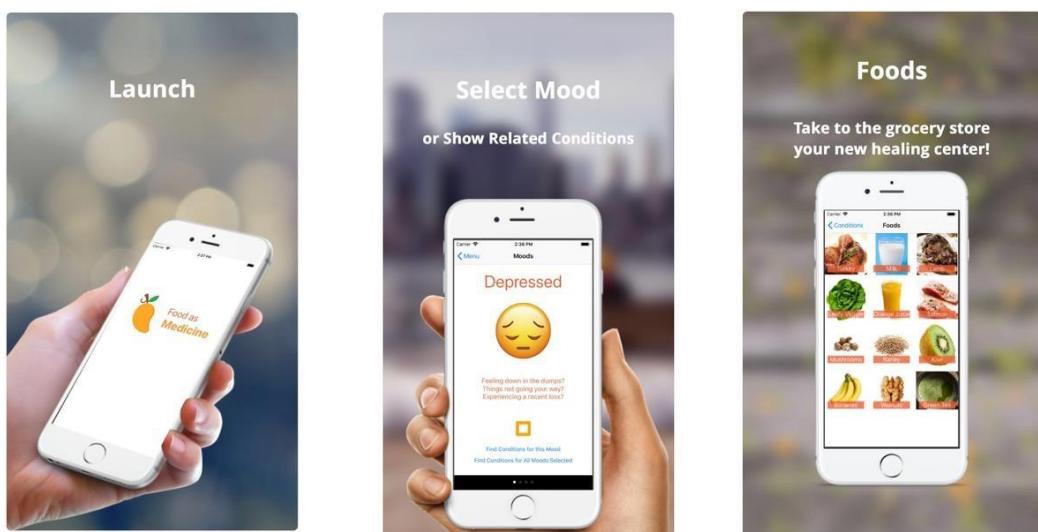


Figure2.5: Screenshot for FoodMedicine Application.



Sentimentic

Is a mobile application that uses Arabic Sentiment Analysis to calculate the percentage of positive, neutral and negative sentiments about a topic of choice, in addition to that, sentiment provide other analytical information to help understand what people think about any topic from different point of views. The analytical information includes: top used emoji, number of retweets of positive, neutral and negative tweets, recurrent words in the analyzed tweets, displaying the top positive, neutral and negative tweets [24].

Many sentiment analysis applications were developed for English, but we are trying to add a new direction and come up with an application with a high accuracy Arabic based sentiment analysis which is not affected by the use of dialects. An application that gives Arab users the ability to know the general sentiment about trending restaurants being discussed. Arabic language has many dialects that should be considered, where in each dialect meanings of words could be very different. Arabic is a morphologically rich language.

2.1.3 COMPARISON BETWEEN THE APPLICATIONS CHARACTERISTICS WITH VIBES

Table2.1: Comparison between the applications characteristics with VIBES

Real time analysis	Applicable	Not applicable	Not applicable	Applicable	Not applicable	Not applicable	Applicable
Analysis approach	Sentiment Analysis	Not applicable	Sentiment Analysis	Sentiment & Emotion Analysis	Emotion Analysis	Sentiment Analysis	Sentiment Analysis
Geographical region	U.S, London	All the world	All the world	All the world	All the world	Saudi Arabia	Riyadh, Jeddah, Dammam
Language	English	13 different languages	14 different languages	All languages	English	Arabic	Arabic
Analysis	Picture Analysis	-	Text	Text/emoji's	Emoji's	Text/emoji's	Text/emoji's
Location	Applicable	Applicable	Not applicable	Not applicable	Not applicable	Not applicable	Applicable
Emotions	Not applicable	Not applicable	Positive/Negative	Positive/Negative	Negative	Positive/Negative	Positive/Negative

2.1.4 INTERVIEWS

To obtain good and accurate results and to know the most important factors that must be seen "focused on" at the end of the project, we conducted interviews with restaurant pioneers who have interest in evaluating different restaurants on their accounts. Twitter or Instagram based on their impression and feelings about the beauty of either the place, quietness, menu or prices...etc., one of the reasons for conducting this interview is to know the features and characteristics that they want to provide in the application. The interview was conducted on a sample of 4 people, which took place on 23 Sep 2018 at 5:00 pm by Online conversation.

In the interview, 6 questions were asked. At the beginning of the interview, the first question was: "Do you prefer to know people's opinions about restaurants and taking them?" With 84% approval and 16% opposition (As Shown in Figure 6).

Followed by "Do you prefer an application that presents you the experience and opinions of visitors about the restaurant in terms of cost, best meals served or services in the restaurant?" With the approval of 98% of restaurants pioneers (As Shown in Figure 7).

Then we asked, "In your opinion if there is an application that displays restaurants that have the most customer satisfaction, will it help people choose the restaurant faster and easier?" 3 of them were totally supportive and encouraged to be present while the other was not quite sure.

The fourth question was, "What are the features you want to be available in the application?" The answers were that it should be easy to use, and whether there is space to evaluate in details and leave comments, and to provide sufficient information about the restaurant of the quality of the food and prices and service Furthermore, to provide a description of the place, and also to show the restaurants based on the user location. And the fifth question was "Would you prefer to express your opinion about a restaurant after the visit it?" The approval rate was 99% (As Shown in Figure8).

The last question was to find out whether "Will they use an application that allows them to express their opinions instead of their accounts in Twitter or Instagram?" Some of them were supportive and encouraged to exist and some of them rejecting it. as well as one of them his answer was "based on the application will be my decision if it has a beautiful interface and has all the specifications that dispense the research, certainly yes."

After the interview, we found out the desire of restaurant pioneers and evaluators in the content of the application and identify the supporting characteristics that will help facilitate the use of the application. Based on the answers, the questionnaires were built.

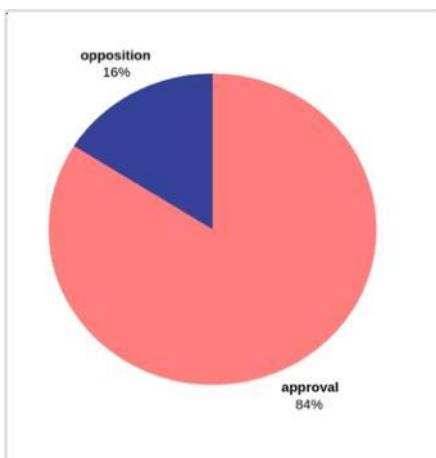


Figure2.6: Interview question1

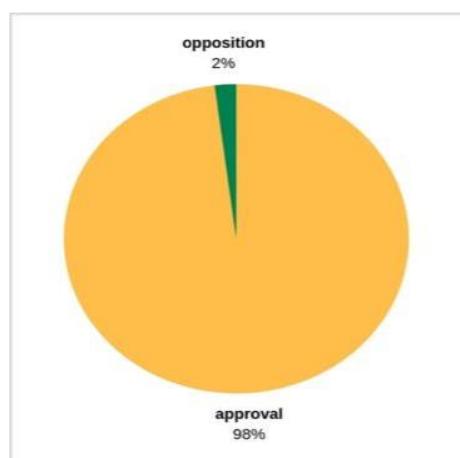


Figure2.7: Interview question2

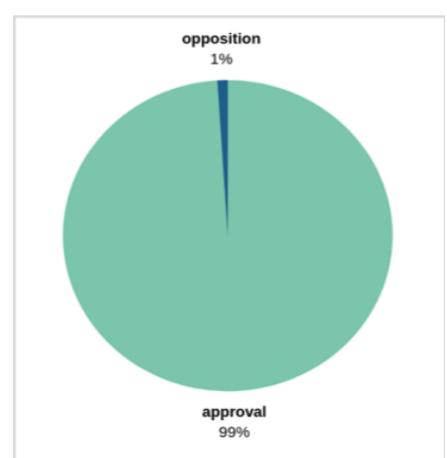


Figure2.8: Interview question3

2.1.5 QUESTIONNAIRE

In VIBES project we conduct a questionnaire that was piloted in Saudi Arabia on 1002 people from both male by 22% and female by 79% (As Shown in Figure 10). that have interest in visiting restaurants in certain range age group from 20-50 but most of them are 20-30 (As Shown in Figure 11). This questionnaire aims to help us in define user requirements. In the questionnaire we asked 4 questions.

First, there is 32.7% agreements that "They are prefer to share their impression/feelings about specific restaurant in social media" (As Shown in Figure 12).

Second, 89.1% agreements that "They prefer to have an application that shows only restaurants that have a high impression of positive emotions either with good service or beauty and quiet place" (As Shown in Figure 13).

Third question, "If there is an application that provides this service, what features do you want"; and the most of answers is "present trending restaurant in specific period", "availability of music and partition" and "ability to search about specific restaurant".

Lastly, most of them suggest that "View the tweets about each restaurant", "user can upload picture about specific restaurant "and "display pictures about restaurant". Refer to (Appendix).

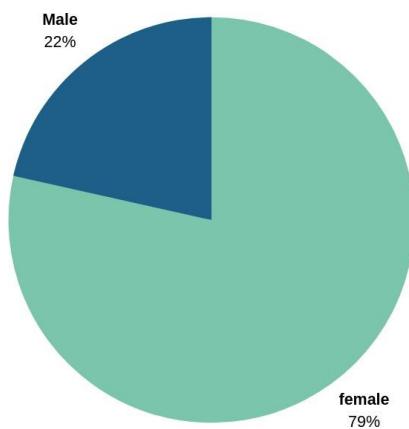


Figure2.9: Questionnaire gender

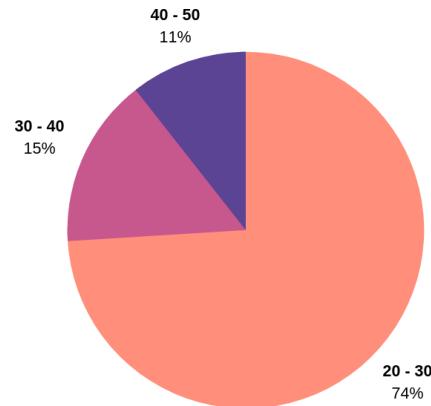


Figure2.10: Questionnaire range of age

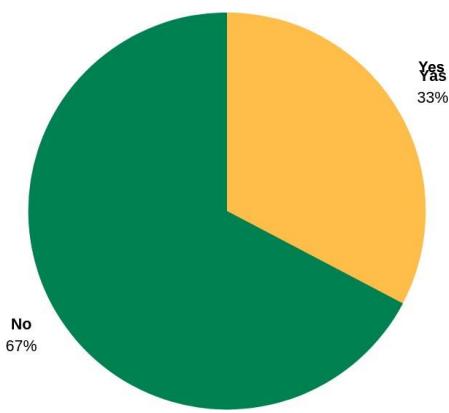


Figure2.11: Questionnaire question 1

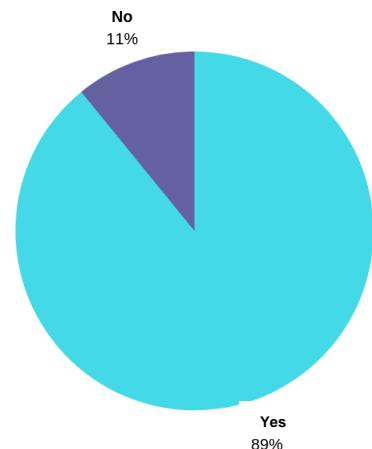


Figure2.12: Questionnaire question 2

2.2 CONCLUSION AND OUTCOMES

With the rise of social networking epoch, there has been a surge of user generated content. Blogging sites have millions of people sharing their through daily because of its characteristic short and simple manner of expression. One of them, Twitter where users post real time reactions to and opinions about everything [25]. After the completion of our literature review, we found that Twitter is the main source that can be used to extract and analyze people's emotions.

According to our research in related studies, we noticed that most of the researchers classify tweets according to sentiment analysis that detects simple attitudes (positive, negative and neutral). Moreover, they have analyzed tweets' text while neglecting the used emoji's. Emoji's are used frequently in social media. A widely assumed view is that emoji's express the emotional state of the user, which has led to research focusing on the expressiveness of emoji's independent from the linguistic context [26].

With the rapid increase in the volume of Arabic opinionated posts on different microblogging mediums, comes an increasing demand for Arabic sentiment analysis tools [27]. And as a result of the described problem, we decided to present a solution by creating a real-time application that gathers Saudi's Arabic text posts from the service Twitter and Google maps, taking into consideration their different dialects and the emojis used. Then, analyzing them in terms of the following emotions: happiness, enjoyment, beauty, sad and satisfaction. However, one of the reasons that have led us to choose iOS platform, is the bias of a large proportion of adults who are target in our project to its used and preference over Android.

We represented many applications that focus on sentiment analysis we found some differences among the applications with respect to features and the reason to create them even though they have a common goal. Then, we have provided a comparison table for the applications we have mentioned earlier with the features that our application will provide. The table will help us to clarify and visualize the application that we aim to develop and to build it in a professional way. Also, we took into consideration all the opinions and suggests that help us in developing our application, by conducting a questionnaire. The questionnaire led us to measure people's interest in knowing impressions and feelings about restaurants before visiting. And one of the reasons for conducting a questionnaire is to know the features and characteristics that they want to provide in the application. In addition, we have made some interviews with some pioneers who have an interest in evaluating different restaurants on their accounts in either Twitter based on their impression and feelings about it to measure the acceptance and get their feedback about having an application that provide the serves in Arabic language. The application will help the users to find the restaurants with high positive emotion for each city by location or by searching for a specific restaurant by provide analysis result, location, and some pictures about the places and provide the visitors' impressions and opinion. Also, allowing users to express their opinion.

Chapter 3 | SYSTEM ANALYSIS



This chapter talks about system analysis, which will help to determine how the system works. At the beginning, we presented our development methodology, which used in order to plan and control the process of the system development. Then, we stated the characteristics that the user interacts with in the system. After that, we demonstrated the system requirements. In this part, we present functional requirements that specify what the system should do. Also, we have non-functional requirements which describe what standard functions should be provided by the system. And at the end, we represented the user behavior when using the system through a set of system analysis models.

3.1 DEVELOPMENT METHODOLOGY

In VIBES project, we will use agile methodology to achieve higher flexibility and to better satisfy actual customer requirements, there is an increasing tendency to develop and deliver software in an incremental fashion. In adopting this process, requirements are delivered in releases and so a decision has to be made on which requirements should be delivered in which release [28].

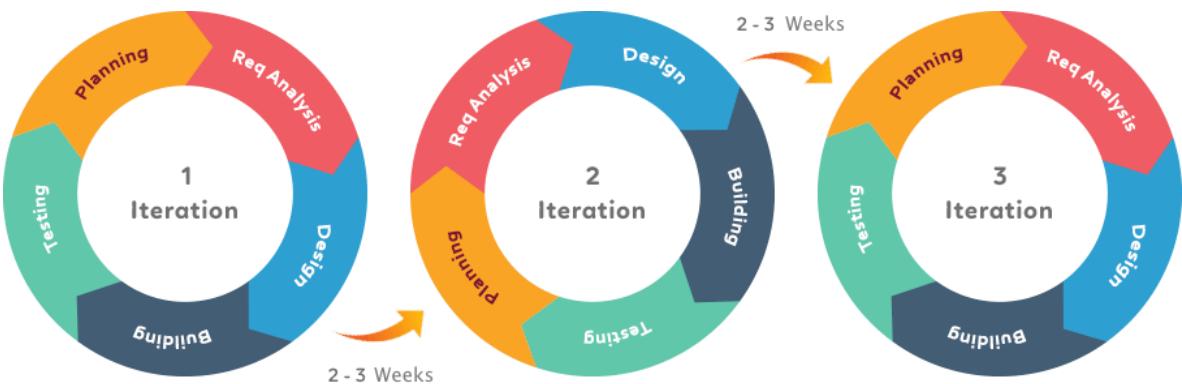


Figure3.1: Agile Methodology

3.2 USER CHARACTERISTICS

- The application will be available in these cities (Riyadh, Jeddah, Dammam), since these are the big cities in kingdom.
- Any user has the ability to search for specific restaurant without registration, its mandatory for those who need to share a picture or post a comment.
- It was designed for all users who have knowledge of Arabic language even if they do not have technical expertise.

3.3 USER REQUIREMENTS

In VIBES project, we will use IEEE technology standards to ensure that products and services perform as intended. They also foster interoperability, create uniform design, installation and testing methods, protect users and their environment and improve the quality of life of countless communities and individuals worldwide. IEEE-SA is a central source of standards in traditional and emerging technologies, particularly in telecommunications, information technology and power generation.

A lot of people debated that every stander can deliver the functional and nonfunctional requirement. Thus, IEEE provides the “SMART” good requirements concept, such as:

1. System features, in other word the functional requirement where it specifies what the system should do in formal language in order to create collaboration between the developers and the stakeholder.
2. External interface requirements, where it includes the user interfaces, hardware interfaces, communication interface.
3. It provides Software Quality assurance attributes, other nonfunctional requirement (performance, security and safety) and business rules.

So, the reason why should we use IEEE? Because it gives the opportunity to make the requirements more specific, measurable, acceptable, realistic and traceable [29].

3.3.1 USER FUNCTIONAL REQUIREMENTS

1.1 CUSTOMER REGISTRATION

1.1.1 DESCRIPTION AND PRIORITY

A customer, who wants to register him /her self in the system, enters his/her details. If the details are verified, the system will register the customer.

1.1.2 STIMULUS/RESPONSE SEQUENCE

Stimulus: Customer clicks in sign up option.

Response: System queries customer for his/her details.

Stimulus: Customer enters the required details.

Response: If status is "correct/accept" system allow customer to register.

1.1.3 FUNCTIONAL REQUIREMENTS

System shall allow a customer to register.

2.1 CUSTOMER LOG IN

2.1.1 DESCRIPTION AND PRIORITY

Customer who is already registered in the system can log in that enable he/she to benefit in system features.

2.1.2 STIMULUS/RESPONSE SEQUENCE

Stimulus: Customer clicks in log in option.

Response: System queries customer for his/her details.

Stimulus: Customer enters the required credentials.

Response: If status is "correct/accept" system allow customer to login.

2.1.3 FUNCTIONAL REQUIREMENTS

System shall allow a customer to log in.

3.1 CUSTOMER SEARCH

3.1.1 DESCRIPTION AND PRIORITY

System detects customer location after registration or before it when he/she skips the register option. After customer login, he/she can search for specific restaurant already.

3.1.2 STIMULUS/RESPONSE SEQUENCE

Stimulus: Customer click on the search option.

Response: System queries customer for the restaurant name/information.

Stimulus: Customer enters the required restaurant that want it.

Response: System view/display the specific restaurant that customer search about it based on the nearest branch.

3.1.3 FUNCTIONAL REQUIREMENTS

System shall allow customer to search for specific restaurant

4.1 RESTAURANT CATEGORIES

4.1.1 DESCRIPTION AND PRIORITY

The System detects customer location after registration or before it when he/she skips the register option. If the customer login or not, the customer can view and select categories that represent a restaurant based on the meal type.

4.1.2 STIMULUS/RESPONSE SEQUENCE

stimulus: Customer click on specific type.

Response: System view the nearest restaurants with the specific type that customer selected.

4.1.3 FUNCTIONAL REQUIREMENTS

System shall allow customer to pick any restaurant based on specific category.

5.1 EMOTIONS LIST

5.1.1 DESCRIPTION AND PRIORITY

System detects customer location after registration or before it when he/she skips the register option. After customer selects specific type of restaurant categorize, he/she can also select a restaurant based on emotions from the provided list.

5.1.2 STIMULUS/RESPONSE SEQUENCE

Stimulus: Customer select specific category.

Response: System view list of emotions that categorize under positive emotions.

Stimulus: Customer select emotion.

Response: System view list of the restaurant that categorize under the selected emotion.

Stimulus: Customer select the restaurant based on he/she emotion.

Response: System view all the branches with the nearest one.

5.1.3 FUNCTIONAL REQUIREMENTS

System shall allow customer to select any restaurant based on specific positive emotion.

6.1 CUSTOMER PROFILE

6.1.1 DESCRIPTION AND PRIORITY

Customer who wants to edit his/her profile enters his/her details. Customer should login in the system before he/she update profile. If the details are verified, system will accept process update the customer profile/information.

6.1.2 STIMULUS/RESPONSE SEQUENCE

Stimulus: Customer clicks on edit the profile option.

Response: System queries customer for his/her details.

Stimulus: Customer enters the required credentials.

Response: If status is "correct/ accept" system allow customer to register.

6.1.3 FUNCTIONAL REQUIREMENTS

System shall allow customer to edit his/her profile.

7.1 CUSTOMER FAVORITE LIST

7.1.1 DESCRIPTION AND PRIORITY

Customer should login in the system before he/she can create favorite list that he/she will use it to save (will be contain) restaurants that he/she would like to visit.

7.1.2 STIMULUS/RESPONSE SEQUENCE

Stimulus: Customer can create list by select restaurants that he/she likes.

Response: System saves these restaurants that customer select in his/her list.

7.1.3 FUNCTIONAL REQUIREMENTS

System shall allow Customer to create favorite list of restaurants that he/she would like to visit.

8.1 ADD COMMENTS

8.1.1 DESCRIPTION AND PRIORITY

Customer should be login in the system before he/she can post comments on the system/application to express their impression or feeling about specific restaurant after he/she visit.

8.1.2 STIMULUS/RESPONSE SEQUENCE

Stimulus: Customer clicks on the add comments option.

Response: System queries customer for he/she impression/opinion or feeling

stimulus: Customer write comment about specific restaurant.

Response: System view customer comment.

8.1.3 FUNCTIONAL REQUIREMENTS

System shall allow customer to post comments about specific restaurant.

9.1 UPLOAD PICTURES

9.1.1 DESCRIPTION AND PRIORITY

Customer should be login in the system before he/she can upload some pictures on the system/application that take it for specific restaurant.

9.1.2 STIMULUS/RESPONSE SEQUENCE

Stimulus: Customer clicks on the upload pictures option.

Response: System queries customer for he/she pictures.

stimulus: Customer upload pictures for specific restaurant.

Response: System view pictures

9.1.3 FUNCTIONAL REQUIREMENTS

System shall allow customer to upload pictures for specific restaurant.

10.1 RESTAURANT'S INFORMATION

10.1.1 DESCRIPTION AND PRIORITY

When customer select restaurant, the system will display necessary information.

10.1.2 STIMULUS/RESPONSE SEQUENCE

Stimulus: Customer clicks on specific restaurant.

Response: System display necessary information to the user.

10.1.3 FUNCTIONAL REQUIREMENTS

System shall allow customer to view restaurant information.

11.1 DELETE COMMENT

11.1.1 DESCRIPTION AND PRIORITY

Customer should be login in the system before he/she can delete comment.

11.1.2 STIMULUS/RESPONSE SEQUENCE

Stimulus: Customer click delete comment button.

Response: System delete customer comment.

11.1.3 FUNCTIONAL REQUIREMENTS

System shall allow customer to delete comment.

12.1 DELETE PICTURE

12.1.1 DESCRIPTION AND PRIORITY

Customer should be login in the system before he/she can delete picture.

12.1.2 STIMULUS/RESPONSE SEQUENCE

Stimulus: Customer click delete picture button.

Response: System delete customer picture.

12.1.3 FUNCTIONAL REQUIREMENTS

System shall allow customer to delete picture.

13.1 REMOVE RESTAURANT FROM LIST

13.1.1 DESCRIPTION AND PRIORITY

Customer should be login in to the system before he/she can delete a restaurant from his/her list.

13.1.2 STIMULUS/RESPONSE SEQUENCE

Stimulus: Customer click delete restaurant from his/her list button.

Response: System delete selected restaurant from the list.

13.1.3 FUNCTIONAL REQUIREMENTS

System shall allow customer to delete specific restaurant from the list.

3.4 SYSTEM REQUIREMENTS

3.4.1 SYSTEM FUNCTIONAL REQUIREMENTS

FR-1 TWEETS AND COMMENTS FLOW:

- [FR-1.1] The system shall be able to connect to Twitter.
- [FR-1.2] The system shall be able to connect to Google maps.

FR-2 ANALYZING:

- [FR-2.1] The system shall be able to analyze tweet and comment at specific time.
- [FR-2.2] The system shall be able to sort and analyze tweets by searching for specific word.
- [FR-2.3] The system shall be able to use sentiment analysis by categories positives and negatives tweets and comments.
- [FR-2.4] The system shall be able to perform text and emoji's analysis.

3.5 NON-FUNCTIONAL REQUIREMENTS

[NFR-1] AVAILABILITY

- [NFR-1.1] The system should be available 99.5%.

[NFR-2] PERFORMANCE

- [NFR-2.1] The system shall be able to receive a huge number of users per second

[NFR-3] PRIVACY

- [NFR-3.1] The system shall be able to takes into consideration the privacy of the users.

[NFR-4] USABILITY

- [NFR-4.1] The system should simple and easy to use by support Arabic and English language.
- [NFR-4.2] The system should have friendly interface by provide graphs pictures.

[NFR-5] RESPONSIVENESS

- [NFR-5.1] The system shall provide short response time.

[NFR-6] FAULT TOLERANCE

- [NFR-6.1] The system must be able to recover data after sudden network failures.

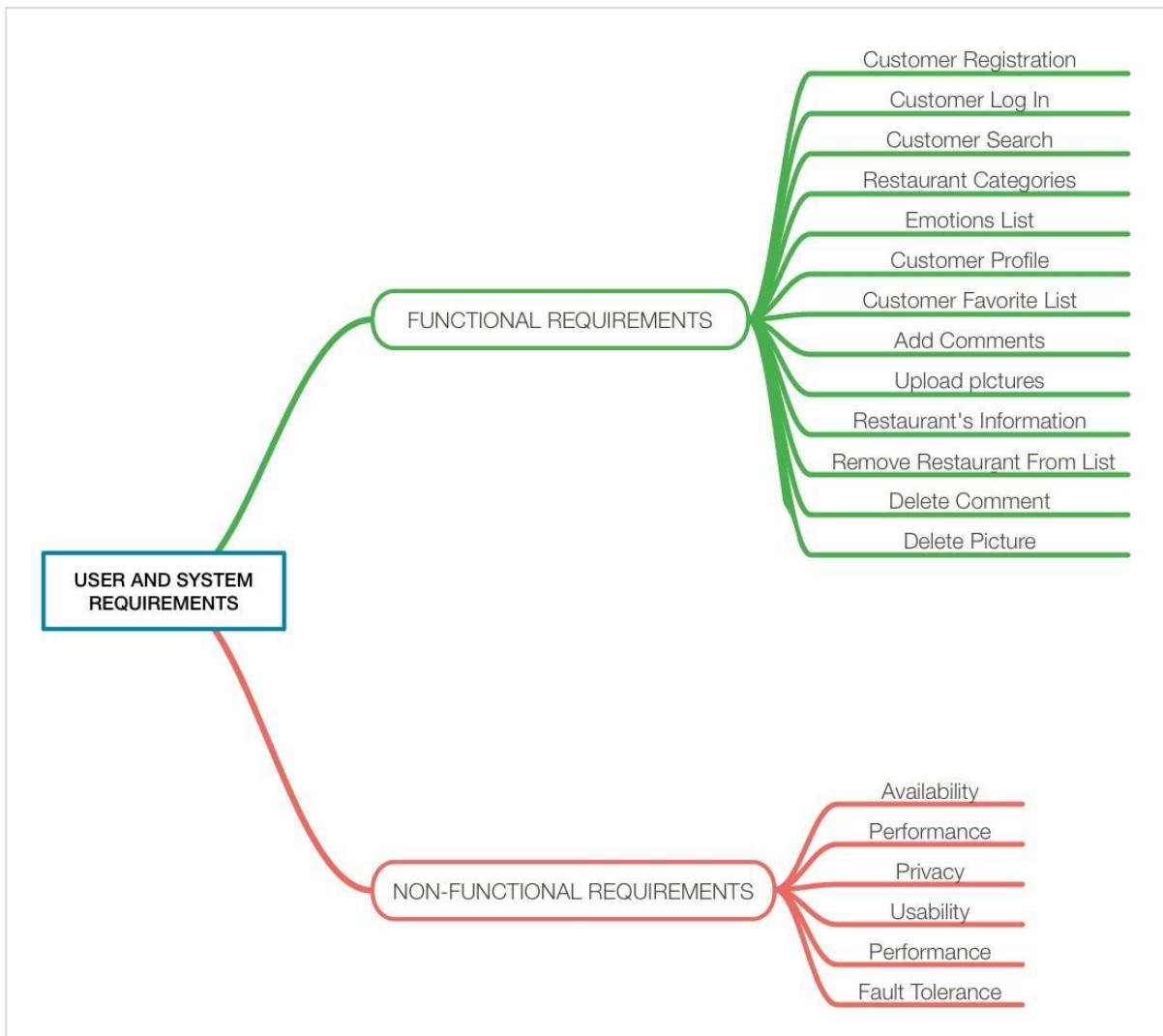


Figure3.2: A simple illustrate for functional and nonfunctional requirements

3.6 SYSTEM ANALYSIS MODELS

This section contains some system analysis models. To start with, use case diagram that is used to visualizes the functional requirements by defining the actors of the system and the functions that are performed by the actor. Second, use case description, detailed information about each use case. Third, a sequential diagram that shows the behavior of the system as messages between an actor and the system. Finally, a design class diagram that provides a graphical view of the system structure and identifying the relationships between the system components, main entities and their attributes.

3.6.1 USE CASE DIAGRAM

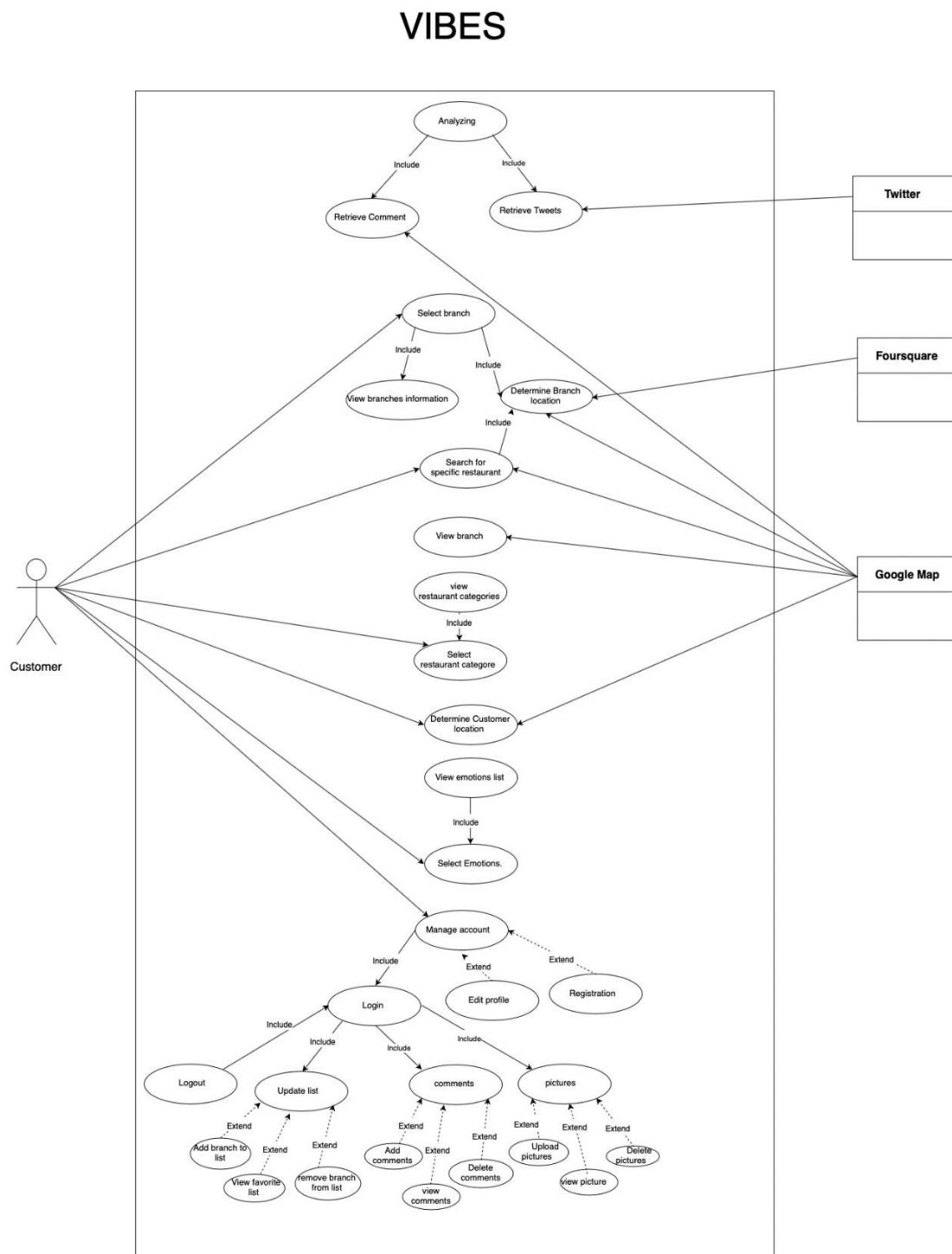


Figure3.3: Use case diagram

3.4.2 DETAILED USE CASE DESCRIPTIONS

Table3.1: View branches information

Use case name:	View branches information.
Scenario:	System view the branches information.
Triggering event:	None.
Brief description:	System will view the branch information for each restaurant from database, Google maps or foursquare API.
Actors:	None.
Related use cases:	Select branch.
Stakeholders:	Customer.
Pre-conditions:	The customer selects specific branch.
Post-conditions:	System display branch information.
Flow of activities:	System
	1. System view restaurant information in each restaurant screen.
Exception conditions:	None.

Table3.2: View restaurant categorize

Use case name:	View restaurant categories.
Scenario:	System will view restaurant categories to the customer.
Triggering event:	None.
Brief description:	System will view restaurant categories that classify restaurants based on the meal type which are breakfast, lunch, dinner or cafes to the customer.
Actors:	None.
Related use cases:	None.
Stakeholders:	Customer.
Pre-conditions:	Customer access to the application.
Post-conditions:	System display restaurant categories.
Flow of activities:	<p style="text-align: center;">System</p> <p>1. System view categories screen for the customer.</p>
Exception conditions:	None.

Table3.3: View emotions list

Use case name:	View emotions list.
Scenario:	System will view a list of positive emotions.
Triggering event:	None.
Brief description:	System will view a list of positive emotions that describe a branch vibe.
Actors:	None.
Related use cases:	Select restaurant category.
Stakeholders:	Customer.
Pre-conditions:	The customer should select a specific category.
Post-conditions:	None.
Flow of activities:	System
	1. System view a list of positive emotions (e.g. Happy, Joy, Interesting...) to the customer.
Exception conditions:	None.

Table3.4: Analyzing

Use case name:	Analyzing.
Scenario:	Performing text and emoji analyzing on Arabic tweets and comments.
Triggering event:	None.
Brief description:	Performing text and emoji analyze on all Arabic tweets and comments that have been fetched from Twitter and Google maps API about specific restaurant/branch. and classify them based on positive or negative.
Actors:	None.
Related use cases:	Retrieve Tweets, Retrieve comments.
Stakeholders:	Twitter API, Google maps API.
Pre-conditions:	System should be connected to Twitter and Google maps API and retrieve Arabic tweets and comments.
Post-conditions:	None.
Flow of activities:	<p style="text-align: center;">System</p> <p>1. System start analyzing tweets and comments.</p>
Exception conditions:	None.

Table3.5: Retrieve comments

Use case name:	Retrieve comments.	
Scenario:	Google maps API will retrieve comments.	
Triggering event:	None.	
Brief description:	Google maps API will retrieve Arabic comments to start analyzing them to positive or negative.	
Actors:	Google maps API.	
Related use cases:	None.	
Stakeholders:	None.	
Pre-conditions:	None.	
Post-conditions:	System start analyzing comments.	
Flow of activities:	Actor	System
	1.1 Google maps API accept connection. 2.1 Google maps API will retrieve comments.	1. System establish a connection with Google maps API. 2. System ask Google maps API to retrieve comments.
Exception conditions:	1. Connection failures.	

Table3.6: Retrieve Tweets

Use case name:	Retrieve Tweets.							
Scenario:	Twitter API will retrieve Tweets.							
Triggering event:	None.							
Brief description:	Twitter API will retrieve Arabic tweets to start analyzing them to positive or negative.							
Actors:	Twitter API.							
Related use cases:	None.							
Stakeholders:	None.							
Pre-conditions:	None.							
Post-conditions:	System start analyzing tweets.							
Flow of activities:	<table> <thead> <tr> <th style="text-align: center;">Actor</th> <th style="text-align: center;">System</th> </tr> </thead> <tbody> <tr> <td>1.1 Twitter API accept connection.</td> <td>1. System establish a connection with Twitter API.</td> </tr> <tr> <td>2.1 Twitter API will retrieve tweets.</td> <td>2. System ask Twitter API to retrieve tweets.</td> </tr> </tbody> </table>	Actor	System	1.1 Twitter API accept connection.	1. System establish a connection with Twitter API.	2.1 Twitter API will retrieve tweets.	2. System ask Twitter API to retrieve tweets.	
Actor	System							
1.1 Twitter API accept connection.	1. System establish a connection with Twitter API.							
2.1 Twitter API will retrieve tweets.	2. System ask Twitter API to retrieve tweets.							
Exception conditions:	1. Connection failures.							

Table3.7: Select branch

Use case name:	Select branch.					
Scenario:	Customer can select nearest branch.					
Triggering event:	None.					
Brief description:	Customer can select the nearest branch based on location, that is detected by the system after registration or before it when skip the register option.					
Actors:	Customer.					
Related use cases:	View branch.					
Stakeholders:	None.					
Pre-conditions:	System display branches.					
Post-conditions:	System display selected branch information.					
Flow of activities:	<table border="1"> <thead> <tr> <th>Actor</th> <th>System</th> </tr> </thead> <tbody> <tr> <td>2. Customer select nearest branch.</td> <td> 1. System views all the branches, which have been arranged, based on the nearness to the customer location. 2.1 System displays branch location and other information to the customer. </td> </tr> </tbody> </table>	Actor	System	2. Customer select nearest branch.	1. System views all the branches, which have been arranged, based on the nearness to the customer location. 2.1 System displays branch location and other information to the customer.	
Actor	System					
2. Customer select nearest branch.	1. System views all the branches, which have been arranged, based on the nearness to the customer location. 2.1 System displays branch location and other information to the customer.					
Exception conditions:	None.					

Table3.8: Determine branch location

Use case name:	Determine branch location.	
Scenario:	Determine restaurant location by using Google maps.	
Triggering event:	None.	
Brief description:	The system will connect to Google maps to determine branch locations. This helps the system to arrange them based on the nearest one to the customer.	
Actors:	Google maps API.	
Related use cases:	None.	
Stakeholders:	Customer.	
Pre-conditions:	System should be connected to Google maps.	
Post-conditions:	System access the branches locations.	
Flow of activities:	Actor	System
	1.1 Accept connection 2.1 Allow the system to access the branch locations.	1. System establishes a connection with Google maps API. 2. System asks Google maps API to detect the locations of the branches. 3. System access branch location.
Exception conditions:	1. Connection failures.	

Table3.9: Search for specific restaurant

Use case name:	Search for specific restaurant.	
Scenario:	Customer search for specific restaurant by its name.	
Triggering event:	None.	
Brief description:	Customer can search for specific restaurant by entering its name in the search engine to view all branches and arrange it based on the nearness.	
Actors:	Customer.	
Related use cases:	None.	
Stakeholders:	None.	
Pre-conditions:	None.	
Post-conditions:	System display all restaurant branches.	
Flow of activities:	Actor	System
	1. Customer search for specific restaurant by enter its name.	1.1 System view all restaurant branches based on the nearest.
Exception conditions:	1.1 Restaurant Not found.	

Table3.10: View branches

Use case name:	View branches.
Scenario:	System view branches based on the nearest.
Triggering event:	None.
Brief description:	System will view the Restaurant's branches that will be arranged based one the nearest one to the customer location.
Actors:	None.
Related use cases:	Select restaurant categories, search for specific restaurant, Select Emotions.
Stakeholders:	Customer.
Pre-conditions:	The customer should select specific category, emotion or by search about a specific restaurant.
Post-conditions:	System arrange the branches based on the nearest to customer location.
Flow of activities:	<p style="text-align: center;">System</p> <p>1. System view restaurant branches and arrange it based on nearness.</p>
Exception conditions:	None.

Table3.11: Select restaurant categories

Use case name:	Select restaurant categories.	
Scenario:	Customer can select restaurant meals type from categories.	
Triggering event:	None.	
Brief description:	Customer can select from the displayed categories that classify restaurants based on the meal type.	
Actors:	Customer.	
Related use cases:	View restaurant categories.	
Stakeholders:	None.	
Pre-conditions:	System display restaurants categories to customer.	
Post-conditions:	System display all the restaurant under the selected category.	
Flow of activities:	Actor	System
	1. Customer select specific category from.	1.1 System view all the restaurants under the selected category type.
Exception conditions:	None.	

Table3.12: Determine customer location

Use case name:	Determine customer location.	
Scenario:	Determine customer location by using Google maps.	
Triggering event:	None.	
Brief description:	The system will connect to Google maps to determine customer locations. That helps the system to arrange restaurant branches based on the nearest one to the customer. If the customer rejects system to detect location, in this case customer will not be allowed to access and use the application.	
Actors:	Google maps API.	
Related use cases:	None.	
Stakeholders:	Customer.	
Pre-conditions:	The customer should be able to access to the application.	
Post-conditions:	System arrange restaurant branches based on the nearness to customer location.	
Flow of activities:	Actor	System
	1.1 Accept connection 2.1 Allow the system to access customer location.	1. System establishes a connection with Google maps. 2. System asks Google maps to detect the customer location. 3. System accesses customer location.
Exception conditions:	1. Connection failures.	

Table3.13: Select emotions

Use case name:	Select emotions.					
Scenario:	Customer select specific positive emotion.					
Triggering event:	None.					
Brief description:	The customer can select specific positive emotion from the displayed list that describes restaurants VIBES.					
Actors:	Customer.					
Related use cases:	View list of emotions.					
Stakeholders:	None.					
Pre-conditions:	System display list of positive emotions.					
Post-conditions:	System display restaurant represent selected emotion.					
Flow of activities:	<table border="1"> <thead> <tr> <th>Actor</th> <th>System</th> </tr> </thead> <tbody> <tr> <td>1. Customer select specific emotion.</td> <td>1.1 System presents the restaurant under the specified emotion based on the nearest one.</td> </tr> </tbody> </table>	Actor	System	1. Customer select specific emotion.	1.1 System presents the restaurant under the specified emotion based on the nearest one.	
Actor	System					
1. Customer select specific emotion.	1.1 System presents the restaurant under the specified emotion based on the nearest one.					
Exception conditions:	None.					

Table3.14: Login

Use case name:	Login.							
Scenario:	Customer can login after registration.							
Triggering event:	None.							
Brief description:	Customer can login after register to be allowed to use some features.							
Actors:	Customer.							
Related use cases:	Registration.							
Stakeholders:	None.							
Pre-conditions:	The customer should be registered in the application.							
Post-conditions:	None.							
Flow of activities:	<table border="1"> <thead> <tr> <th>Actor</th> <th>System</th> </tr> </thead> <tbody> <tr> <td>1. Customer click login option.</td> <td>1.1 System queries customer to enter information.</td> </tr> <tr> <td>2. Customer fill required fields.</td> <td>2.1 System validation entries of the customer. 2.2 System display categories screen.</td> </tr> </tbody> </table>	Actor	System	1. Customer click login option.	1.1 System queries customer to enter information.	2. Customer fill required fields.	2.1 System validation entries of the customer. 2.2 System display categories screen.	
Actor	System							
1. Customer click login option.	1.1 System queries customer to enter information.							
2. Customer fill required fields.	2.1 System validation entries of the customer. 2.2 System display categories screen.							
Exception conditions:	2.1 Invalid entries.							

Table3.15: Edit profile

Use case name:	Edit profile.							
Scenario:	Customers edit their profile.							
Triggering event:	None.							
Brief description:	Customers can edit their profile information by changing some of their details such as, name, email and password.							
Actors:	Customer.							
Related use cases:	Login.							
Stakeholders:	Customer.							
Pre-conditions:	Customers should be login in the application to be allowed to edit profile.							
Post-conditions:	System edit profile information.							
Flow of activities:	<table border="1"> <thead> <tr> <th>Actor</th> <th>System</th> </tr> </thead> <tbody> <tr> <td>1. Customer click the edit profile option.</td> <td>1.1 System view edit profile screen.</td> </tr> <tr> <td>2. Customer edit the required field.</td> <td>2.1 System validation entries of the customer. 2.2 System save the edited fields.</td> </tr> </tbody> </table>	Actor	System	1. Customer click the edit profile option.	1.1 System view edit profile screen.	2. Customer edit the required field.	2.1 System validation entries of the customer. 2.2 System save the edited fields.	
Actor	System							
1. Customer click the edit profile option.	1.1 System view edit profile screen.							
2. Customer edit the required field.	2.1 System validation entries of the customer. 2.2 System save the edited fields.							
Exception conditions:	2.1 Invalid entries. 2.2 saving failure.							

Table3.16: Registration

Use case name:	Registration.							
Scenario:	Customer can create an account in the application.							
Triggering event:	None.							
Brief description:	Customer can create an account in the application to use specific Application features.							
Actors:	Customer.							
Related use cases:	None.							
Stakeholders:	Customer.							
Pre-conditions:	When customer want to add comments, upload pictures.							
Post-conditions:	Log in.							
Flow of activities:	<table border="1"> <thead> <tr> <th style="text-align: center;">Actor</th> <th style="text-align: center;">System</th> </tr> </thead> <tbody> <tr> <td>1. Customer clicks in sign up option.</td> <td>1.1 System queries customer to enter information.</td> </tr> <tr> <td>2. Customer fill required fields</td> <td>2.1 System validation entries of the customer. 2.2 System save the customer information.</td> </tr> </tbody> </table>	Actor	System	1. Customer clicks in sign up option.	1.1 System queries customer to enter information.	2. Customer fill required fields	2.1 System validation entries of the customer. 2.2 System save the customer information.	
Actor	System							
1. Customer clicks in sign up option.	1.1 System queries customer to enter information.							
2. Customer fill required fields	2.1 System validation entries of the customer. 2.2 System save the customer information.							
Exception conditions:	2.1 Invalid entries.							

Table3.17: Add branch to list

Use case name:	Add branch to list.							
Scenario:	Customers can add restaurants to favorite list.							
Triggering event:	None.							
Brief description:	Customers can add restaurant to favorite list that already exists after customer registers to save the restaurant that he/she would like to visit.							
Actors:	Customer.							
Related use cases:	Login, Select Restaurant.							
Stakeholders:	None.							
Pre-conditions:	The customer should be login in the application and select a specific restaurant in order to add it to favorite list.							
Post-conditions:	System add selected restaurant to favorite list.							
Flow of activities:	<table border="1"> <thead> <tr> <th style="text-align: center;">Actor</th> <th style="text-align: center;">System</th> </tr> </thead> <tbody> <tr> <td>1. Customer clicks on list option.</td> <td>1.1 System displays the list to customer. 1.2 System allows customer to update on list.</td> </tr> <tr> <td>2. Customer adds restaurants to list.</td> <td>2.1 system saves added restaurants in the list.</td> </tr> </tbody> </table>	Actor	System	1. Customer clicks on list option.	1.1 System displays the list to customer. 1.2 System allows customer to update on list.	2. Customer adds restaurants to list.	2.1 system saves added restaurants in the list.	
Actor	System							
1. Customer clicks on list option.	1.1 System displays the list to customer. 1.2 System allows customer to update on list.							
2. Customer adds restaurants to list.	2.1 system saves added restaurants in the list.							
Exception conditions:	2.2 saving failure.							

Table3.18: View favorite list

Use case name:	View favorite list.
Scenario:	System view the favorite list.
Triggering event:	None.
Brief description:	System views the favorite list that exists for each registered customer.
Actors:	None.
Related use cases:	Login, Remove restaurant from list.
Stakeholders:	Customer.
Pre-conditions:	The customer should be login in the application.
Post-conditions:	None.
Flow of activities:	<p style="text-align: center;">System</p> <p>1. System view the favorite list.</p>
Exception conditions:	None.

Table3.19: Remove branch from list

Use case name:	Remove branch from list.	
Scenario:	Customers can remove restaurant from favorite list.	
Triggering event:	None.	
Brief description:	Customers can remove a specific restaurant from favorite list.	
Actors:	Customer.	
Related use cases:	Log in, Add restaurant to list.	
Stakeholders:	Customer.	
Pre-conditions:	The customer should be login in the application and already has been added restaurants to the favorite list.	
Post-conditions:	System remove selected restaurant from favorite list.	
Flow of activities:	Actor	System
	1. Customer click on list option. 2. Customer remove specific restaurants from the list.	1.1 System display the list to customer. 1.2 System allow customer to update on list. 2.1 system save the change.
Exception conditions:	2.1 saving failure.	

Table3.20: Add comments

Use case name:	Add comments.							
Scenario:	Customer can add comments for specific restaurants.							
Triggering event:	None.							
Brief description:	Customers can add comments for specific restaurants to express their impressions/opinions that help other customers to get more honest feedback.							
Actors:	Customer.							
Related use cases:	Log in, Select Restaurant.							
Stakeholders:	Customer.							
Pre-conditions:	The customer should be login in the application and select specific restaurant to add comments.							
Post-conditions:	System add comment.							
Flow of activities:	<table border="1"> <thead> <tr> <th style="text-align: center;">Actor</th> <th style="text-align: center;">System</th> </tr> </thead> <tbody> <tr> <td>1. Customer click on add comments option.</td> <td>1.1 System display all comments that has been posted for selected restaurant. 1.2 System allow customer to add new comment.</td> </tr> <tr> <td>2. Customer add comment.</td> <td>2.1 System save added comment. 2.2 System display comment to other customers.</td> </tr> </tbody> </table>	Actor	System	1. Customer click on add comments option.	1.1 System display all comments that has been posted for selected restaurant. 1.2 System allow customer to add new comment.	2. Customer add comment.	2.1 System save added comment. 2.2 System display comment to other customers.	
Actor	System							
1. Customer click on add comments option.	1.1 System display all comments that has been posted for selected restaurant. 1.2 System allow customer to add new comment.							
2. Customer add comment.	2.1 System save added comment. 2.2 System display comment to other customers.							
Exception conditions:	2.1 Saving failure.							

Table3.21: View comments

Use case name:	View comments.
Scenario:	System view the comments.
Triggering event:	None.
Brief description:	System view the comments that added by customers in each restaurant.
Actors:	None.
Related use cases:	Add comments, views branch information.
Stakeholders:	Customer.
Pre-conditions:	Customers add comments.
Post-conditions:	None.
Flow of activities:	<p style="text-align: center;">System</p> <p>1. System view the comments to specific restaurant.</p>
Exception conditions:	None.

Table3.22: Delete comments

Use case name:	Delete comments.							
Scenario:	Customer can delete comments about specific restaurants.							
Triggering event:	None.							
Brief description:	Customer can delete comments that already post it for specific restaurant.							
Actors:	Customer.							
Related use cases:	Log in, Add comment							
Stakeholders:	Customer.							
Pre-conditions:	The customer should be login in the application and already has been added comment for specific restaurant.							
Post-conditions:	System delete comment.							
Flow of activities:	<table border="1"> <thead> <tr> <th>Actor</th> <th>System</th> </tr> </thead> <tbody> <tr> <td>1. Customer selects comment that has been added before for specific restaurant.</td> <td>1.1 System views selected comment to the customer.</td> </tr> <tr> <td>2. Customer clicks on delete comment option.</td> <td>2.1 System saves the change.</td> </tr> </tbody> </table>	Actor	System	1. Customer selects comment that has been added before for specific restaurant.	1.1 System views selected comment to the customer.	2. Customer clicks on delete comment option.	2.1 System saves the change.	
Actor	System							
1. Customer selects comment that has been added before for specific restaurant.	1.1 System views selected comment to the customer.							
2. Customer clicks on delete comment option.	2.1 System saves the change.							
Exception conditions:	2.1 Saving failure.							

Table3.23: Upload pictures

Use case name:	Upload pictures.							
Scenario:	Customer can upload pictures.							
Triggering event:	None.							
Brief description:	Customers can upload pictures for specific restaurant to share moments with other customers.							
Actors:	Customer.							
Related use cases:	Login.							
Stakeholders:	None.							
Pre-conditions:	The customer should be login in the application and select specific restaurant to upload pictures.							
Post-conditions:	System upload picture.							
Flow of activities:	<table border="1"> <thead> <tr> <th style="text-align: center;">Actor</th> <th style="text-align: center;">System</th> </tr> </thead> <tbody> <tr> <td>1. Customer clicks on upload pictures option.</td> <td>1.1 System views all preloaded pictures for selected restaurant. 2.2 System allow customer to upload new pictures.</td> </tr> <tr> <td>2. Customer uploads picture.</td> <td>2.1 System saves uploaded picture. 2.2 System displays picture to other customers.</td> </tr> </tbody> </table>	Actor	System	1. Customer clicks on upload pictures option.	1.1 System views all preloaded pictures for selected restaurant. 2.2 System allow customer to upload new pictures.	2. Customer uploads picture.	2.1 System saves uploaded picture. 2.2 System displays picture to other customers.	
Actor	System							
1. Customer clicks on upload pictures option.	1.1 System views all preloaded pictures for selected restaurant. 2.2 System allow customer to upload new pictures.							
2. Customer uploads picture.	2.1 System saves uploaded picture. 2.2 System displays picture to other customers.							
Exception conditions:	2. Picture size inappropriate. 2.1 Saving failure.							

Table3.24: View pictures

Use case name:	View pictures.
Scenario:	System view the picture.
Triggering event:	None.
Brief description:	System view the picture that has been added by customers in each restaurant.
Actors:	None.
Related use cases:	Add picture, view branch information.
Stakeholders:	Customer.
Pre-conditions:	None.
Post-conditions:	System displays picture to selected restaurant.
Flow of activities:	<p style="text-align: center;">System</p> <p>1. System views the pictures of specific restaurant.</p>
Exception conditions:	None.

Table3.25: Delete pictures

Use case name:	Delete pictures.							
Scenario:	Customer can delete pictures.							
Triggering event:	None.							
Brief description:	The customer can delete pictures that are already uploaded for a specific restaurant.							
Actors:	Customer.							
Related use cases:	Login, Upload pictures.							
Stakeholders:	Customer.							
Pre-conditions:	The customer should be login in the application and already has uploaded pictures for specific restaurant.							
Post-conditions:	Picture deleted.							
Flow of activities:	<table border="1"> <thead> <tr> <th style="text-align: center;">Actor</th> <th style="text-align: center;">System</th> </tr> </thead> <tbody> <tr> <td>1. Customer selects picture that has been uploaded before for specific restaurant.</td> <td>1.1 System displays selected pictures to the customer.</td> </tr> <tr> <td>2. Customer clicks on delete picture option.</td> <td>2.1 System saves the change.</td> </tr> </tbody> </table>	Actor	System	1. Customer selects picture that has been uploaded before for specific restaurant.	1.1 System displays selected pictures to the customer.	2. Customer clicks on delete picture option.	2.1 System saves the change.	
Actor	System							
1. Customer selects picture that has been uploaded before for specific restaurant.	1.1 System displays selected pictures to the customer.							
2. Customer clicks on delete picture option.	2.1 System saves the change.							
Exception conditions:	2.1 Saving failure.							

Table3.26: Logout

Use case name:	Logout.					
Scenario:	Customer can logout from the account.					
Triggering event:	None.					
Brief description:	Customer can logout from the account.					
Actors:	Customer.					
Related use cases:	Log in.					
Stakeholders:	None.					
Pre-conditions:	The customer should be registered in the application.					
Post-conditions:	None.					
Flow of activities:	<table border="1"> <thead> <tr> <th>Actor</th> <th>System</th> </tr> </thead> <tbody> <tr> <td>1. Customer click logout option.</td> <td>1.1 System will terminate customer access.</td> </tr> </tbody> </table>	Actor	System	1. Customer click logout option.	1.1 System will terminate customer access.	
Actor	System					
1. Customer click logout option.	1.1 System will terminate customer access.					
Exception conditions:	None.					

3.6.3 SEQUENTIAL DIAGRAMS

View restaurant's categories:

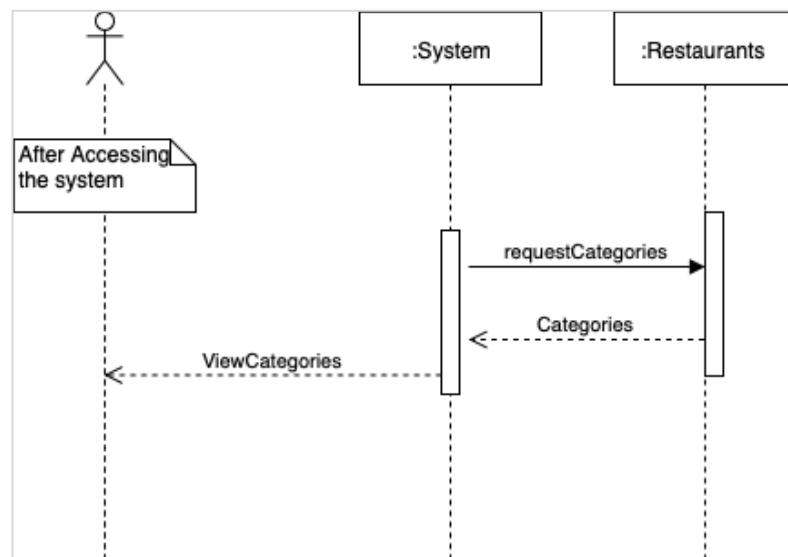


Figure3.4: View restaurant's categories Sequential Diagrams

View emotions list:

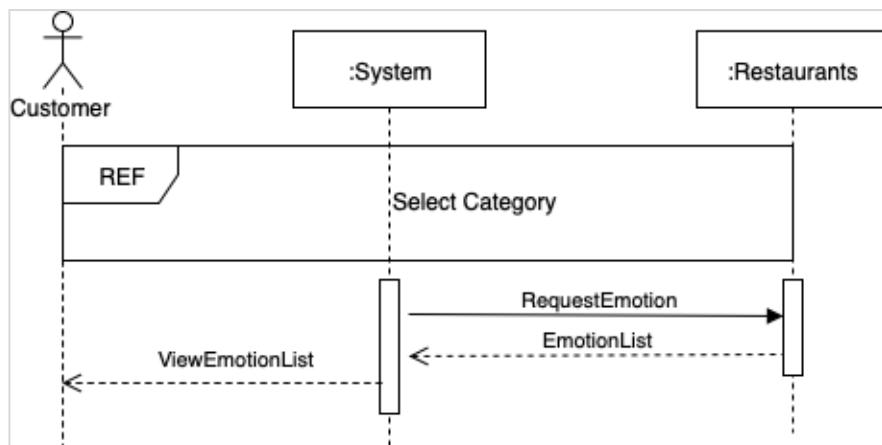


Figure3.5: View emotion list Sequential Diagrams

Retrieve comment:

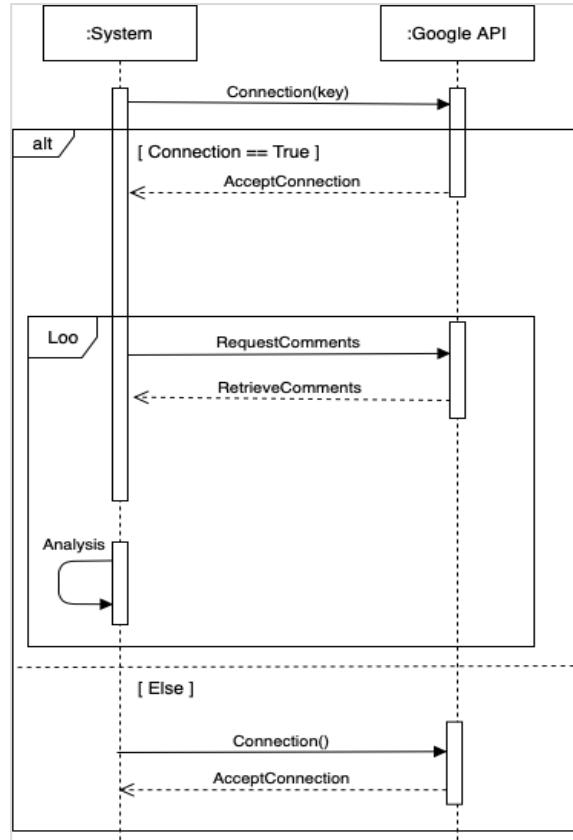


Figure3.6: Retrieve comment Sequential Diagrams

Retrieve tweets:

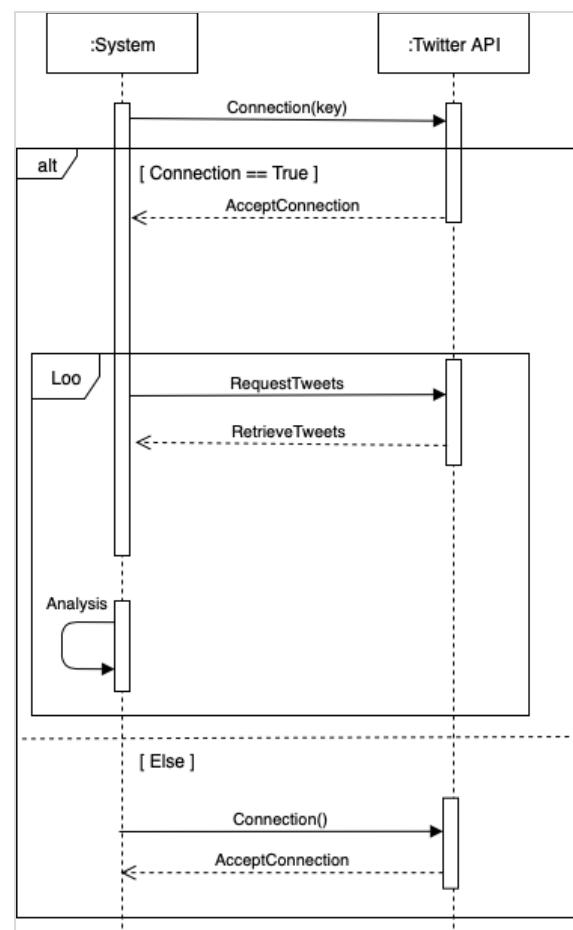


Figure3.7: Retrieve tweets Sequential Diagrams

Select Branch & View branch information:

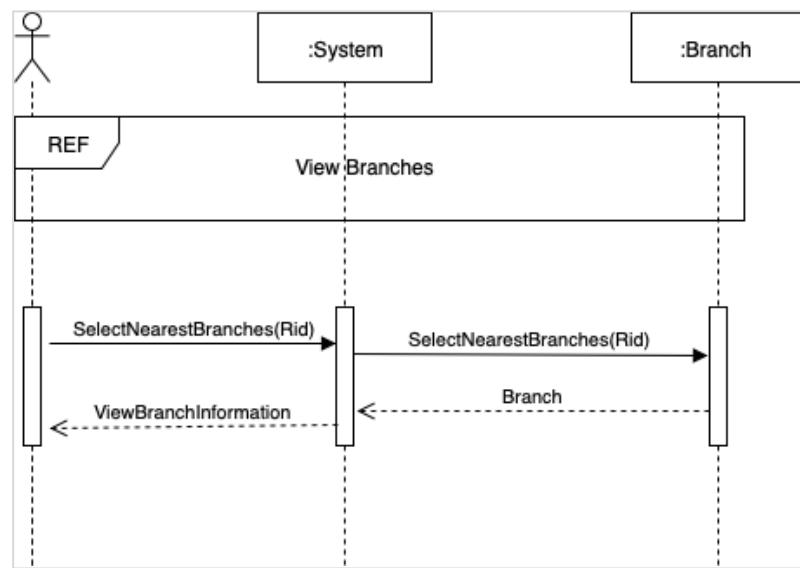


Figure3.8: Select branch & View branch information Sequential Diagrams

Determine branch location:

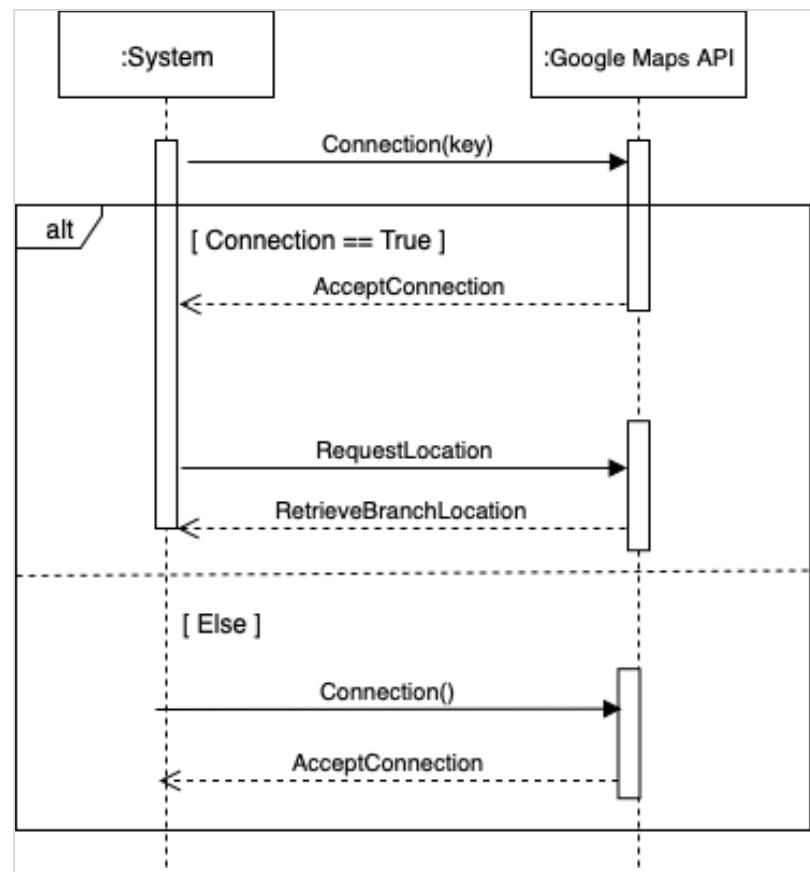


Figure3.9: Determine branch location Sequential Diagrams

Search for specific restaurant:

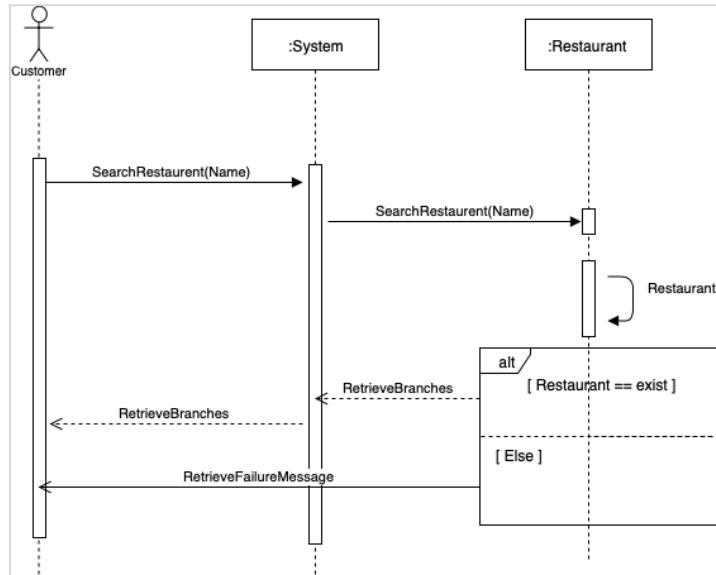


Figure3.10: Search for specific restaurant Sequential Diagrams

View branches:

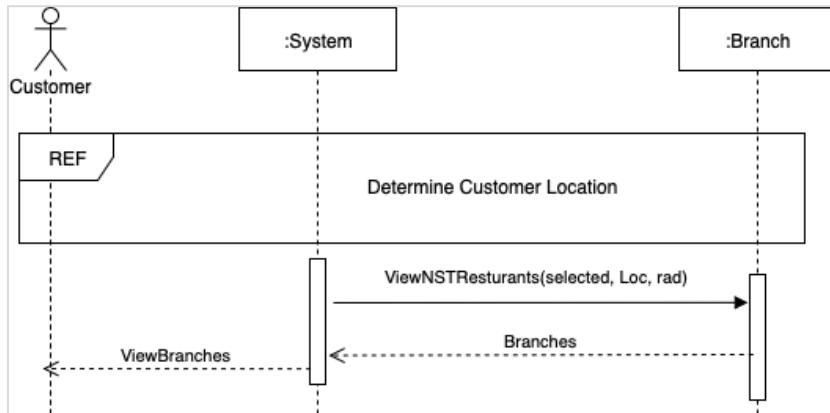


Figure3.11: View branches Sequential Diagrams

Select restaurant categories:

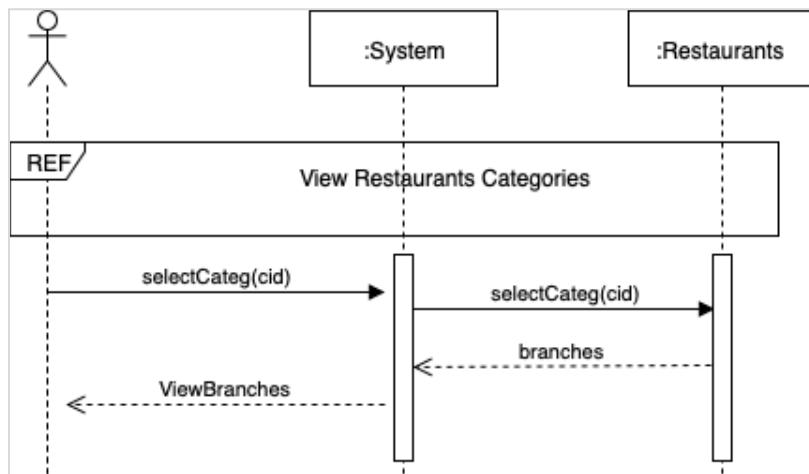


Figure3.12: Select restaurant categories Sequential Diagrams

Determine customer location:

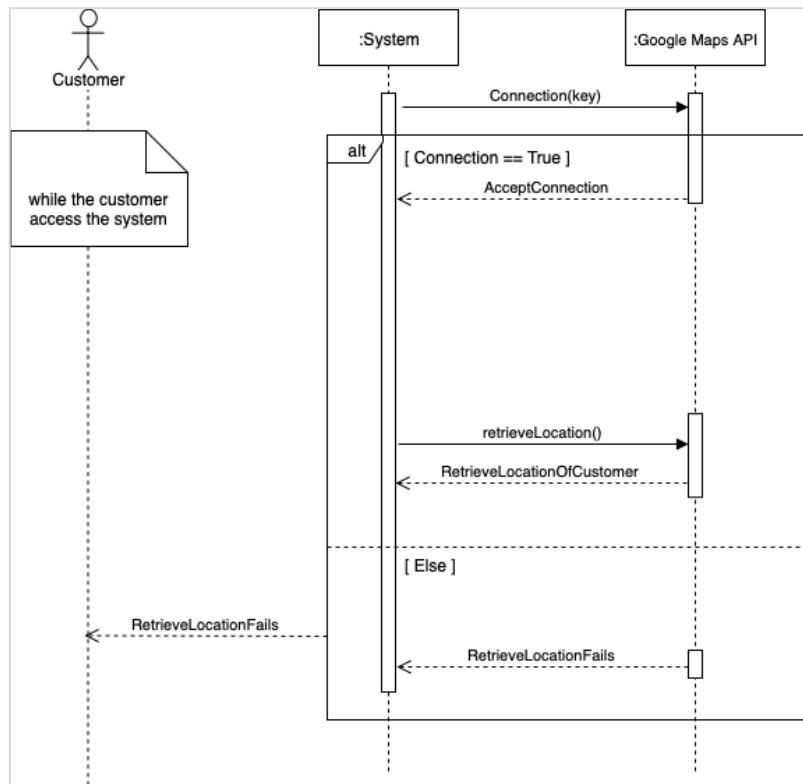


Figure3.13: Determine customer location Sequential Diagrams

Select emotions:

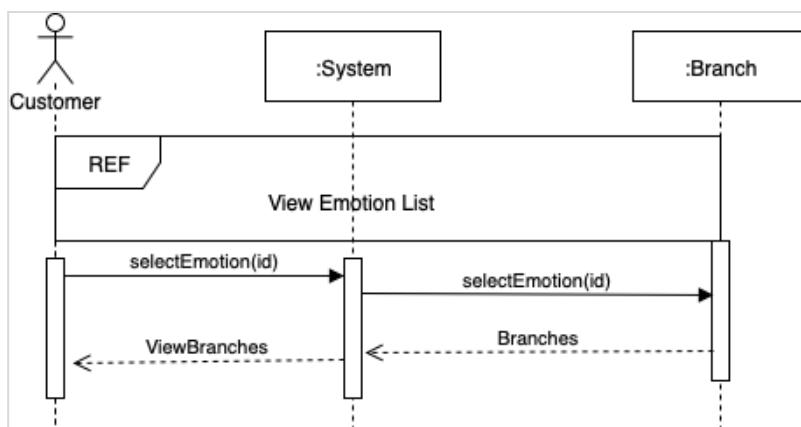


Figure3.14: Select emotions Sequential Diagrams

Login:

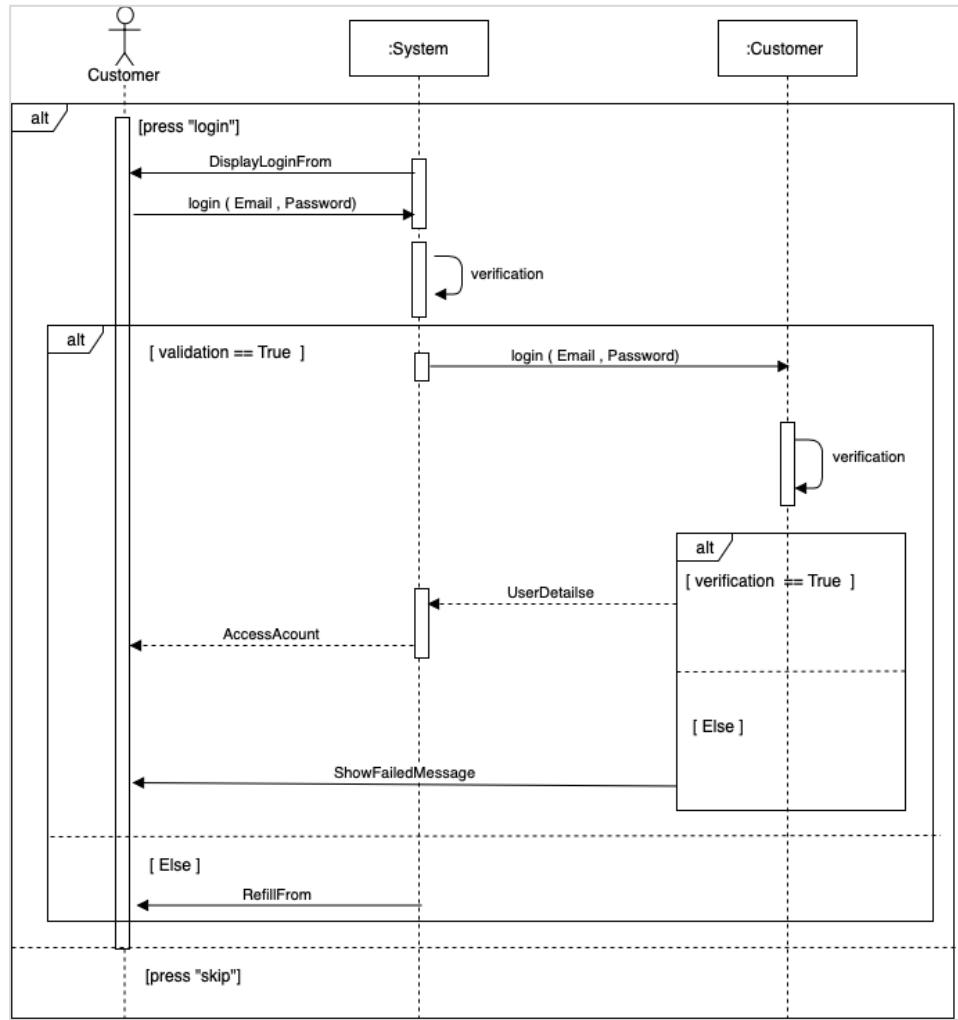


Figure3.15: Login Sequential Diagrams

Edit profile:

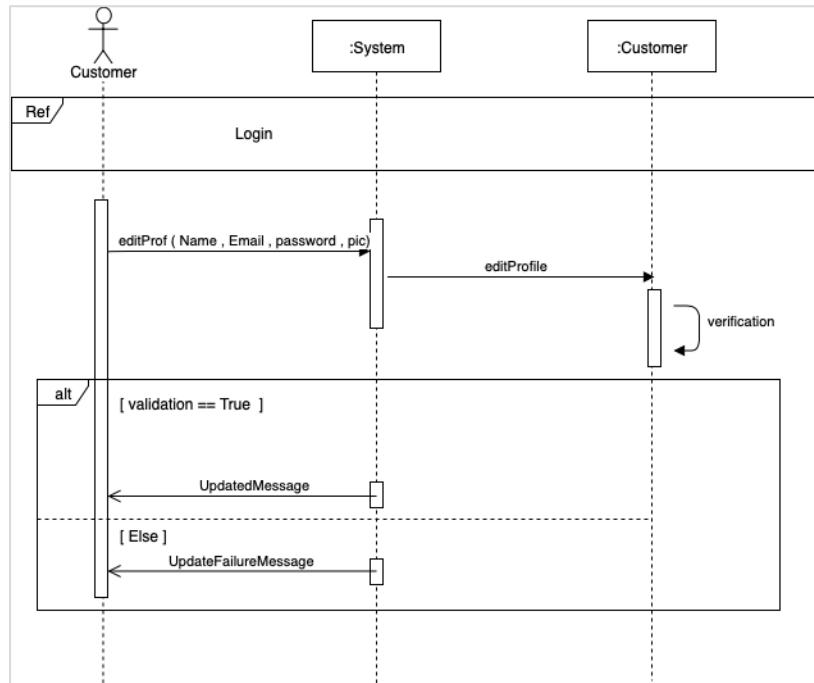


Figure3.16: Edit profile Sequential Diagrams

Registration:

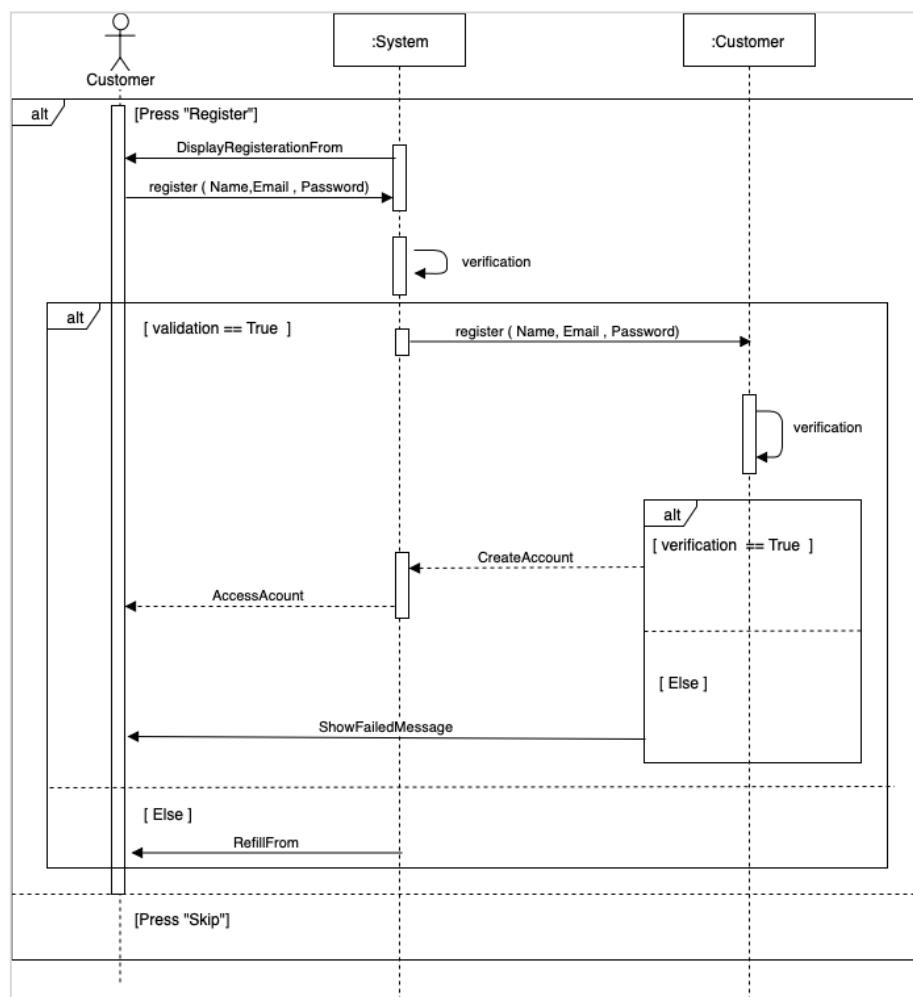


Figure3.17: Registration Sequence Diagram

Add branch to list:

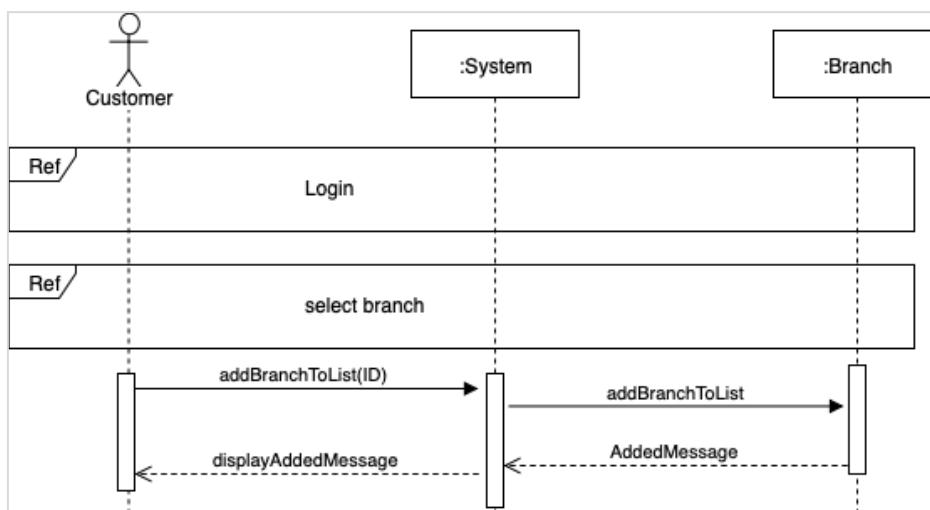


Figure3.18: Add branch to list Sequential Diagrams

View favorite list:

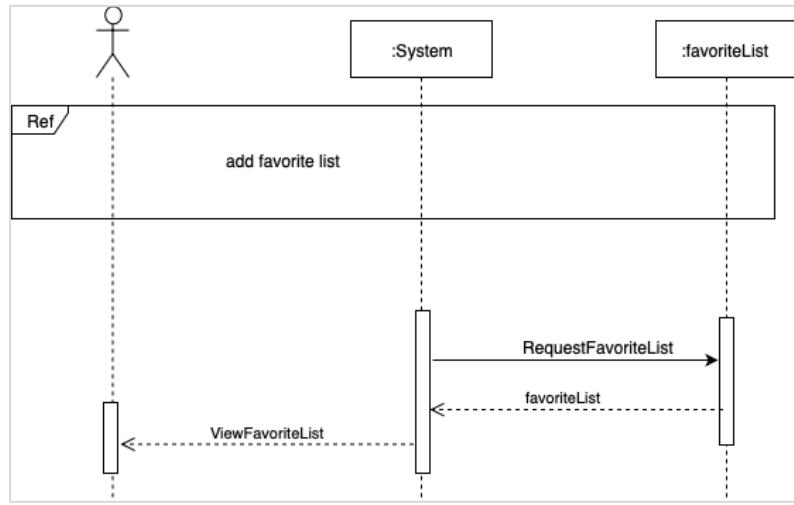


Figure3.19: View favorite list Sequential Diagrams

Remove branch from list:

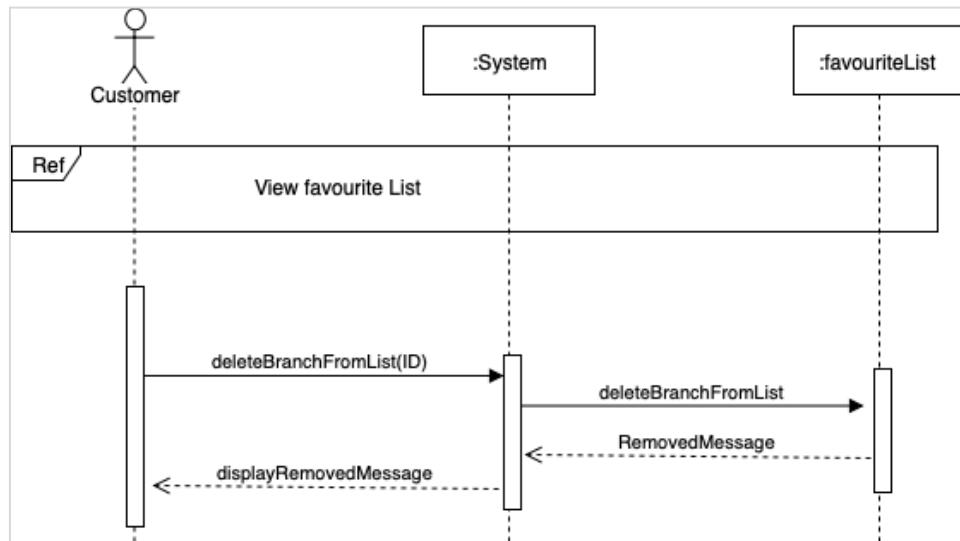


Figure3.20: Remove branch from list Sequential Diagrams

Add comments:

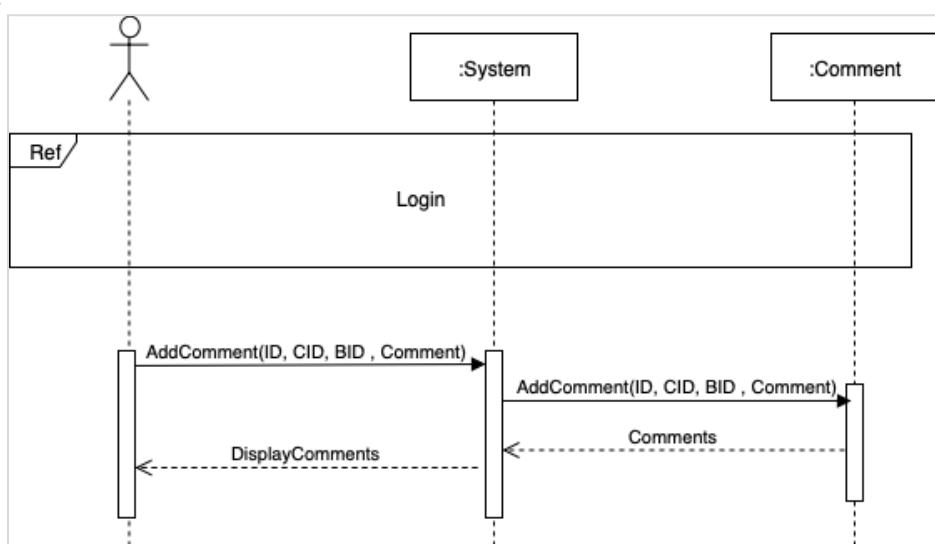


Figure3.21: Add comments Sequential Diagrams

View comments:

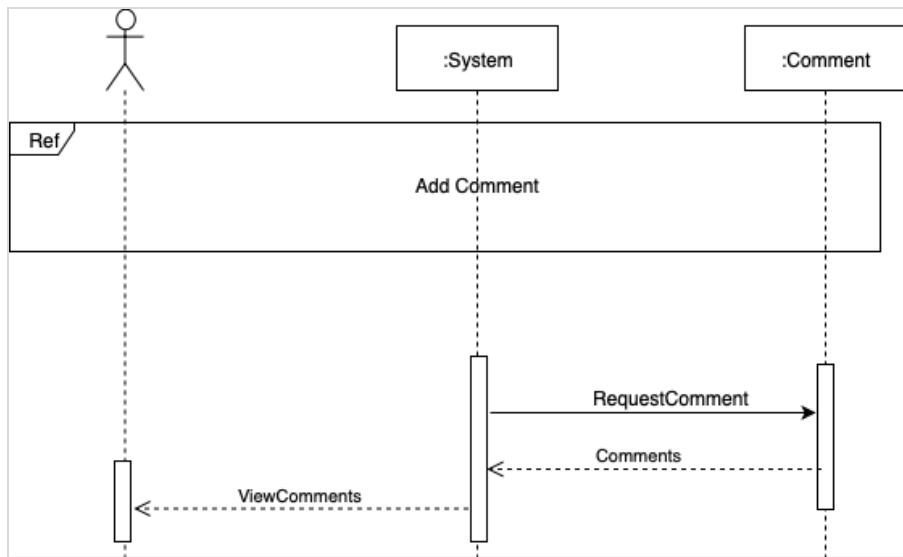


Figure3.22: View comments Sequential Diagrams

Delete comment:

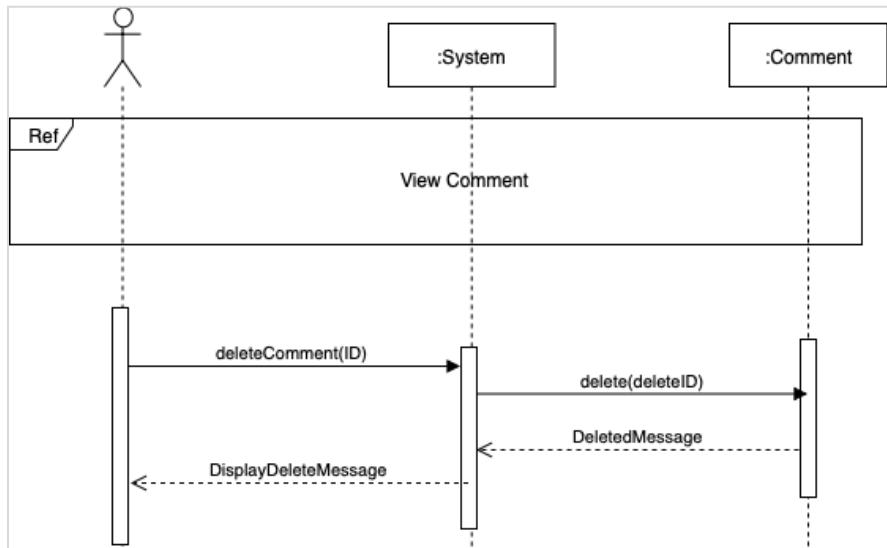


Figure3.23: Delete comments Sequential Diagrams

Upload picture:

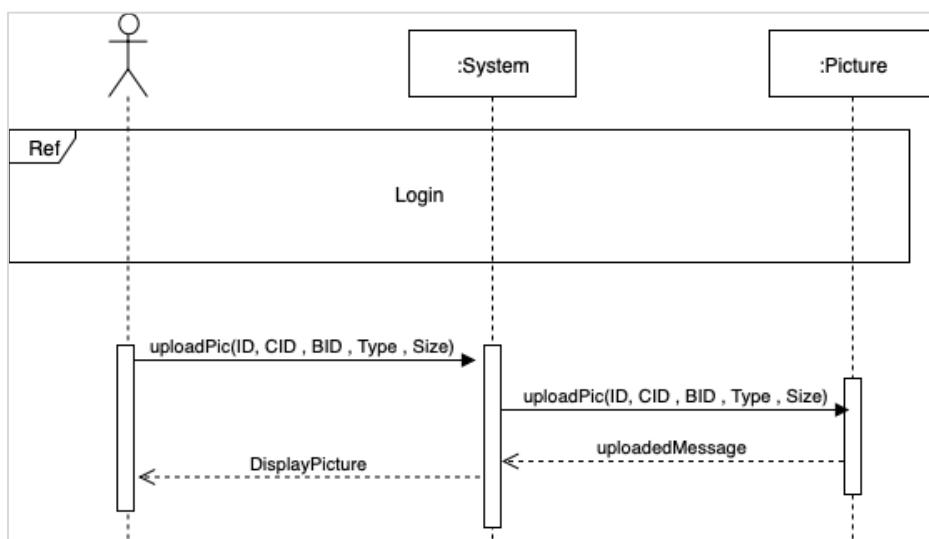


Figure3.24: Upload pictures Sequential Diagrams

View picture:

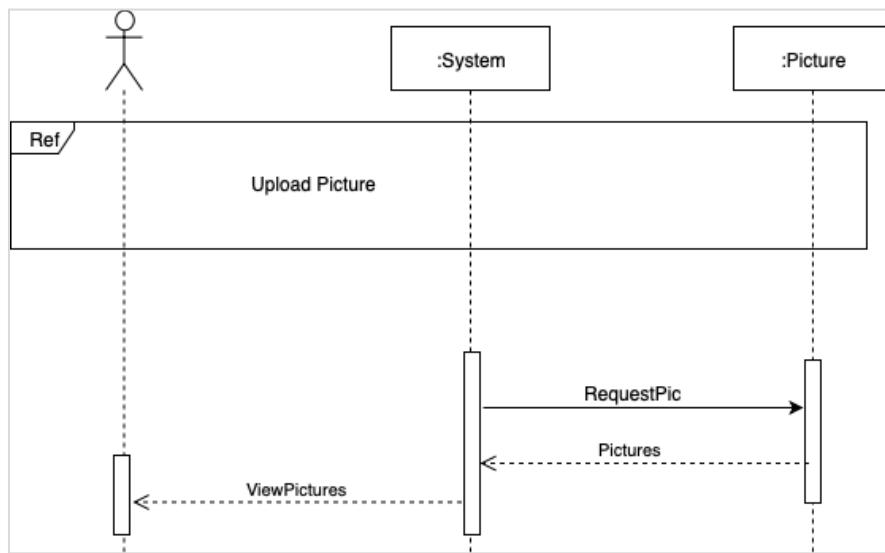


Figure3.25: View pictures Sequential Diagrams

Delete picture:

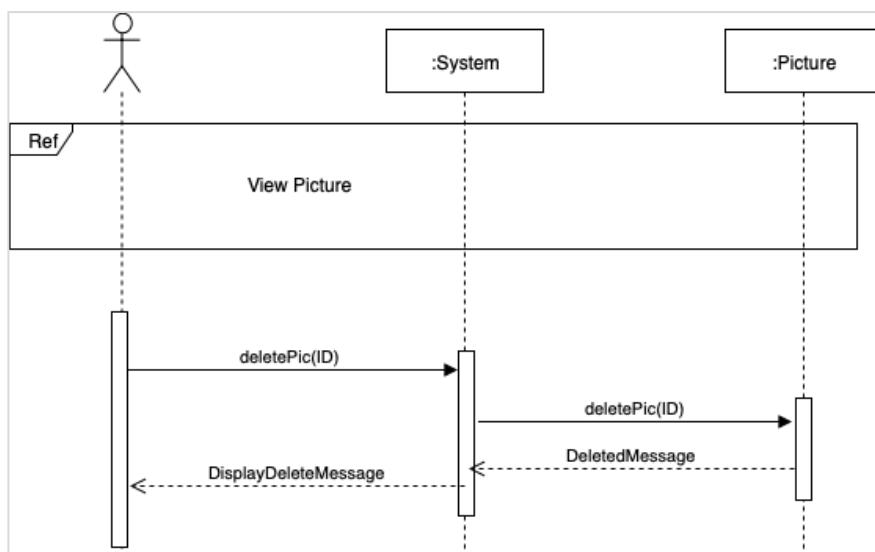


Figure3.26: Delete pictures Sequential Diagrams

Logout:

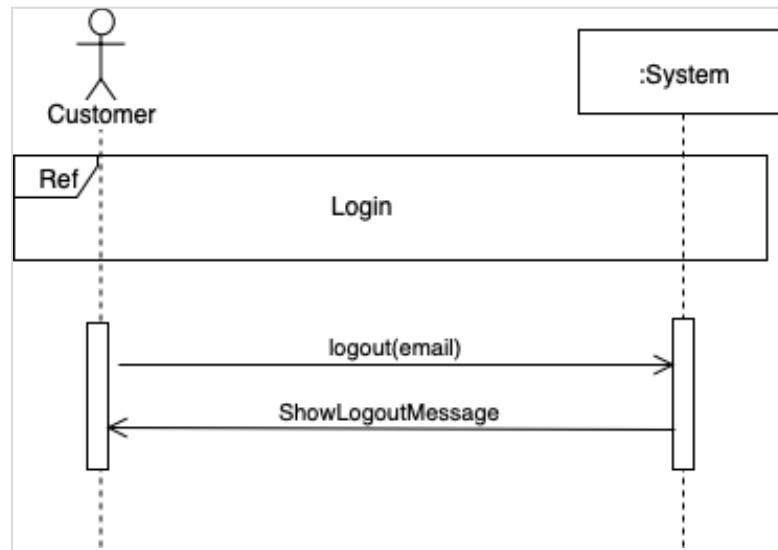


Figure3.27: Logout

3.6.4 DESIGN CLASS DIAGRAM

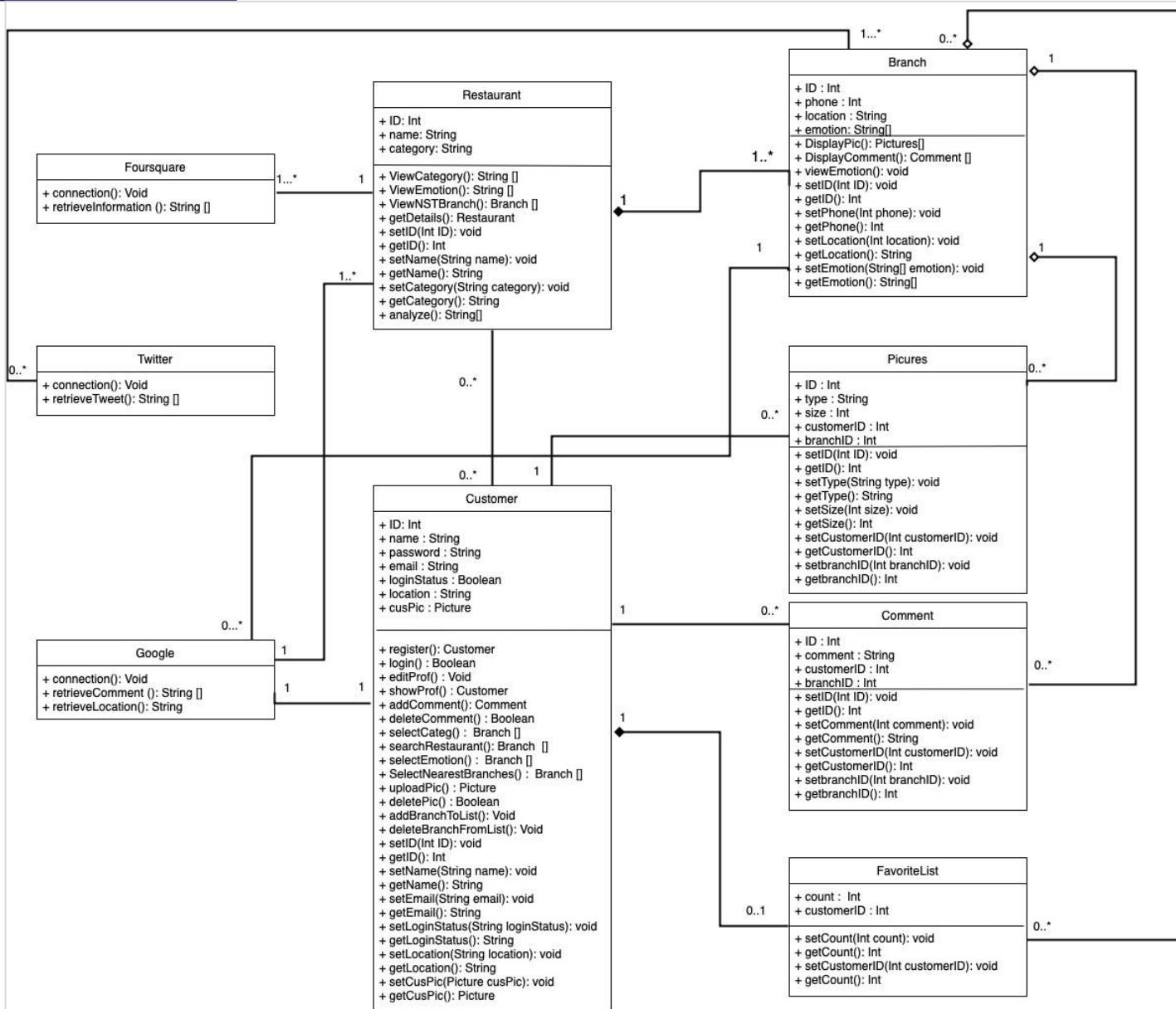


Figure3.28: Design class diagram

Chapter 4 | SYSTEM DESIGN



This chapter describes the system design which is the process of defining the suitable architecture, data design and user interface that meets our requirements presented in the previous chapters. The first section, we describe the system architecture that we use. Then we describe data design that starts with entity relationships (ER diagram). After that we present database schema and data dictionary which includes entity dictionary, relationship dictionary and attributes dictionary. Finally, a user interface design provided to visualize our application.

4.1 SYSTEM ARCHITECTURE

System architecture is "the fundamental organization of a system, embodied in its components, their relationships to each other and the environment, and the principles governing its design and evolution [30]. For our system we decide to use Three tier architecture.

Three-tier architecture allows any of the three tiers to be upgraded or replaced independently. The user interface is implemented on a Client and uses a standard graphical user interface with different modules running on the application server. The relational database management system on the database server contains the computer data storage logic. The three tiers are:

1. Presentation Tier: Occupies the top level and displays information related to services available on a website. This tier communicates with other tiers by sending results to the browser and other tiers in the network.
2. Application Tier: Also called the, logic tier, business logic or logic tier, this tier is pulled from the presentation tier. It controls application functionality by performing detailed processing.
3. Data Tier: Houses database servers where information is stored and retrieved. Data in this tier is kept independent of application servers or business logic [31].

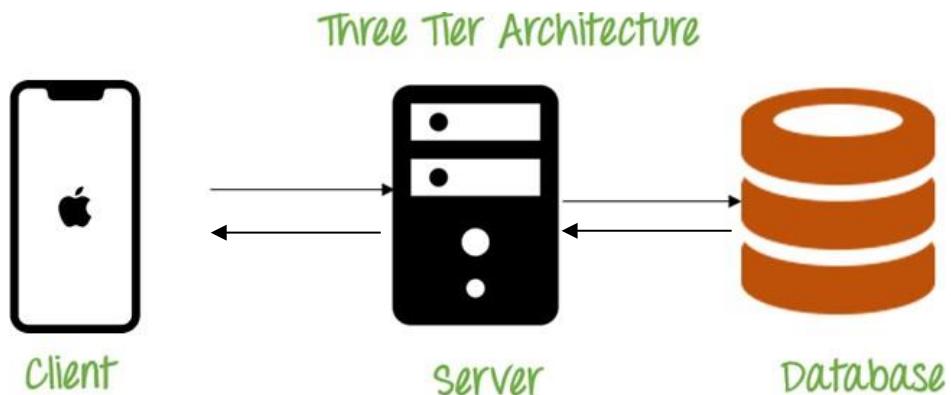


Figure4.1: Three Tier Architecture

4.2 DATA DESIGN

4.2.1 ENTITY RELATIONSHIP DIAGRAM

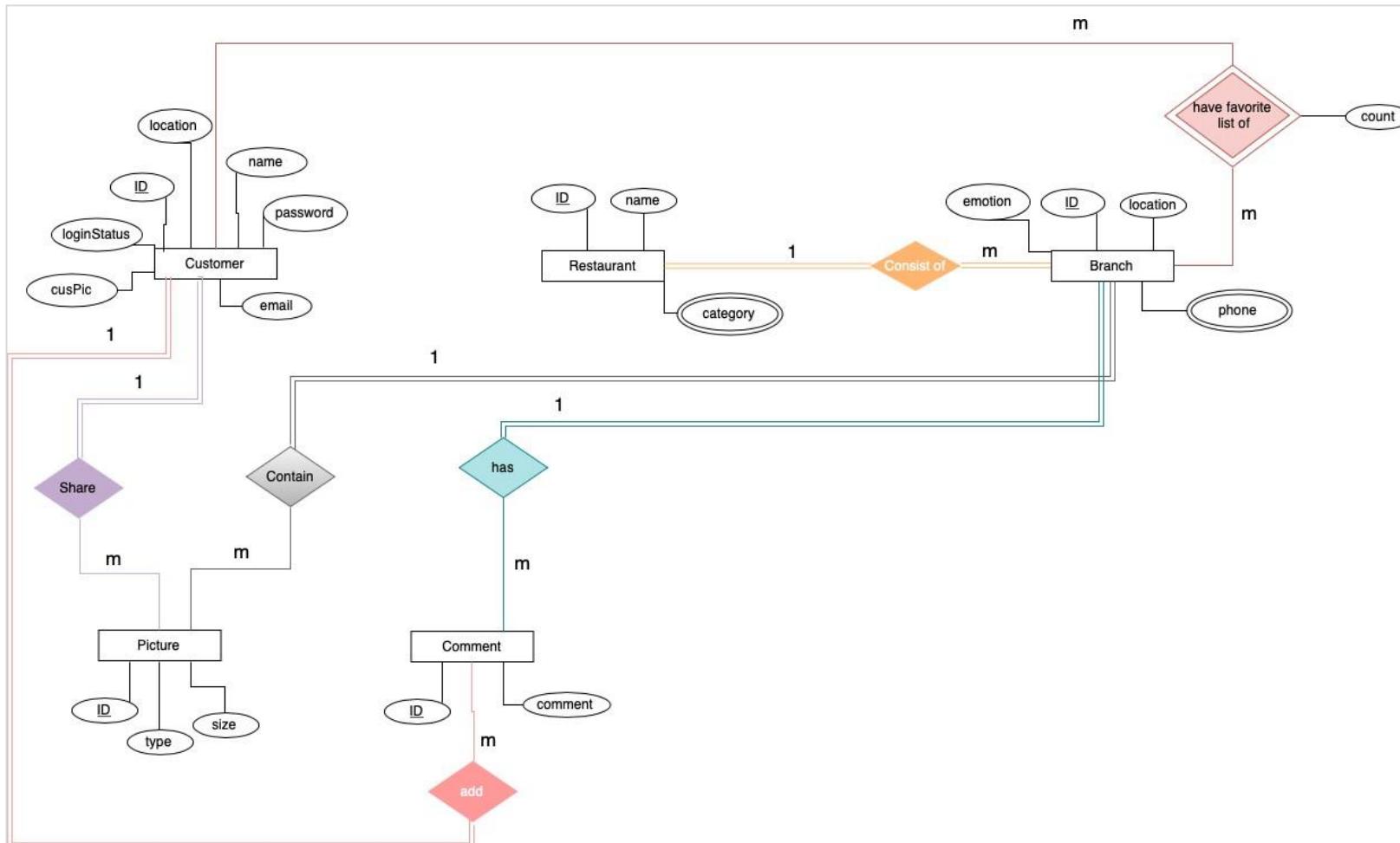


Figure4.2: Entity relationship diagram

4.2.2 DATABASE SCHEMA

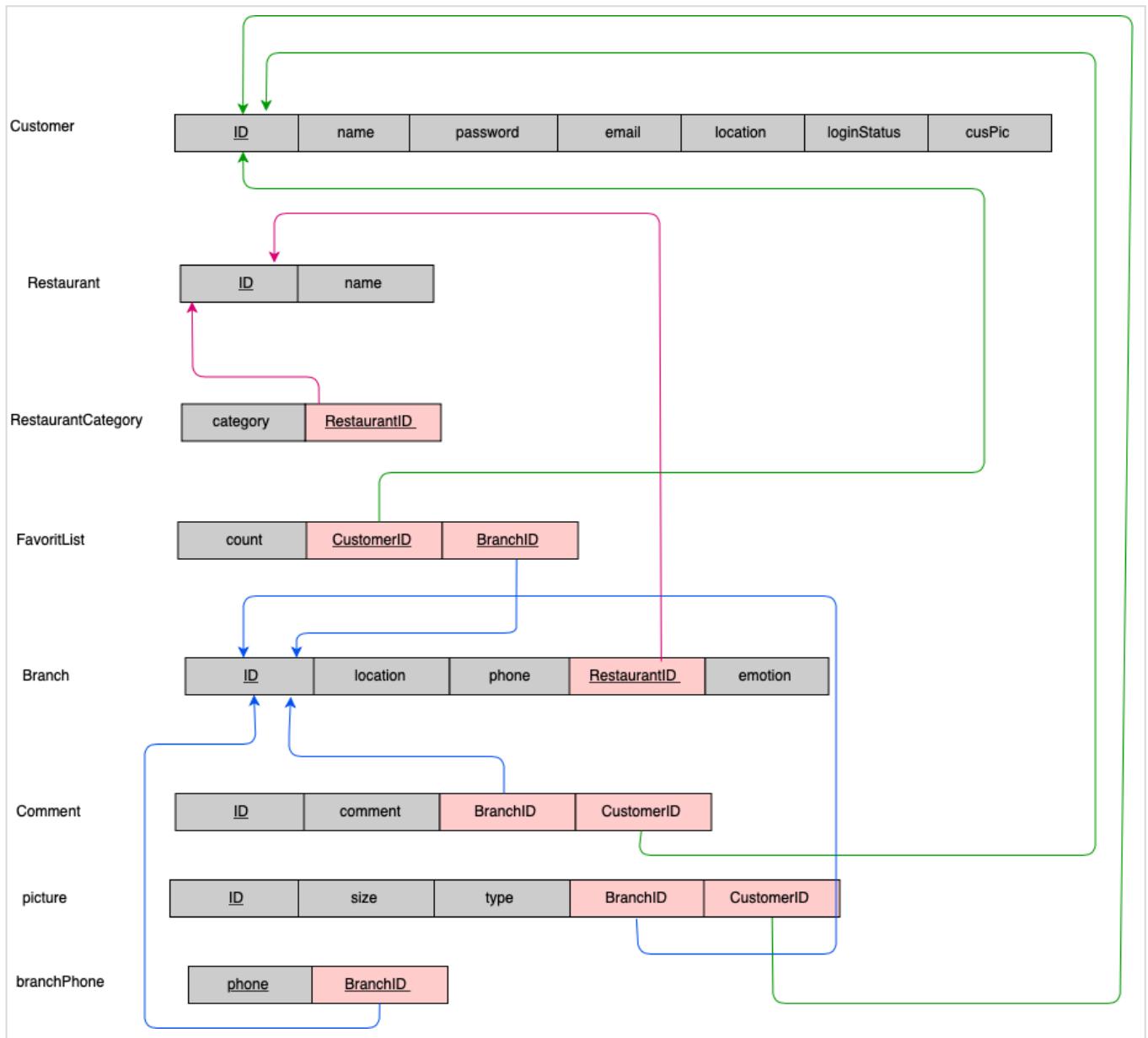


Figure4.3: Database schema

4.2.3 DATA DICTIONARY

Table4.1: Data Dictionary

Entity name	Description	Aliases	Occurrence
Customer	People who are using the application.	Customer	<ul style="list-style-type: none"> - Customers can register to the application. - Customer can log in to the application. - Customer can edit his/her profile. - Customer can select restaurant categories. - Customer can select branch. - Customer can select emotions. - Customer can search for a restaurant. - Customer can upload pictures. - Customer can remove picture. - Customer can upload comments. - Customer can remove comments. - Customer can add branch to his/her favorite list. - Customer can remove branch from his/her favorite list.
Restaurant	It's the place where people served by waiters who will provide menu of meals.	Restaurant	<ul style="list-style-type: none"> - Each restaurant has a category. - Each restaurant has a detailed information.
Branch	The restaurant is consisting of branches. That's mean it's the same restaurant on different locations.	Branch	<ul style="list-style-type: none"> - Each branch has its pictures, comments and rate. - Each branch can add to a favorite list. - Branches arranged by nearest to the customer. - Each branch selected by a customer. - Each branch has a specific emotion.
Favorite List	Collection of branches. Which are preferred by the customers.	Favorite List	<ul style="list-style-type: none"> - Each favorite list may have branches added by customer.
Picture	An image shared by a customer of a specific branch.	Picture	<ul style="list-style-type: none"> - Each picture can be uploaded by a customer. - Each picture can be removed by a customer.
Comment	Text that customers write to express their opinions.	Comment	<ul style="list-style-type: none"> - Each comment can be added by a customer. - Each comment can be removed by a customer.

Table4.2: Relationship Dictionary

Entity name	Multiplicity	Relationship	Entity name	Multiplicity
Customer	- (0, *) - (0, *) - (0,1) - (0, *)	- Select - Share - Manage - Add	- Restaurant - Picture - Favorite List - Comment	- (0, *) - (1,1) - (1,1) - (1,1)
Restaurant	- (0, *) - (1, *)	- Selected by - Consist of	- Customer - Branch	- (0, *) - (1,1)
Branch	- (0, *) - (1,1) - (0, *) - (0, *)	- Contained in - Belong to - May has - Contain	- Favorite List - Restaurant - Comment - picture	- (0, *) - (1, *) - (1,1) - (1,1)
Favorite List	- (1,1) - (0, *)	- Managed by - Involve	- Customer - Branch	- (1,1) - (0, *)
Picture	- (1,1) - (1,1)	- Exists in - Shared by	- Branch - Customer	- (0, *) - (0, *)
Comment	- (1,1) - (1,1)	- Appears in - Added by	- Branch - Customer	- (0, *) - (0, *)

Table4.3: Attributes Dictionary

Entity name	Attributes	Description	Data Type	Length	Null	Multivalued	PK
Customer	- ID - name - password - email - Location - login Status - cusPic	- Unique Identifier for customer. - Identifier for customer. - Used to verify the identity of user. - Short for electronic mail. - position of the customer. - Determination of customer registration. - Picture represented in customer profile	- INT - Varchar - Varchar - Varchar - Varchar - Boolean	- 5 - 15 - 10 - 45 - 80 - 4	No No No No No No	No No No No No No	Yes No No No No No
Restaurant	- ID - name - Category	- Unique Identifier for restaurant. - Identifier for restaurant. - Type of a restaurant's meals.	- INT - Varchar - Varchar - Varchar	- 5 - 15 - 15 - 10	No No No No	No No Yes Yes	Yes No No No
Branch	- ID - Location - Phone - emotion	- Unique Identifier for branch. - position of the branch. - Number to reach particular branch. - specific pattern of expression.	- INT - Varchar - long	- 5 - 80 - 10	No No No	No No Yes	Yes No No
Favorite List	- Count	- Number of branches in the list.	- INT	- 3	Yes	No	No
Picture	- ID - type - Size - CID - BID	- Unique Identifier for picture. - Picture extension. - picture size - Unique Identifier for customer. - Unique Identifier for branch.	- INT - Varchar - INT	- 5 - 5 - 5	No No No	No No No	Yes No No
Comment	- ID - Comment - CID - BID	- Unique Identifier for Comment. - impression of customer opinion. - Unique Identifier for customer. - Unique Identifier for branch.	- INT - Varchar	- 5 - 200	No Yes	No No	Yes No

4.3 USER INTERFACE DESIGN

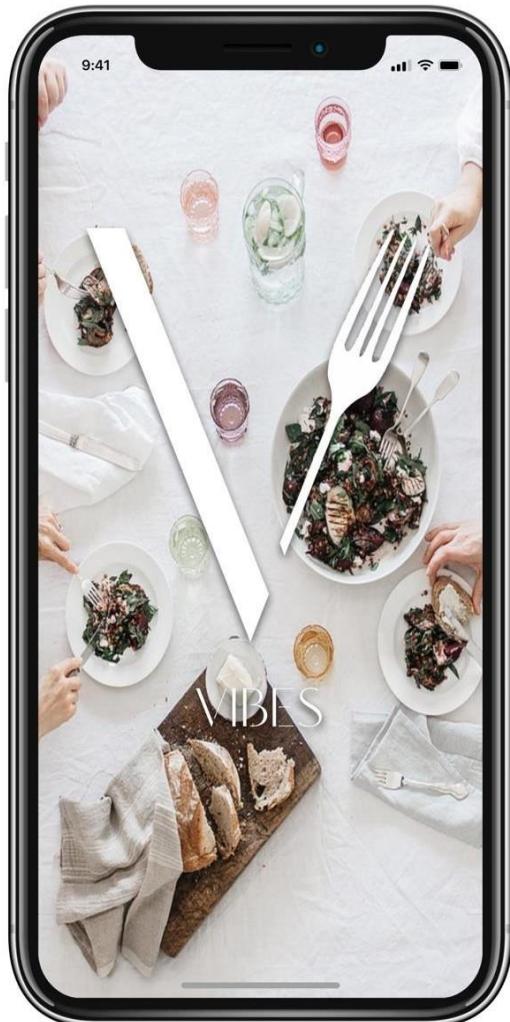


Figure4.4: Start screen



Figure4.5: Registration Screen



Figure4.6: Login Screen

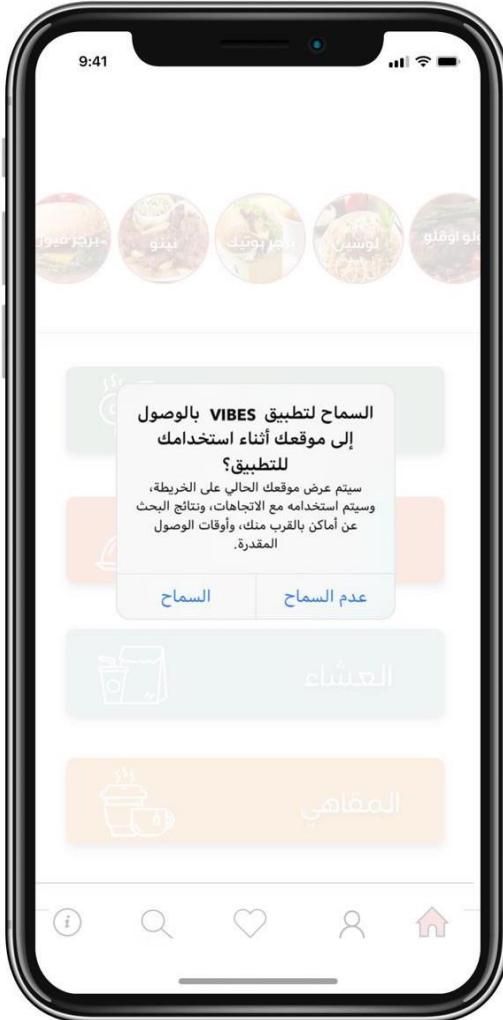


Figure4.7: Customer Location



Figure4.8: Home screen



Figure4.9: Restaurants screen



Figure4.10: Information screen



Figure4.11: Comment screen



Figure4.12: Pictures screen



Figure4.13: Restaurants for specific emotion screen



Figure4.14: Sentiment analysis rate screen

اسم المستخدم

شنود

كلمة المرور

البريد الإلكتروني

shahadalkhaiel@gmail.com

تعديل

Figure4.15: Edit profile screen



Figure4.16: Favorites list



Figure4.17: Search restaurants



Figure4.18: About VIBES

Chapter 5 | SYSTEM IMPLEMENTATION



In this chapter, we discuss system implementation and integration in detail. First, we present the hardware and the software tools that we used to develop our project. Then, we discuss the system integration by describing how the different components were integrated with each other to work as a whole. In the end, we show code samples for different components.

5.1 REQUIRED HARDWARE & SOFTWARE

5.1.1 HARDWARE

Table5.1:Required hardware.

Device	Quantity	Description
Laptop	5	Each member will need a laptop with either Windows or macOS operating systems for coding, searching and documentation.
iPhone	5	Different versions of iPhones used for application testing and searching.
iPad	3	Different versions of iPads used for application testing and searching.

5.1.2 SOFTWARE

Table5.2: Required software.

Category	Tool	Description
Programming language	Swift	Swift is a language developed by Apple . It is designed for writing apps for Apple platforms, including macOS, iOS, tvOS, and watchOS ^[32] .
	Python	Is an interpreted, object-oriented programming language, that has gained popularity because of its clear syntax and readability ^[33] .
Code editor	XCode	The integrated development environment (IDE) from Apple that is used to create, compile and test Mac OS X and iOS (iPhone/iPad/iPod) applications ^[34] .
	Idle	Its code editor that enables you to edit, run, and debug Python programs in a simple GUI environment ^[35] .
	Atom	Used as editor for python code.
	Visual Studio Code	Used as editor for python code.
Database	Real-time firebase	Cloud-hosted NoSQL database that lets you store and sync data between your users in real time ^[36] .
Server	PythonAnywhere	An online integrated development environment (IDE) based on the Python programming language.
Data Sources	Twitter API	Used to access and fetch Arabic tweets.
	Foursquare API	Used to access and get restaurants information.
	Google maps API	Used to detect current user location.
Libraries	Firebase	Gives you the tools to develop high-quality applications ^[37] .
	Firebase Database	Store and sync data with our NoSQL cloud database. Data is synced across all clients in real-time ^[38] .
	Firebase AUTH	The entry point of the Firebase Authentication SDK ^[39] .
	Firebase Storage	Is a service that supports uploading and downloading large objects to Google Cloud Storage ^[40] .
	SDWebImage	Provides an async image downloader with cache support.
	SVProgressHUD	Clean and lightweight progress HUD for your iOS ^[41] .
	TwitterKit	Twitter Kit is our open source iOS and Android SDK for displaying Tweets, authorizing Twitter users, and working with the Twitter API ^[42] .
	CoreLocation	Provides services for determining a device's geographic location ^[43] .

MapKit	Framework lets you embed a fully functional map interface into your application [44].
Alamofire	Swift-based HTTP networking library for iOS and macOS. It provides an elegant interface [45].
FoursquareAPIClient	Swift networking library for Foursquare API [46].
Async	It is composable, allowing you to build complex workflow and support error handling [47].
Tweepy	Open-sourced, hosted on GitHub and enables Python to communicate with Twitter platform and use its API [48].
NLTK	Provide function for streaming, tokenization [49].
json	Is a lightweight data-interchange format. It is easy for humans to read and write [50].
Pyarabic	Provides basic functions to manipulate Arabic text [51].
Flask	Framework for building complex web applications.

5.2 SYSTEM IMPLEMENTATION

5.2.1 IMPLEMENTATION PROCESS

The system consists of three main components:



Figuere5.1: Therr tier.

1. **Application Tier:** Which responsible for data fetching, preprocessing and emotion analysis by using python language.
2. **Database Tier:** Which responsible for store users accounts, posts and comments, favorite list by using firebase database.

We decide to use Firebase Real-Time database because through our research we noticed the lack of sources for Swift languages dealing with other types of database. Also, Firebase Real-Time database has many features that we need in our application. It provides Cloud-hosted real-time document store. iOS, Android, and JavaScript clients share one Real-time Database instance and automatically receive updates with the newest data [52]. Moreover, the Real-time Database allows developers to set up and provision a back-end database as a service more easily and integrates with Firebase Authentication to provide simple and intuitive authentication. Also, its low barrier of entry, low maintenance costs, and fast queries.

In addition, it provides both eventual and Immediate consistency approach. Eventual is a staggered operation, once the events are stored each of the projectors will process them at a later time where Immediate is an all or nothing operation, if anything goes wrong then the entire process is halted [53]. No events are stored, and no events are processed.

3. Presentation Tier: Which responsible for display nearest restaurants to the user with its emotion by using XCode program specifically swift language (IOS Application).

Our dataset is lexicon based it contain different emotions about restaurants terms used in Saudi Arabia. It has been created manually having 2000 words. Dataset is divided into six categories, each of them represent an emotion and contain words that reflect this emotion. The dataset was collected from questionnaire that ask people about emotions that they use to express their feeling about restaurant and from hashtags about specific restaurant in twitter. In addition, for each written word we tried to extract all possible derivatives. Dataset categories are happy, impressed, comfortable, angry, contrition and sad.

غاضب	نالم	حزين	منبهـ	سعـد	مرـاح
حسـه	كـنـيـب	ازـعـاج	مـدـهـش	يـمـي	زـيـن
مـقـاطـعـة	حـسـافـة	مـمـلـ	أـفـضـلـ	اسـعـدـنـي	مـرـضـي
تـوـبـه	ماـيـسـوـي	طـفـشـ	ادـمـانـ	يـالـبـيـهـ	مـرـيجـ
خـالـيـسـ	نـدـمـانـ	تعـيـسـ	روـعـهـ	مـمـتـعـ	لـطـيفـ

Fiquer5.2: Sample of lexicon.

1. Application Tier:

A. Data Fetching: Tweets collected by connection with Twitter API to retrieve all Arabic tweets for restaurants. And these tweets are filtered through send restaurant name as parameter, and location which is in Saudi Arabia only.

B. data Preprocessing: There is multiple preprocessing techniques in order to eliminate noisy and inconsistent data.

1. Tokenization: That is mean to extract words from a sentence by breaking sentence into words called tokens. We use in tokenization process NLTK library.

Example:

" جدا ، جميل ، المطعم جدا جميل". After tokenization " جدا ، جميل ، المطعم جدا جميل".

2. Remove Stop Words: Stop words appears frequently in most sentence, just to link words. It meaningless.

Sample of stop words:

[" لـعـلـ " ، " بـسـ " ، " لـكـنـ " ، " كـانـ " ، " لـيـتـ " ، " حـتـىـ " ، " ذـوـلـاـ " ، " هـذـيـ " ، " هـذـهـ " ، " كـلـ " ، " يـاـ " ، " عـلـىـ " ، " إـلـىـ " ، " مـنـ " ، " عـنـ "]

Example:

Before remove stop words: " هذا المطعم رهيب ". After removal: " المطعم رهيب ".

3. Remove Links, Hashtags and Mention: Links ("http://"), hashtag (#), and mention (@) all of these are not useful in emotion analysis.

Example:

"@sarah حلـوـ #ـالمـطـعـمـ رـهـيـبـ http://ninosrestaurant.com" After removal: " المـطـعـمـ رـهـيـبـ ".

4. Remove Arabic and English Numbers: Numbers are not important in our analysis so its removed.
Example:

٢. ”المطعم يعجبني“ After removal: ”مطعم يعجبني“

5.Remove punctuation: Arabic and English punctuation are removed for example:

“المطعم خطير” After removal: “خطير” المطعم

6.Remove Diacritics: Arabic diacritics removed.

Example:

“المطعم يهيل” After removal:

C. Data Analysis: After applying all pre-processing techniques. for this sentence “**يفتح النفس المطعم رايق**” we got this list of tokens **”النفس“ ، ”يفتح“ ، ”رايق“ ، ”المطعم“**. Now mapping each token to relevant emotion in our dataset based on classified terms. We found the word **”رايق“** under comfortable category. Then comfortable counter equal one. Another example the word **”سيء“** under angry. Then angry counter equal minus one. And so on until end of specific restaurants tweets. At the end, each emotion has specific number that.

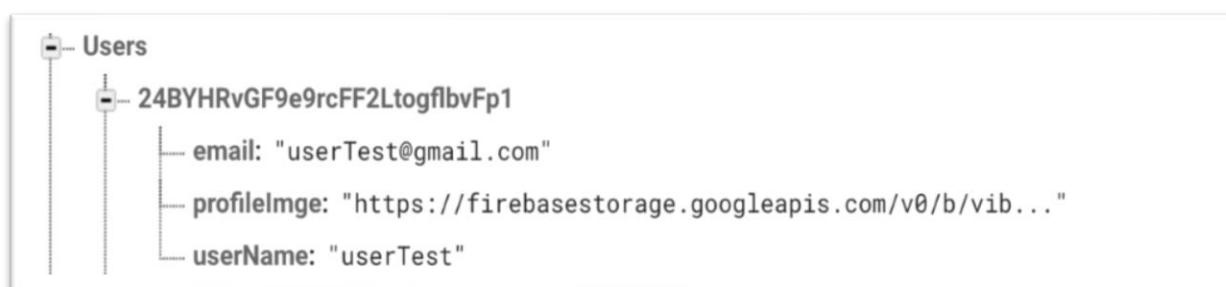
2. Data Tier: We use Firebase, that stores users account as shown in figure [2], user's images and comments for each restaurant as shown in figure [3] and users favorite list as shown in figure [4].

- Restaurant Object



Ficher5.3: Restaurant object.

- **Users Object**



wFiquer5.4: User object.

- Favorite list Object



Fiquer5.5: Favorite list object.

4. **Presentation Tier:** In this stage we use swift language to build VIBES application that serves Saudi Arabia population. VIBES provide different features such as present restaurants which is close to the user location, allow them to search about specific restaurant, upload picture and comment, add restaurant to favorite list and edit profile.

5.3 SYSTEM INTEGRATION

At the beginning we tried to apply analysis process by swift language but, we faced different difficulties since it's new language and there are no enough resources. Regarding of these issues we switch to another language which is python. consequently, we use PythonAnywhere which is an online integrated development environment (IDE) based on the Python programming language it allows to upload or download files. Based on that we upload Analysis files to be able to apply analysis and calculate restaurants rating. Also, it is preconfigured with loads of useful libraries that we need like, Flask which is a simple framework for building complex web applications. Then we integrate our analysis result with XCode. That we create real-time database in firebase and establish connection with XCode through google service sheet that responsible for establishing connection between them.

5.4 SAMPLE CODE

- Connection with Twitter API:

```

CONSUMER_KEY = "7S0Cq0a3Zwhc3luua8rJOE6J8"
CONSUMER_SECRET = "cyQKIP3AAo71Bj5AH12eeQr7LN830KQcHLDI1sd15t48x3ZBXo"
ACCESS_TOKEN = "805291100-0dq0733nuA3jl74fPpDw6KP771hAMV28SXWY06jk"
ACCESS_TOKEN_SECRET = "mEyMWoEHhSfPciXtTy2H8Ga56MzbeGuFFoG0aBXKZsNWa"
auth = tweepy.OAuthHandler(CONSUMER_KEY, CONSUMER_SECRET)
auth.set_access_token(ACCESS_TOKEN, ACCESS_TOKEN_SECRET)
api = tweepy.API(auth)
non_bmp_map = dict.fromkeys(range(0x10000, sys.maxunicode + 1), 0xffffd)
api = tweepy.API(auth, wait_on_rate_limit=True)
  
```

Fiquer5.6: Connection with twitter API.

- Connection with Foursquare API:

```
et url = "https://api.foursquare.com/v2/search/recommendations?ll=(coordin.latitude),\
(coordin.longitude)&section=food&v=20160607&intent=\
(name)&limit=20&client_id=ZMSMIQAE0PIKGYAUHBM4IMSFFQA4WXEZNG5FYUHGBABFPE3C&client_secret=KYOC41BAQCFKGM5FN0SUASNR5JAK1B4KMR204
M3CEPQEL4GO&oauth_token=NKRPOKY5ZDZIBMCU3Tzs4BMP4ZMIQZBQPLBTCPXSIGPWfJ1L"
```

Fiquer5.7: Connection with Foursquare API.

- Determine current user location:

```
locationManager = CLLocationManager()
locationManager.delegate = self

//set my location accuracy to best
locationManager.desiredAccuracy = kCLLocationAccuracyBest
locationManager.requestWhenInUseAuthorization()

//start the location service
locationManager.startUpdatingLocation()

//show my location
mapview.showsUserLocation = true|
```

Fiquer5.8: Determine current user location.

- Tweet Analysis:

```

#files
PositiveEmotion= open("PositiveEmotion","r")
tweet=open("tokenize","r")
negativeEmotion=open("negativeEmotion","r")
#seek
PositiveEmotion.seek(0,0)
tweet.seek(0,0)
negativeEmotion.seek(0,0)
#lists
PositiveEmotionList=PositiveEmotion.readlines()
tweetList=tweet.readlines()
negativeEmotionList=negativeEmotion.readlines()
#positive emotions

for emo in PositiveEmotionList:
    # to obtain number
    ratePositive=emo.split("~")[0]
    rateNumP=int(ratePositive)
    # to obtain word
    emotionP=emo[2:]

    for tweP in tweetList:
        if tweP == emotionP:
            if rateNumP == 1:
                comfortable1+=1
            if rateNumP == 2:
                happy2+=1
            if rateNumP == 3:
                impressed3+=1

```

Fiquer5.9: Tweets analysis.

- Tweet Pre-processing:

```
# remove links
text = re.sub(r'^(http\S+)', '', text)
text = re.sub(r'^((ht|f)tp\S+)', '', text)

# remove @
text = re.sub(r'^(@[a-zA-Z0-9_])', '', text)

# remove #
text=text.replace("#", "").replace("_", " ")

# remove numbers single character
text = re.sub(r'^([a-zA-Z0-9_])', '', text)

#remove arabic numbers
text = re.sub(r'^[\u0621-\u064A]+', '', text)

#remove punctuation
for c in string.punctuation:

    text= text.replace(c,"")

# detect emojis
emo= re.findall(r'^[\w\s,]', text)

#tatweel
text=araby.strip_tatweel(text)

#harakat
text=araby.strip_tashkeel(text)

#حروف متراكبة، توحيدها يرجعها إلى حروف بسيطة
text=araby.normalize_ligature(text)

#tokenize
words=word_tokenize(text)

#remove stopwords

for w in words:

    if w in stop_words:

        words.remove(w)
```

Fiquer5.10: Tweets pre-processing.

- Classify Restaurants Under Specific Emotion:

```

@IBAction func segmentedAction(_ sender: Any) {
    searchBar.text = ""
    switch segmentedControl.selectedSegmentIndex {
    case 3:
        isFiltered = false

    case 2:
        isFiltered = true
        filtereRestaurant = restaurantDetails.filter { restaurant -> Bool in
            restaurant.feeling == "سعید"
        }
    case 1:
        isFiltered = true
        filtereRestaurant = restaurantDetails.filter { restaurant -> Bool in
            restaurant.feeling == "منبهر"
        }
    case 0:
        isFiltered = true
        filtereRestaurant = restaurantDetails.filter { restaurant -> Bool in
            restaurant.feeling == "مریج"
        }
    default:
        break
    }
    tableView.reloadData()
}

```

Fiquer5.11: Classify Restaurants under specific emotion.

- Add Restaurant to Favorite List:

```

func configureLikedRestaurant() {
    isLiked = LikedRestaurant.shared.restaurantList.contains { (details) -> Bool in
        details.restaurantId == restaurant.restaurantId
    }

    if isLiked {
        LikeButton.title = "إزالة الإعجاب"
    }else{
        LikeButton.title = "اعجاب"
    }
}

@objc func likeRestaurant() {
    let userID = Auth.auth().currentUser?.uid

    if isLiked {
        DataService.instance.REF_Users.child(userID!).child("Likes").child(restaurant.restaurantId).removeValue()
        LikedRestaurant.shared.restaurantList = LikedRestaurant.shared.restaurantList.filter{$0.restaurantId != restaurant.restaurantId}
    }else{
        LikedRestaurant.shared.restaurantList.append(restaurant)
        DataService.instance.REF_Users.child(userID!).child("Likes").child(restaurant.restaurantId).setValue(restaurant.restaurantId)
    }

    isLiked = !isLiked
    configureLikedRestaurant()
}

```

Fiquer5.12: Add specific restaurant to favorite list.

Chapter 6 | SYSTEM TESTING



System Testing is a level of software testing where a complete and integrated software is tested. The purpose of this test is to evaluate the system's compliance with the specified requirements [54]. In this chapter we apply various types of testing which are unit testing, integration testing, performance testing and user acceptance testing and lastly, we clarify a set of test cases of all the important functions in our system.

6.1 UNIT TESTING

Unit testing is a level of software testing where individual units/components of a software are tested. The purpose is to validate that each unit/function of the software performs as designed. A unit is the smallest testable part of any software [55]. Based on that we are going to apply this type of testing on each function in our application.

Table6.1: Unit testing.

Function No.	Function name	Description	Result
F1	Registration	Any one should be able to register through fill required field.	Pass
F2	Login	Any of the users should be able to log in to their accounts using valid data.	Pass
F3	Determine customer location	When customer open application, pop up message will appear to ask user for his location	Pass
F4	View trending restaurants	Most trending restaurants from Foursquare will appear in Home page	Pass
F5	View restaurants category	Once user accesses the application and his location determined, category will appear	Pass
F6	Select restaurants category	User is able to select category	Pass
F7	View emotion list	List of positive emotion will appear after selection category	Pass
F8	Select emotion	Once user select emotion from list, restaurant with the same emotion will appear.	Pass
F9	Analyze tweet	Ability to apply preprocessing technique on each tweet and classify tweet to specific emotion.	Pass
F10	Search for specific restaurant	User able to search for specific restaurant by its name.	Pass
F11	View restaurants	Shows restaurants list to user	Pass
F12	Select specific restaurant	User will be able to select specific restaurant	Pass
F13	View restaurant information	Restaurant information will be appearing to the customer.	Pass
F14	Add comment	User will be able to add comment	Pass
F15	Upload picture	User will be able to upload picture	Pass

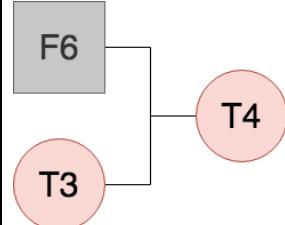
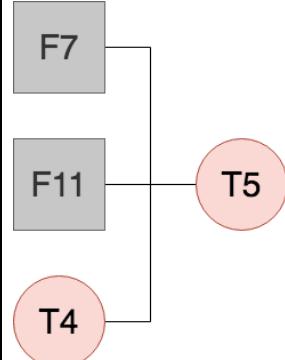
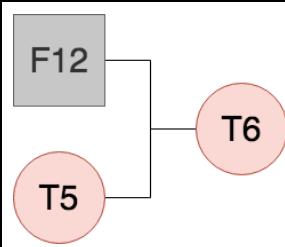
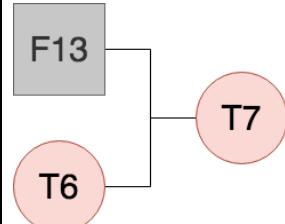
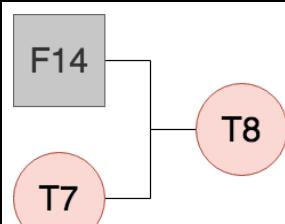
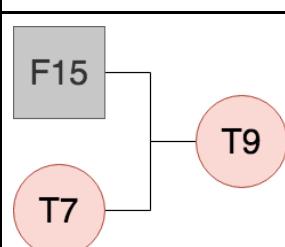
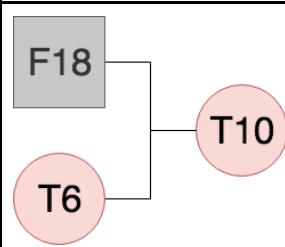
F16	Edit user profile	User will be able to edit profile information	Pass
F17	View favorite list	user will be able to view selected restaurants	Pass
F18	Add restaurant to favorite list	user will be able to add restaurants to favorite list	Pass
F19	Remove restaurant from favorite list	user will be able to remove restaurants to favorite list	Pass

6.2 INTEGRATION TESTING

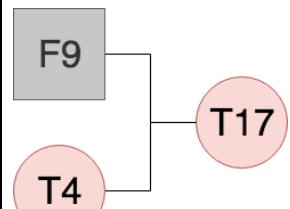
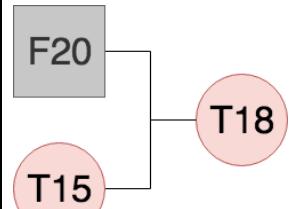
Is a level of software testing where individual units are combined and tested as a group. The purpose of this level of testing is to expose faults in the interaction between integrated units. Test drivers and test stubs are used to assist in Integration Testing [56].

Table6.2: Integration testing.

Test Number	Functions Numbers	Description	Result	Graphs
T1	F1, F2	Validate user registration and login	Pass	<pre> graph LR F1[F1] --- T1((T1)) F2[F2] --- T1 </pre>
T2	T1, F3	Detect user location when the T1 is valid	Pass	<pre> graph LR T1((T1)) --- F3[F3] </pre>
T3	T2, F4, F5	Home page will be present when the T2 is valid	Pass	<pre> graph LR T2((T2)) --- F4[F4] T2 --- F5[F5] F4 --- T3((T3)) F5 --- T3 </pre>

T4	T3, F6	User will be able to select specific category when T3 is valid	Pass	
T5	T4, F11,F7	Nearest restaurants page will be present when T4 is valid	Pass	
T6	T5, F12	User will be able to select specific restaurant when T5 is valid	Pass	
T7	T6, F13	Restaurant information page will be present when T6 is valid	Pass	
T8	T7, F14	User will be able to write comment when T7 is valid	Pass	
T9	T7, F15	User will be able to upload picture hen T7 is valid	Pass	
T10	T6, F18	User will be able to add restaurant to favorite list picture when T6 is valid	Pass	

T11	T3, F17	User will be able to view added restaurants to favorite list when T7 is valid	Pass	<pre> graph LR F17[F17] --- T11((T11)) T3((T3)) --- F17 </pre>
T12	T3, F12, F13	User will be able to select specific restaurant from trending bar when T3 is valid	Pass	<pre> graph LR F12[F12] --- T12((T12)) F13[F13] --- T12 T3((T3)) --- F12 </pre>
T13	T3, F17, F19	User will be able to remove restaurants from favorite list when T10 is valid	Pass	<pre> graph LR F17[F17] --- T13((T13)) F19[F19] --- T13 T3((T3)) --- F17 </pre>
T14	T5, F10	User will be able to search for specific restaurant when T5 is valid	Pass	<pre> graph LR F10[F10] --- T14((T14)) T5((T5)) --- F10 </pre>
T15	T5, F8, F11	User will be able to select specific restaurant with specific emotion when T5 is valid	pass	<pre> graph LR F8[F8] --- T15((T15)) F11[F11] --- T15 T5((T5)) --- F8 </pre>
T16	T3, F16	User will be able to edit profile information when T1 is valid	Pass	<pre> graph LR F16[F16] --- T16((T16)) T3((T3)) --- F16 </pre>

T17	T4, F9	when user select specific category analysis result will present for each restaurant when T5	Pass	
T18	T15, F20	User will be able to logout when T15 is valid	Pass	

6.3 PERFORMANCE TESTING

Performance Testing is a type of software testing to ensure software applications will perform well under our expected workload [57]. to determine the speed, responsiveness and stability of the application [58]. This test was performed to test how VIBES will behave under load and stress, this task was accomplished by XCode.



Figure6.1: Disk performance.

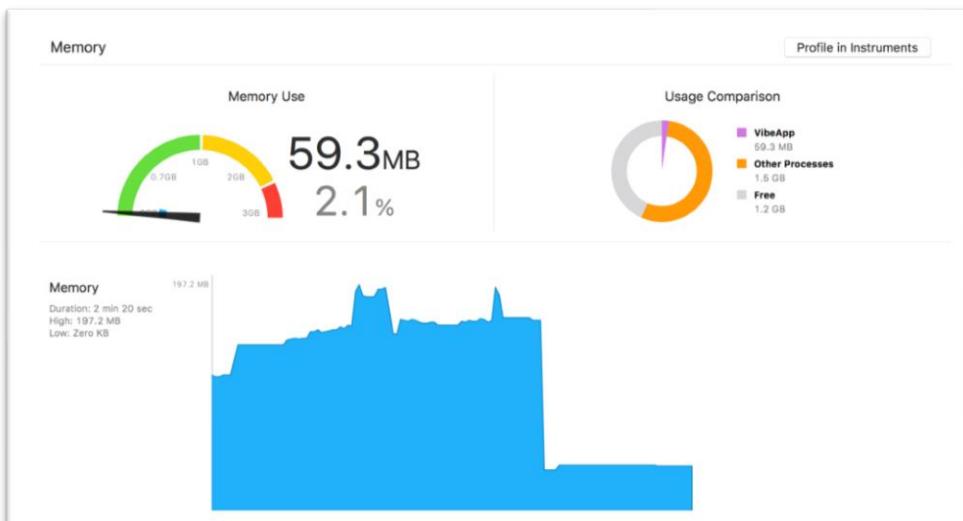


Figure6.2: Memory performance.

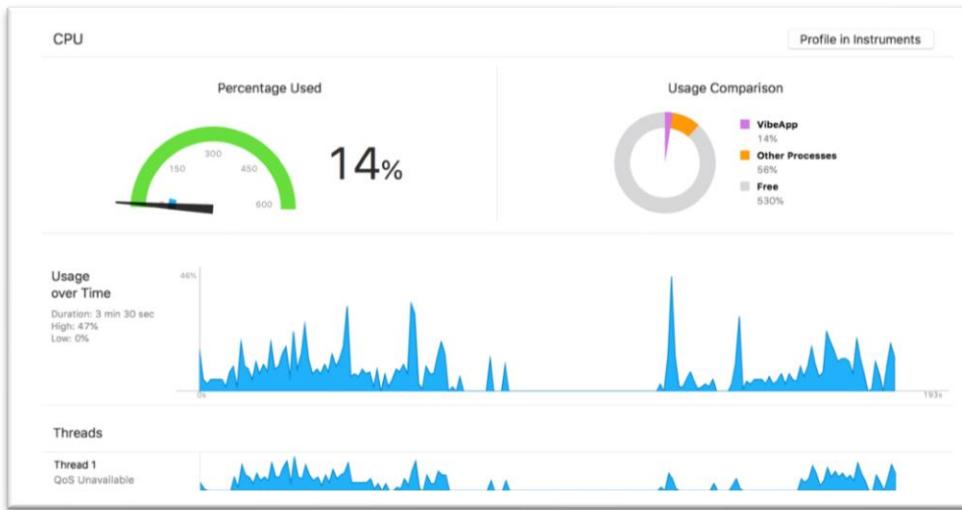


Figure6.3: CPU performance.

6.4 USER ACCEPTANCE TESTING

User acceptance testing (UAT) is the last phase of the software testing process. During UAT, actual software users test the software to make sure it can handle required tasks in real-world scenarios, according to specifications. UAT is one of the final and critical software project procedures that must occur before newly developed software is rolled out to the market [59].

Table6.3: 1st user acceptance testing.

Tester Name	Miral
Task	Feedback
Sign Up Screen	Clear and convenient but it should display a message when the user selects a picture.
Home Screen	Display the trendy restaurants and categories the restaurants based on the type of food its serves.
Restaurants Screen	Provide a quick access to the restaurants based on the visitors reviews and comments. It's contained many features such as the distance, rating and search.
Restaurant Information Screen	Collects and saves all the favorite restaurants in one screen and provide a quick access to the favorite restaurants when need it.
Favorite List Screen	Collects and saves all the favorite restaurants in one screen and provide a quick access to the favorite restaurants when need it.
Edit Profile Screen	Allows the user to edit his information in an easy way.
About Us Screen	I like the design of the screen and it's provided an overview of the application in a simple and brief way.
Analysis Result	Shows the rating in an attractive way.

Table6.4: 2nd user acceptance testing.

Tester Name	Reema
Task	Feedback
Sign Up Screen	Simple and clear.
Home Screen	Perfect.
Restaurants Screen	Organized and simple.
Restaurant Information Screen	Organized and simple.
Favorite List Screen	I LIKE how the application splits the favorite list based on moods.
Edit Profile Screen	Easy to use.
About Us Screen	Simple and clear.
Analysis Result	GREAT

Table6.5: 3rd user acceptance testing.

Tester Name	Nouf
Task	Feedback
Sign Up Screen	Simple and beautiful
Home Screen	Fantastic
Restaurants Screen	Easy to use and I really like the emoji of each restaurant to identify the mood of people who went to the restaurant.
Restaurant Information Screen	Easy to read and beautiful
Favorite List Screen	What I like in favorite list is each mood will filter into different list to make easy for user
Edit Profile Screen	Simple and beautiful
About Us Screen	Fantastic
Analysis Result	Easy to read and very helpful

Table6.6: 4th user acceptance testing.

Tester Name	Ahad
Task	Feedback
Sign Up Screen	Simple & clear design, however, there should be a pop-up message indicating to choose a photo.
Home Screen	Loved that the places are sorted and categorized.
Restaurants Screen	Clear and convenient display of the restaurants.
Restaurant Information Screen	Contains the most important information regarding the restaurant. What amazed me the most was the integration with Twitter.
Favorite List Screen	A very convenient feature, provides the user fast accessibility.
Edit Profile Screen	Simple & clear, but the profile photo I choose when signing up did not appear.
About Us Screen	Delivered clear message of the application, loved that it was in Arabic.
Analysis Result	This specific feature is what makes this application different than the others, the user can choose based on other's experience.

Table6.7: 5th user acceptance testing.

Tester Name	Raja
Task	Feedback
Sign Up Screen	Good, I like it.
Home Screen	I like how it is organized, but it takes long time to be loaded.
Restaurants Screen	Good idea that you classify Restaurants based on category, but the images are not clear.
Restaurant Information Screen	I like it, It contains a useful information that I need.
Favorite List Screen	Good.
Edit Profile Screen	Easy to use. but needs to simple background
About Us Screen	Great and simple.
Analysis Result	Great and user friendly.

6.4.1 CONCLUSION

At the end of the testing, we have reached the idea of the application was easy to get since most users' responses agree upon these criteria. About the usability also, most users agree upon the ease of use, the clarity and navigating the application easily. And we found all the users agree upon the performance and rating result. some users agree upon the speed of load application while the others were neutral.

6.5 TEST CASES

Is a set of conditions or variables under which a tester will determine whether a system under test satisfies requirements or works correctly. The process of developing test cases can also help find problems in the requirements or design of an application [60].

6.5.1 REGISTRATION – VALID ENTRIES TEST

Table6.8: Registration with valid entries test case.

Test case name	Registration - Valid entries Test.
Test description	Test the Registration with valid entries.
Function to be tested	Registration.
Environment	Firebase. IOS device.
Test Execution	<ul style="list-style-type: none">• Open the application.• Press “تسجيل حساب جديد” button.• Fill the required text fields.
Expected Results	New user account created.
Actual Results	Pass, it worked as expected.

6.5.2 REGISTRATION – INVALID ENTRIES TEST

Table6.9: Registration with invalid entries test case.

Test case name	Registration - Invalid entries Test.
Test description	Test the Registration with invalid entries.
Function to be tested	Registration.
Environment	Firebase. IOS device.
Test Execution	<ul style="list-style-type: none">• Open the application.• Press “تسجيل حساب جديد” button.• Fill the required text fields.
Expected Results	Pop up message that ask the user to check entries.
Actual Results	Pass, it worked as expected.

6.5.3 LOGIN – VALID ENTRIES TEST

Table6.10: Login with valid entries test case.

Test case name	Login - valid entries Test.
Test description	Test the login with valid entries.
Function to be tested	Login.
Environment	Firebase. IOS device.
Test Execution	<ul style="list-style-type: none">• Open the application.• Press “تسجيل دخول” button.• Fill the required text fields.
Expected Results	User login to the application with valid entries.
Actual Results	Pass, it's works as expected.

6.5.4 LOGIN – INVALID ENTRIER TEST

Table6. 11: Login with invalid entries test case.

Test case name	Login - invalid entries Test.
Test description	Test the login with invalid entries.
Function to be tested	Login.
Environment	Firebase. IOS device.
Test Execution	<ul style="list-style-type: none"> • Open the application. • Press “تسجيل دخول” button. • Fill the required text fields.
Expected Results	Pop up message that ask the user to check entries.
Actual Results	Pass, it's works as expected.

6.5.5 CUSTOMER LOCATION ACCESSIBILITY TEST

Table6. 12: Customer location accessibility test case.

Test case name	Customer location accessibility Test.
Test description	Test accessibility to user current location.
Function to be tested	Determined customer location.
Environment	Google Maps API. Firebase. IOS device.
Test Execution	<ul style="list-style-type: none"> • Open the application. • Press “تسجيل حساب جديد” button, press “تحطى“ button or press “تسجيل دخول” button. • Pop up message to ask user for allowing access to the current location.
Expected Results	Application access to user current location.
Actual Results	Pass, it's works as expected.

6.5.6 CUSTOMER LOCATION INACCESSIBILITY TEST

Table6.13: Customer location inaccessibility test case.

Test case name	Customer location inaccessibility Test.
Test description	Test inaccessibility to user current location.
Function to be tested	Determined customer location.
Environment	Google Maps API. Firebase. IOS device.
Test Execution	<ul style="list-style-type: none"> • Open the application. • Press “تسجيل حساب جديد” button, press “تخطي تسجيل دخول” button or press “button. • Pop up message to ask user for allowing access to the current location.
Expected Results	Application don't access to user current location.
Actual Results	Pass, it's works as expected.

6.5.7 TRENDING RESTAURANTS AVAILABILITY TEST

Table6.14: Trending restaurant test case.

Test case name	Trending restaurants availability Test.
Test description	Test the availability of trending restaurants.
Function to be tested	View trending restaurants.
Environment	Foursquare API. Firebase. IOS device.
Test Execution	<ul style="list-style-type: none"> • Open the application. • Press “تسجيل حساب جديد” button, press “تخطي تسجيل دخول” button or press “button. • Home screen with trending restaurants will present.
Expected Results	Trending restaurants at the top of Home screen under the title “الأكثر شيوغاً”
Actual Results	Pass, it worked as expected.

6.5.8 RESTAURANTS CATEGORIES AVAILABILITY TEST

Table6.15: Restaurants categories availability test case.

Test case name	Restaurant categories availability Test.
Test description	Test the availability of restaurants category.
Function to be tested	View restaurant category.
Environment	Firebase. IOS device.
Test Execution	<ul style="list-style-type: none"> • Open the application. • Press “ تسجيل حساب جديد ” button, press “ تسجيل دخول ” button or press “ تخطي ” button. • Home screen with restaurants categories will present.
Expected Results	Restaurant categories at the Home screen will present.
Actual Results	Pass, it's works as expected.

6.5.9 RESTAURANTS CATEGORIES SELECT ABILITY TEST

Table6.16: Restaurant categories select ability test case.

Test case name	Restaurant categories select ability Test.
Test description	Test the select ability of restaurant categories.
Function to be tested	Select restaurants category.
Environment	Firebase. IOS device.
Test Execution	<ul style="list-style-type: none"> • Open the application. • Press “ تسجيل حساب جديد ” button, press “ تسجيل دخول ” button or press “ تخطي ” button. • Home screen with restaurants categories will present. • Select specific category.
Expected Results	List of restaurants under the selected category will present.
Actual Results	Pass, it worked as expected.

6.5.10 EMOTION LIST AVAILABILITY TEST

Table6.17: Emotion list availability test case.

Test case name	Emotion list availability Test.
Test description	Test the availability of emotion list.
Function to be tested	View emotion list.
Environment	Firebase. IOS device.
Test Execution	<ul style="list-style-type: none">• Open the application.• Press “تسجيل حساب جديد” button, press “تخطي” button or press “تسجيل دخول” button.• Home screen will appear.• Select specific category.
Expected Results	Emotion list at the top of Restaurant screen will Present.
Actual Results	Pass, it's works as expected.

6.5.11 EMOTION SELECT ABILITY TEST

Table6.18: Emotion select ability test case.

Test case name	Emotion select ability Test.
Test description	Test the select ability of emotion.
Function to be tested	Select emotion list.
Environment	Firebase. IOS device.
Test Execution	<ul style="list-style-type: none">• Open the application.• Press “تسجيل حساب جديد” button, press “تخطي” button or press “تسجيل دخول” button.• Home screen will present.• Select specific category.• Select specific emotion.
Expected Results	List of restaurants under selected emotion will present.
Actual Results	Pass, it's works as expected.

6.5.12 ANALYSIS TEST - EXISTING TWEETS

Table6.19: Analysis with existing tweets test case.

Test case name	Analysis Test - Existing tweets
Test description	Test fetch and preprocess tweets for each restaurant to analyze them.
Function to be tested	Analyze tweets.
Environment	Firebase. IOS device. Twitter API. Flask. PythonAnyWhere.
Test Execution	<ul style="list-style-type: none"> • Open the application. • Press “تسجيل حساب جديد” button, press “تخطي” button or press “ تسجيل دخول” button. • Home screen will present. • Select specific category. • Present analysis result as percentage.
Expected Results	Analysis result for each restaurant will present as percentages at Restaurant screen.
Actual Results	Pass, it's works as expected.

6.5.13 ANALYSIS TEST - NOT EXISTING TWEETS

Table6.20: Analysis with not existing tweets.

Test case name	Analysis Test - Not exciting tweets
Test description	Test fetch tweet if there is no tweet, restaurant will get 50% “عادي” by default
Function to be tested	Analyze tweets.
Environment	Firebase. IOS device. Twitter API. Flask. PythonAnyWhere.
Test Execution	<ul style="list-style-type: none"> • Open the application. • Press “تسجيل حساب جديد” button, press “تخطي” button or press “ تسجيل دخول” button. • Home screen will present. • Select specific category. • Present analysis result as percentage.
Expected Results	Analysis result for each restaurant will present as percentages at Restaurant screen.
Actual Results	Pass, it's works as expected.

6.5.14 SEARCH FEATURE TEST:

Table6.21: Search feature test case.

Test case name	Search feature Test.
Test description	Test the search ability.
Function to be tested	Search for specific restaurant.
Environment	Firebase. IOS device.
Test Execution	<ul style="list-style-type: none"> • Open the application. • Press “تسجيل حساب جديد” button, press “ تسجيل دخول ” button or press “تخطي ” button. • Home screen will present. • Select specific category. • Search bar will present at the top of list of restaurants.
Expected Results	Searched restaurants under the search bar will present.
Actual Results	Pass, it's works as expected.

6.5.15 LIST OF RESTAURANTS AVAILABILITY TEST

Table6.22: List of restaurents availability test case.

Test case name	List of restaurants availability Test.
Test description	Test the availability of list of restaurants in each category.
Function to be tested	View restaurants.
Environment	Foursquare API. Google Maps API Twitter API Firebase. IOS device. Flask. PythonAnywhere.
Test Execution	<ul style="list-style-type: none"> • Open the application. • Press “تسجيل حساب جديد” button, press “ تسجيل دخول ” button or press “تخطي ” button. • Home screen will present. • Select specific category. • List of restaurants will present.
Expected Results	List of restaurants at Restaurant screen will present.
Actual Results	Pass, it's works as expected.

6.5.16 SPECIFIC RESTAURANT SELECT ABILITY TEST

Table6.23: Specific restaurabt select ability test case.

Test case name	Specific restaurant select ability Test.
Test description	Test the select ability of specific restaurant.
Function to be tested	Select specific restaurant.
Environment	Firebase. IOS device.
Test Execution	<ul style="list-style-type: none"> • Open the application. • Press “تسجيل حساب جديد” button, press “تخطي” button or press “تسجيل دخول” button. • Home screen will present. • Select specific category. • Select specific restaurant.
Expected Results	Restaurant information will present.
Actual Results	Pass, it's works as expected.

6.5.17 RESTAURANT INFORMATION AVAILABILITY TEST

Table6.24: Restaurant information availability test case.

Test case name	Restaurant information availability Test
Test description	Test availability of information in each restaurant.
Function to be tested	View restaurant information.
Environment	Foursquare API. Firebase. IOS device.
Test Execution	<ul style="list-style-type: none"> • Open the application. • Press “تسجيل حساب جديد” button, press “تخطي” button or press “تسجيل دخول” button. • Home screen will present. • Select specific category. • Select specific restaurant.
Expected Results	Information at Restaurant information screen will present.
Actual Results	Pass, it's works as expected.

6.5.18 ADD COMMENT FEATURE TEST

Table6.25: Add comment feature test case.

Test case name	Add comment feature Test.
Test description	Test register user ability of adding comment to specific restaurant.
Function to be tested	Add comment.
Environment	Firebase. IOS device.
Test Execution	<ul style="list-style-type: none">• Open the application.• Press “تسجيل حساب جديد” “ تسجيل دخول ” button or press “ تسجيل دخول ” button.• Home screen will present.• Select specific category.• Select specific restaurant.• Press “ كافة التعليقات ” .• Fill the required text field.
Expected Results	Added comment will present at Restaurant information screen.
Actual Results	Pass, it's works as expected.

6.5.19 UPLOAD PICTURE FEATURE TEST

Table6.26: Upload picture feature test case.

Test case name	Upload picture feature Test.
Test description	Test register user ability of uploading picture to specific restaurant.
Function to be tested	Upload Picture.
Environment	Firebase. IOS device.
Test Execution	<ul style="list-style-type: none">• Open the application.• Press “تسجيل حساب جديد” “ تسجيل دخول ” button or press “ تسجيل دخول ” button.• Home screen will appear.• Select specific category.• Select specific restaurant.• press “ إضافة صورة جديدة ” .• Choose image from library or take photo using camera.
Expected Results	Uploaded picture will present at Restaurant information screen.
Actual Results	Pass, it's works as expected.

6.5.20 UPDATE USER INFORMATION WITH VALID ENTRIES TEST

Table6.27: Update user information with valid entries test case.

Test case name	Update user information ability Test - Valid entries Test.
Test description	Test user ability to edit profile with valid entries.
Function to be tested	Edit profile.
Environment	Firebase. IOS device.
Test Execution	<ul style="list-style-type: none"> • Open the application. • Press “تسجيل حساب جديد” button or press “تسجيل دخول” button. • Home screen will present. • From tab bar press (👤) icon. • Fill the required text fields.
Expected Results	Pop up message to inform user that the information updated.
Actual Results	Pass, it's works as expected.

6.5.21 UPDATE USER INFORMATION WITH INVALID ENTRIES TEST

Table6.28: Update user information with invalid entries test case.

Test case name	Update user information ability Test - Invalid entries Test.
Test description	Test user ability to edit profile with invalid entries.
Function to be tested	Edit profile.
Environment	Firebase. IOS device.
Test Execution	<ul style="list-style-type: none"> • Open the application. • Press “تسجيل حساب جديد” button or press “تسجيل دخول” button. • Home screen will present. • From tab bar press (👤) icon. • Fill the required text fields.
Expected Results	Pop up message to inform user to check the information entered.
Actual Results	Pass, it's works as expected.

6.5.22 FAVORITE LIST AVAILABILITY TEST

Table6.29: Favorite list availability test case.

Test case name	Favorite list availability Test.
Test description	Test the availability of favorite list.
Function to be tested	View favorite list.
Environment	Firebase. IOS device.
Test Execution	<ul style="list-style-type: none">• Open the application.• Press “تسجيل حساب جديد” button or press “ تسجيل دخول ” button.• Home screen will present.• From tab bar press () icon.
Expected Results	Favorite list with restaurants will present.
Actual Results	Pass, it's works as expected.

6.5.23 ADD RESTAURANT TO FAVORITE LIST TEST

Table6.30: Add restaurant to favorite list feature test case.

Test case name	Add to favorite list feature Test.
Test description	Test the ability of adding restaurant to favorite list.
Function to be tested	Add restaurant to favorite list.
Environment	Firebase. IOS device.
Test Execution	<ul style="list-style-type: none">• Open the application.• Press “تسجيل حساب جديد” button or press “ تسجيل دخول ” button• Home screen will appear.• Select specific category.• Select specific restaurant.• Press “إعجاب” .
Expected Results	Selected restaurant will be added directly to favorite list.
Actual Results	Pass, it's works as expected.

6.5.24 REMOVE RESTAURANT FROM FAVORITE LIST TEST

Table6.31: Remove restaurant from favorite list feature test case.

Test case name	Remove from favorite list feature Test.
Test description	Test the ability of removing restaurant from favorite list.
Function to be tested	Remove restaurant from favorite list.
Environment	Firebase. IOS device.
Test Execution	<ul style="list-style-type: none">• Open the application.• Press “تسجيل حساب جديد” button or press “تسجيل دخول” button.• Home screen will present.• From tab bar press () icon.• Select specific restaurant.• Press “إزالة الإعجاب”.
Expected Results	Selected restaurant will be removed from favorite list.
Actual Results	Pass, it's works as expected.

6.5.25 LOGOUT ABILITY TEST

Table6.32: Logout ability test case.

Test case name	Logout ability Test
Test description	Test the ability to logout from user account.
Function to be tested	Logout.
Environment	Firebase. IOS device.
Test Execution	<ul style="list-style-type: none">• Open the application.• Press “تسجيل حساب جديد” button or press “تسجيل دخول” button.• Home screen will appear.• From tab bar press () icon.• press “تسجيل خروج”.
Expected Results	User logout from the account.
Actual Results	Pass, it's works as expected.

Chapter 7 | SYSTEM DEMONSTRATION



7.1 SYSTEM SCREEN FLOW

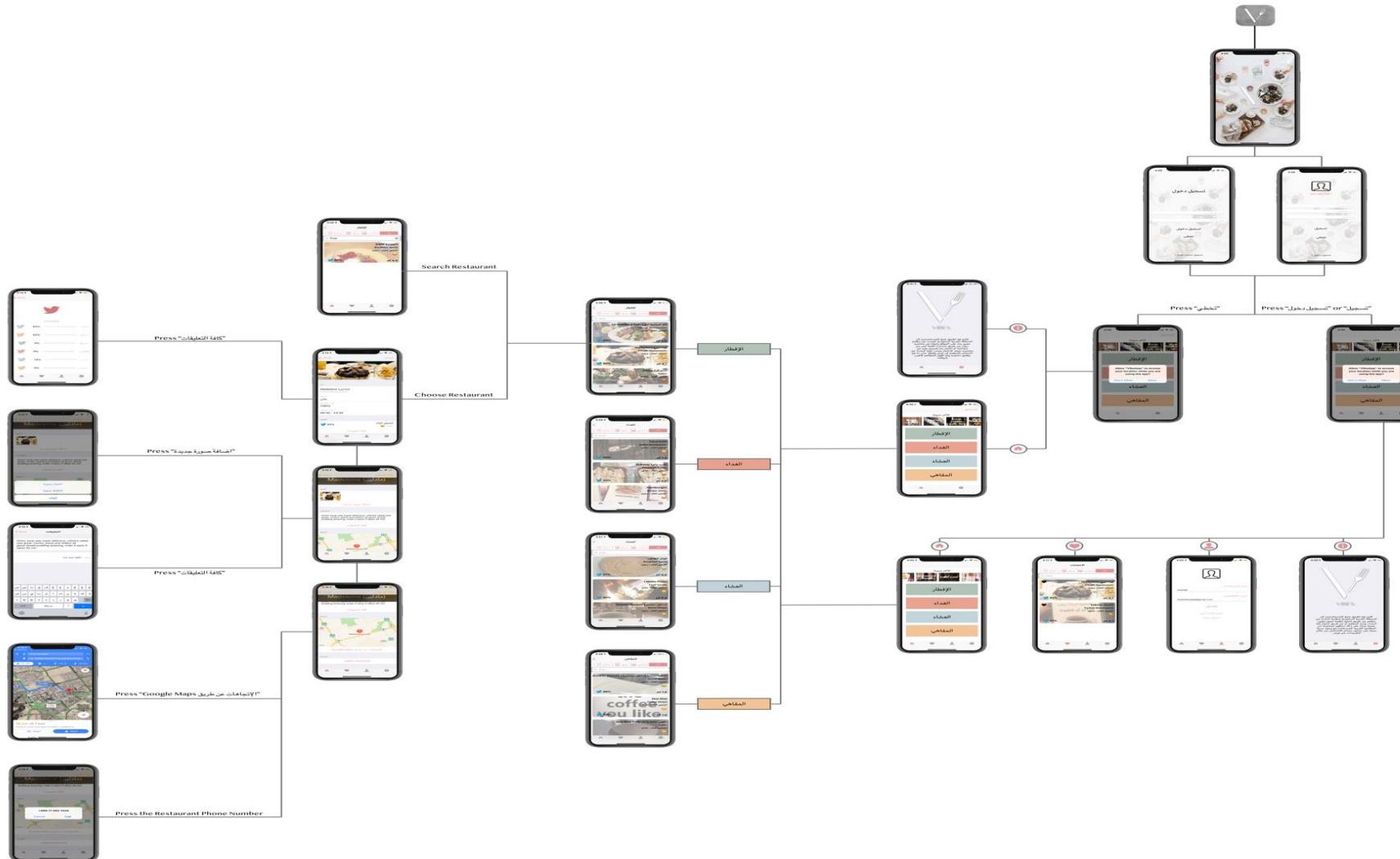


Figure7.1: VIBES site map "system screen flow".

7.2 SYSTEM SCREENS SNAPSHTOS

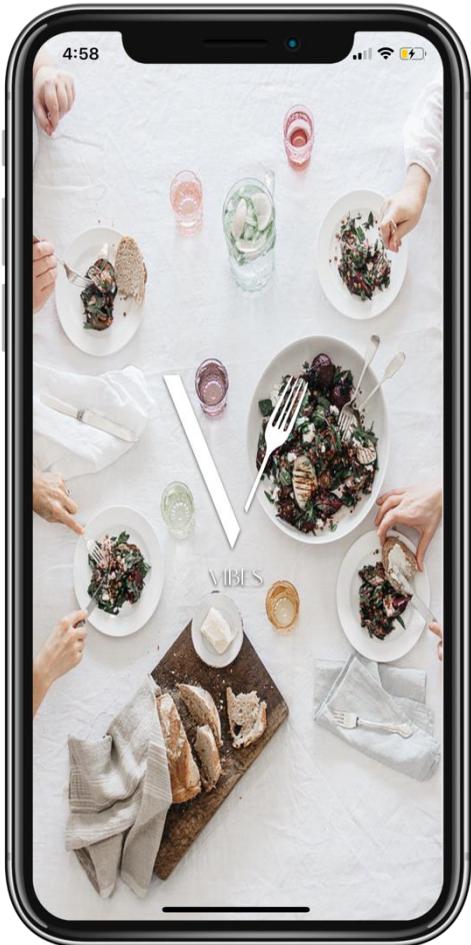


Figure7.2: VIBES splash screen.



Figure7.3: Login screen.



Figure7.4: Sign up screen.

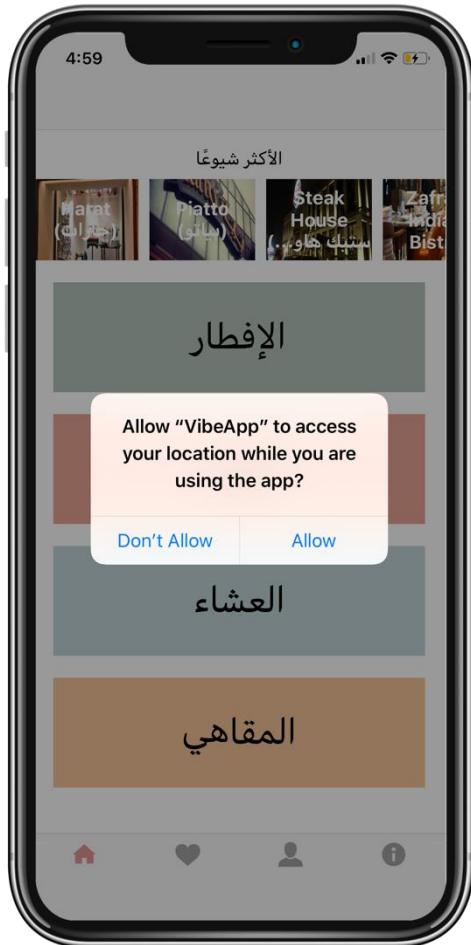


Figure7.5: Determine current location screen.

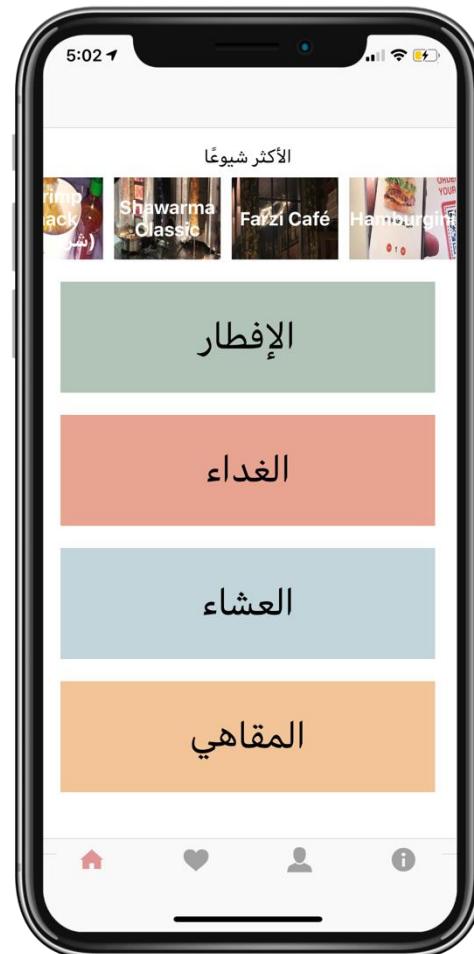


Figure7.6: VIBES home screen.

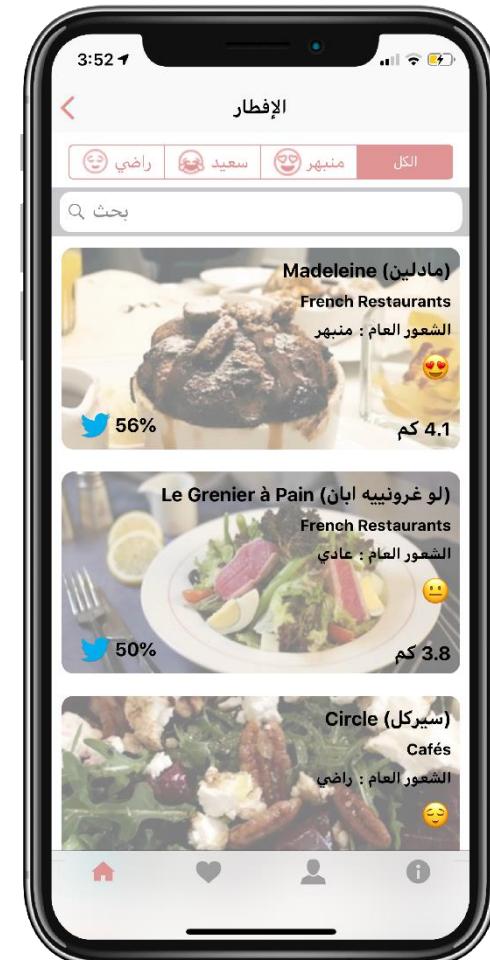


Figure7.7: Restaurants screen.



Figure7.8: Restaurants under "منبهر" emotion.



Figure7.9: Restaurants under "سعید" emotion.

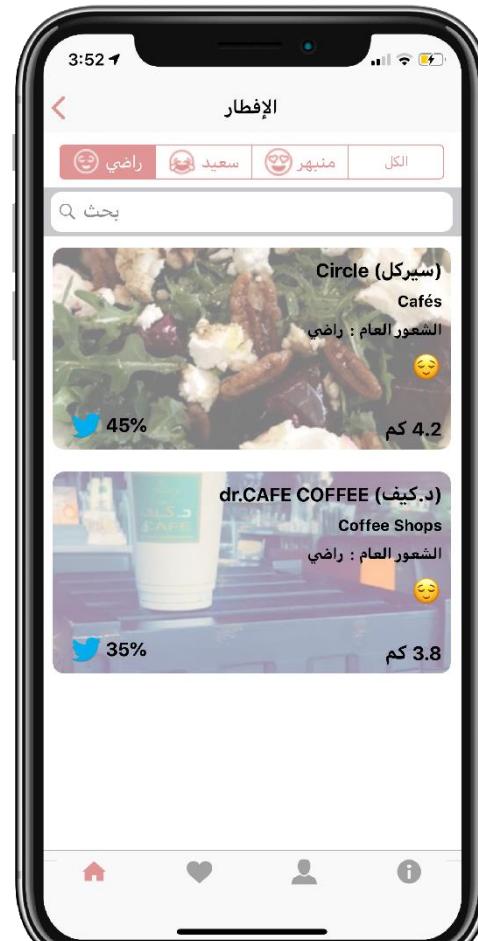


Figure7.10: Restaurants under "راضي" emotion.



Figure7.11: Search bar.



Figure7.12: Restaurant information screen1



Figure7.13: Restaurant information scre



Figure7.14: Restaurant information screen3.



Figure7.15: Emotion analysis screen.



Figure7.16: Favorite list screen.



Figure7.17: Favorite list under "منبهر" emotion.



Figure7.18: Favorite list under "سعید" emotion.



Figure7.19: Favorite list under "راضي" emotion.



Figure7.20: User profile screen.



Figure7.21: Edit profile screen.



Figure7.22: About VIBES screen.

Chapter 8 | CONCLUSION



8.1 SUMMARY

This document describes VIBES application project and its development process in eight chapters. In the first chapter, we have summarized an introduction which defines the problem, goals and objectives, solution, scope and hardware and software tools of the project. The second chapter contains information gathering, where we have reviewed and compared similar applications. Furthermore, we discussed interviews and a questionnaire we conducted in this phase. In the third chapter, we determined the development methodology and the main requirements of the application along with the system analysis models. Fourth chapter shows design of the data models and user interfaces. Fifth chapter present hardware and software that we used and discuss system integration that shows how did we integrate different component with each other, discuss system implementation with present sample code. Sixth chapter shows various types of testing which are unit testing, integration testing, performance testing and user acceptance testing and show some test cases. Last chapter shows system screens flow and snapshots.

8.2 IMPACT OF THE PROJECT

Socially, our project aims to provide Saudis with an application that provides them a guide information of restaurants/cafes in a specific city with an emotion rate that represents the amount of (dis)satisfaction for each restaurant. Commercially, helps restaurant companies improve customer service or other facilities so that will increase competition among them. Locally, provide Saudi people with a guide application that focuses only on Saudi Arabian restaurants. Other impacts can be providing an application that integrates restaurant information from Foursquare. And use Twitter to analyze Arabic tweets then assign an emotion and get the positiveness and negativeness for each restaurant so that help to get benefit from multiple sources to provide an accurate rate and also to save the time for customer.

8.3 LIMITATIONS AND FUTURE WORK

One of the biggest limitations that we faced in VIBES is “how to deal with Twitter APIs “to access some information that we need in sentiment analysis and how to integrate two different languages with each other. In addition, because of choose iOS operating system the development required a new programming language, it was a new challenge to learn it in a short period. Also, we faced some difficulties when we tried to get comments from google map since it's required us to pay huge amounts. In addition, there is no enough resources. According to that we decided to focus on Arabic tweets that collected from Twitter API. Furthermore, we have encountered difficulty with time, as our project confused with other course projects regarding of that and since we have a lot of functions, we weren't having enough time to discover all functions, so we have not applied delete comment and picture features.

Application will be available on Saudi Arabia cities. We may decide to add a new feature such as 3D technology that enable customer to take overall view to restaurant, allow customer to reserve via VIBES. Also, develop VIBES to support “Android “operating system, English language.

8.4 LESSONS LEARNED

During working on project, we gain different skills such as work under pressure skill, time management skill, strong communication skill, new experience in sentiment analysis and working in team effectively.

8.5 ACKNOWLEDGMENTS

Alhamdulillah, Praise be to Allah, first we thank Allah for giving us ability to accomplish the project. Second, we would like to convey our thanks and express the deepest appreciation for those who showed us kindness, respect and willing to offer us help, support and guidance.

We are grateful to our supervisor Dr. Rana, for giving us the empowerment and trust, as well as for sharing her knowledge with us in addition to giving us the opportunity to carry out the project. In addition, we sincerely thank the Committee members, for their interest in the project and reviewing it to give us their feedback.

Finally, it was not possible to accomplish this project without our parent's moral and emotional support that it was helpful in facing difficulties.

REFERENCES

- [1] Osama, G. (2015). The role of social network in shaping public opinion.
- [2] "The Real Business of Restaurants Isn't Food — It's Emotion". [online]. Available at: <https://www.foodabletv.com/blog/2016/10/5/the-real-business-of-restaurants-isnt-food-its-emotion> [Accessed 16 Sep. 2018].
- [3] Yoon, J. and Woody, K. (2011). The Role of Negative Emotions in Explaining Restaurant Customer Dissatisfaction and Behavioral Intention.
- [4] Lee, Y., Back, K. and Kim, J. (2009). Family Restaurant Brand Personality and Its Impact on Customer's Emotion, Satisfaction, and Brand Loyalty.
- [5] "Our Moods, Our Foods. The messy relationship between how we feel and what we eat". March 6, 2014. [Online]. Available: <https://www.theatlantic.com/health/archive/2014/03/our-moods-our-foods/284238/>. [Accessed: 27-Nov-2018].
- [6] Mary, G. (2002). Juggling Food and Feelings: Emotional Balance in the Workplace. United States of America: Lexington Books.
- [7] Raj, V., Ritu, T. and Nirmal, R. (2016). Sentiment Analysis of Social web data: A Review.
- [8] "Why Are iOS Apps Still Better Than Android Apps?". February 15, 2016. [Online]. Available: <https://www.makeuseof.com/tag/ios-apps-still-better-android-apps/>. [Accessed: 27-Nov-2018].
- [9] Mohammed, M., Dagmar, M. and Adrian, P. (2016). Emotion Level Sentiment Analysis: The Affective Opinion Evaluation.
- [10] Apoory, A., Boyi, B., Ilia, V., Owen, R. and Rebecca, R. (2017). Sentiment Analysis of Twitter Data.
- [11] Ben, E., Tim, R., Isabelle, A., Matko, B. and Sebastian, R. (2016). emoji2vec: Learning Emoji Representations from their Description.
- [12] Battiato, S., Gallo, G., Schettino, R. and Stanco, F. (2017). Image Analysis and Processing ICIAP.
- [13] Alexander, P. and Patrick, P. (2017). Twitter as a Corpus for Sentiment Analysis and Opinion Mining.
- [14] Rehab, D., Raed, M., Narmeen, Sh. and Sally, R. (2015). Sentiment Analysis in Arabic Tweets.
- [15] Mashael, S. and Rashid, M. (2018). Sentiment Analysis Model for Arabic Tweets to Detect Users' Opinions about Government Services in Saudi Arabia: Ministry of Education as a case study.
- [16] Islam, R., Muhammad, K., Ashir, A., Abu Raihan, K., Hua, W. and Anwaar, U. (2018). Depression detection from social network data using machine learning techniques.
- [17] Hernández, F., Mora, E. and Viscaíno, H. (2018). When a new technological product launching fails: A multi-method approach of facial recognition and E-WOM sentiment analysis "Physiology & Behavior"
- [18] Samhaa, Beltagy., Talaat, K., Amal, H. and Mohammad, H. S. (2016). Combining Lexical Features and a Supervised Learning Approach for Arabic Sentiment Analysis.
- [19] Rebora, S. and Pianzola, F. (2018). Analysing Comments in the Margins on Wattpad
- [20] "From the mouth of the experts. FOODMOOD App". [Online]. Available: <http://www.foodmood.co.uk/about.html>. [Accessed: 27-Nov-2018].
- [21] "How Foursquare and Other Apps Guess What You Want to Eat. How effective are restaurant recommendation algorithms?" April 24, 2015. [Online]. Available: <https://www.eater.com/2015/4/24/8486279/restaurant-recommendation-apps-foursquare-algorithms>. [Accessed: 27-Nov-2018].

- [22] "ParallelDots AI APIs - Sentiment Analysis". [Online]. Available: <https://www.paralleldots.com/sentiment-analysis>. [Accessed: 27-Nov-2018].
- [23] "Visualizing Tweets Sentiment". [Online]. Available: https://www.csc2.ncsu.edu/faculty/healey/tweet_viz/. [Accessed: 27-Nov-2018].
- [24] "SentimenticApp Account On Twitter". April 24, 2018. [Online]. Available: <https://twitter.com/SentimenticApp>. [Accessed: 27-Nov-2018].
- [25] Greer, D. and Ruhe, G. (2004). Software Release Planning: An Evolutionary and Iterative Approach.
- [26] Akshi, K. and Teeja, S. (2012). Sentiment Analysis on Twitter. IJCSI International Journal of Computer Science Issues.
- [27] Ye, T., Thiago, G., Giulio, D., Emilia, M. and Chao, S. (2017). Facebook Sentiment: Reactions and Emojis.
- [28] Samhaa, B. and Ahmad, A. (2013). Open issues in the sentiment analysis of Arabic social media: A case study
- [29] Kamata, M. I., & Tamai, T. (2007, October). How does requirements quality relate to project success or failure? In *Requirements Engineering Conference, 2007. RE'07. 15th IEEE International* (pp. 69-78). IEEE.
- [30] "System Architecture". April 10, 2015. [Online]. Available: <https://www.mitre.org/publications/systems-engineering-guide/se-lifecycle-building-blocks/system-architecture>. [Accessed: 27-Nov-2018].
- [31] "Three-Tier Application". [Online]. Available: <https://www.techopedia.com/definition/24306/three-tier-application>. [Accessed: 27-Nov-2018].
- [32]. "Swift," Swift Definition. [Online]. Available: <https://techterms.com/definition/swift>. [Accessed: 12-Apr-2019].
- [33]. "What is Python? - Definition from WhatIs.com," WhatIs.com. [Online]. Available: <https://whatis.techtarget.com/definition/Python>. [Accessed: 12-Apr-2019].
- [34]. "Encyclopedia," Xcode Definition from PC Magazine Encyclopedia. [Online]. Available: <https://www.pcmag.com/encyclopedia/term/55875/xcode>. [Accessed: 12-Apr-2019].
- [35]. Geek University, "IDLE editor | Python," Geek University. [Online]. Available: <https://geek-university.com/python/idle-editor/>. [Accessed: 12-Apr-2019].
- [36]. "Firebase Realtime Database | Store and sync data in real time | Firebase," Google. [Online]. Available: <https://firebase.google.com/products/realtime-database/>. [Accessed: 12-Apr-2019].
- [37]. "Documentation | Firebase," Google. [Online]. Available: <https://firebase.google.com/docs/>. [Accessed: 12-Apr-2019].
- [38]. "Firebase Realtime Database | Firebase Realtime Database | Firebase," Google. [Online]. Available: <https://firebase.google.com/docs/database/>. [Accessed: 12-Apr-2019].
- [39]. "FirebaseAuth | Firebase," Google. [Online]. Available: <https://firebase.google.com/docs/reference/android/com/google/firebase/auth/FirebaseAuth>. [Accessed: 12-Apr-2019].
- [40]. "FirebaseStorage | Firebase," Google. [Online]. Available: <https://firebase.google.com/docs/reference/android/com/google/firebase/storage/FirebaseStorage>. [Accessed: 12-Apr-2019].
- [41]. SVProgressHUD, "SVProgressHUD/SVProgressHUD," GitHub, 01-Dec-2018. [Online]. Available: <https://github.com/SVProgressHUD/SVProgressHUD>. [Accessed: 12-Apr-2019].
- [42]. "Overview - Twitter Developers," Twitter. [Online]. Available: <https://developer.twitter.com/en/docs/publisher-tools/twitterkit/overview.html>. [Accessed: 12-Apr-2019].
- [43]. "Core Location," Core Location | Apple Developer Documentation. [Online]. Available: <https://developer.apple.com/documentation/corelocation>. [Accessed: 12-Apr-2019].
- [44]. "Location and Maps Programming Guide," Displaying Maps, 21-Mar-2016. [Online]. Available: <https://developer.apple.com/library/archive/documentation/UserExperience/Conceptual/LocationAwarenessPG/MapKit/MapKit.html>. [Accessed: 12-Apr-2019].
- [45]. "Alamofire Tutorial: Getting Started," raywenderlich.com. [Online]. Available: <https://www.raywenderlich.com/35-alamofire-tutorial-getting-started>. [Accessed: 12-Apr-2019].

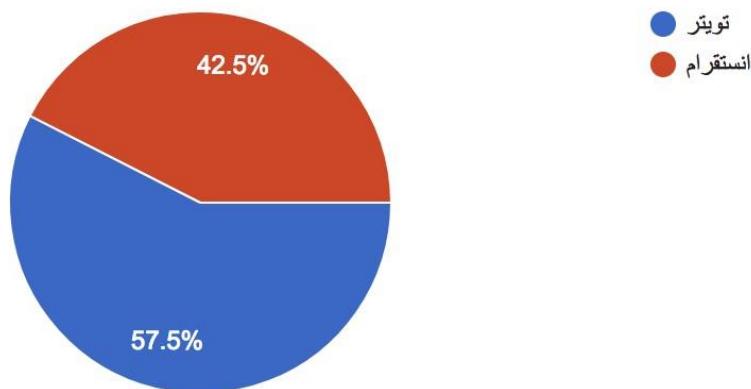
- [46]. Koogawa, “FoursquareAPIClient,” CocoaPods. [Online]. Available: <https://cocoapods.org/pods/FoursquareAPIClient>. [Accessed: 12-Apr-2019].
- [47]. “Concrete proposal for async semantics in Swift,” Gist. [Online]. Available: <https://gist.github.com/lattner/429b9070918248274f25b714dcfc7619>. [Accessed: 12-Apr-2019].
- [48]. “Tweepy,” Tweepy. [Online]. Available: <http://www.tweepy.org/>. [Accessed: 12-Apr-2019].
- [49]. “Stemming and Lemmatization in Python,” DataCamp Community. [Online]. Available: <https://www.datacamp.com/community/tutorials/stemming-lemmatization-python>. [Accessed: 12-Apr-2019].
- [50]. “Introducing JSON,” JSON. [Online]. Available: <https://www.json.org/>. [Accessed: 12-Apr-2019].
- [51]. “PyArabic,” PyPI. [Online]. Available: <https://pypi.org/project/PyArabic/>. [Accessed: 12-Apr-2019].
- [52]. “Firebase Realtime Database | Store and sync data in real time | Firebase,” Google. [Online]. Available: <https://firebase.google.com/products/realtime-database/>. [Accessed: 12-Apr-2019]
- [53] “immediate vs eventual “. [online]. Available: <https://dev.to/barryosull/immediate-vs-eventual-consistency-5cna>. [Accessed: 12-Apr-2019].
- [54]. “System Testing,” Software Testing Fundamentals, 03-Mar-2018. [Online]. Available: <http://softwaretestingfundamentals.com/system-testing/>. [Accessed: 12-Apr-2019].
- [55]. “Unit Testing,” Software Testing Fundamentals, 03-Mar-2018. [Online]. Available: <http://softwaretestingfundamentals.com/unit-testing/>. [Accessed: 12-Apr-2019].
- [56]. “Integration Testing,” Software Testing Fundamentals, 03-Mar-2018. [Online]. Available: <http://softwaretestingfundamentals.com/integration-testing/>. [Accessed: 12-Apr-2019].
- [57]. “Performance Testing Tutorial: What is, Types, Metrics & Example,” Meet Guru99 - Free Training Tutorials & Video for IT Courses. [Online]. Available: <https://www.guru99.com/performance-testing.html>. [Accessed: 12-Apr-2019].
- [58]. “What is performance testing? - Definition from WhatIs.com,” SearchSoftwareQuality. [Online]. Available: <https://searchsoftwarequality.techtarget.com/definition/performance-testing>. [Accessed: 12-Apr-2019].
- [59]. M. Setter, “What is User Acceptance Testing (UAT Testing)? - Usersnap,” Usersnap Blog, 07-Aug-2018. [Online]. Available: <https://usersnap.com/blog/user-acceptance-testing-right/>. [Accessed: 12-Apr-2019]
- [60]. “Test Case,” Software Testing Fundamentals, 15-Jul-2018. [Online]. Available: <http://softwaretestingfundamentals.com/test-case/>. [Accessed: 12-Apr-2019].

APPENDICES

Appendices A – Questionnaire:

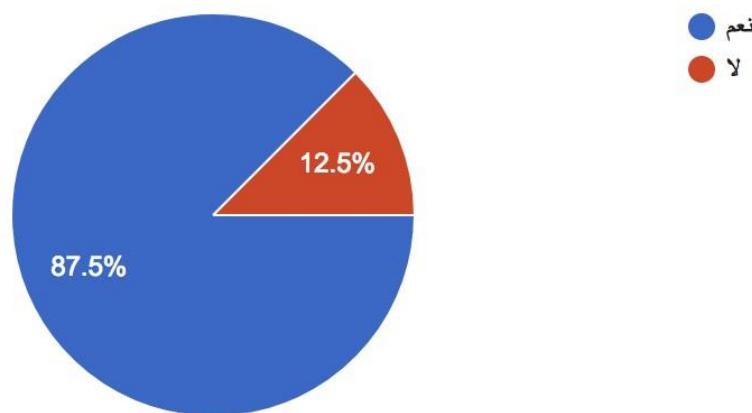
عند تجربة مطعم معين، هل تفضل وضع رأيك عنه؟

رداً 40



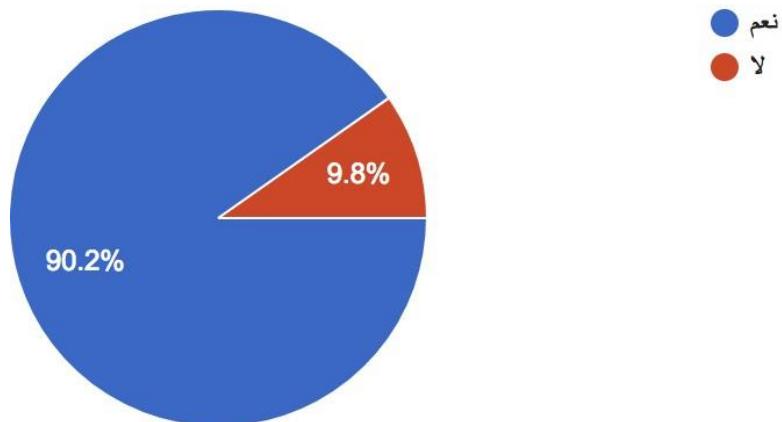
لو ان لديك تطبيق يعرض لك مشاعر السكان لبعض المدن في المملكة العربية السعودية عن المطاعم،
هل من الممكن الإستفادة منه من ناحية اتخاذ القرار؟

رداً 40



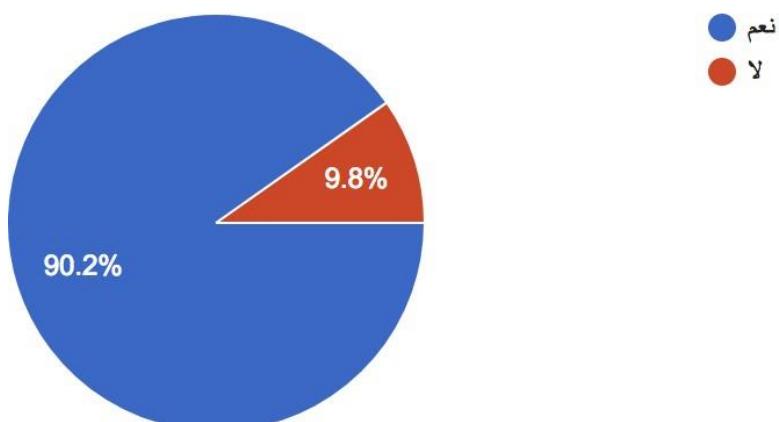
هل تفضل معرفة عواطف وأراء الأشخاص تجاه مطعم معين قبل تجربته؟

ردًا 41



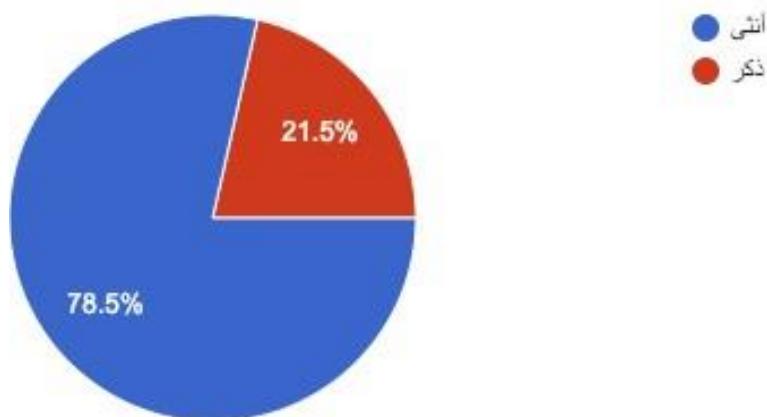
هل يعتبر التويتر وانستقرام ساحة للتعبير عن المشاعر بالنسبة للشعب السعودي؟

ردًا 41



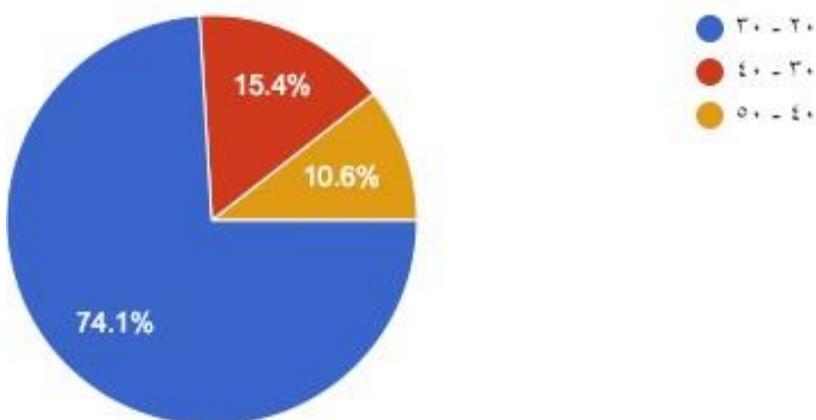
الجنس

نـد 1,002



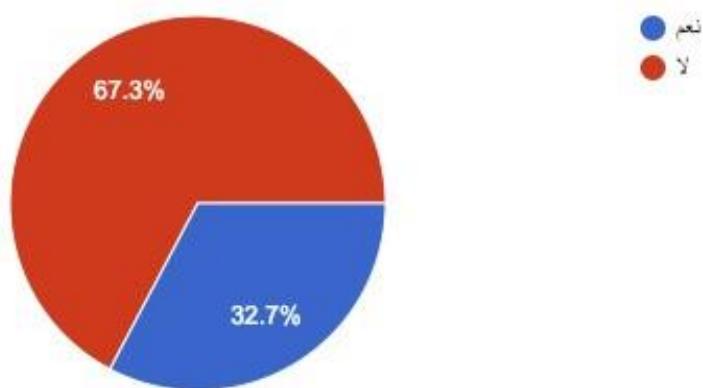
العمر

نـد 1,002



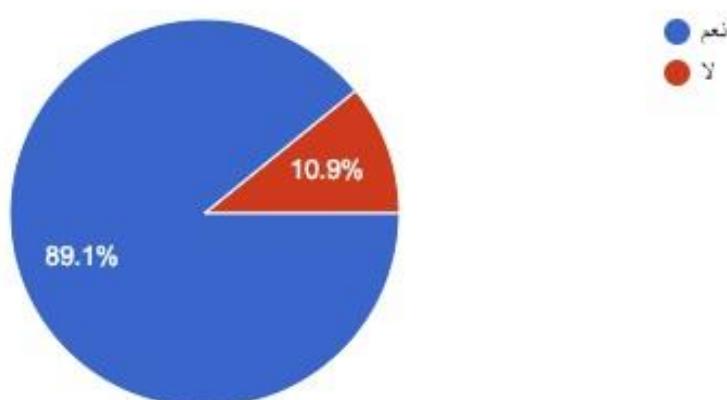
بعد تجربة مطعم هل تفضل ترك إنطباعك ومشاعرك عن المكان في حسابك على التويتر أو الإنستقرام؟

رد 1,002



هل تفضل وجود تطبيق يعرض فقط المطاعم التي تركت لدى زبانتها إنطباع عالي من المشاعر الإيجابية سوا إما بحسن خدمة أو جمال وهدوء مكان أو غير ذلك؟

رد 1,002



اطرح اقتراحاتك ...

رذا 240

يكون هناك حساب لكل مستخدم و يصبح له متابعين ويتابع مستخدمين اخرين ايضا و المستخدم الفعال " يعني الذي يشارك تجربته و يقوم المطعم عند كل زيارة وهكذا" يعطي هذا المستخدم مميزات كـ"شارات مثلاً" انه مستخدم جوال اذا كان يذهب للمطعم باكثر من مكان يعني تكون هناك مهمة و عند تنفيذها يحصل على الشارات ... ويعطى نقاط عند ذهابه لكل مطعم وتختلف هذه النقاط من مطعم الى اخر ... ايضا يمكن ان يكون هناك ميزة للمستخدم الفعال وهي او يكون له مثلاً ١٠٪ تخفيض على فاتورة المطعم

ان يتضمن التطبيق رأي الزبون + ان يظهر تقييم المطعم مثل تقييم الفنادق عدد النجوم + رأي الزبون مدعم بالصور و شكرا لكم على هذه الاستفادة

المعنى لكم التوفيق فكرة البرنامج جداً جميلة الله يوفقكم يا رب

قلة الاسعار والتوصيل مجاناً

حلو ان تحظوا فيه صورة مصغره للمطعم ومثلا توفر طلبات للجلسة يعني مثلاً ابداً مخدات للجلسة بحيث المطعم يوفرها فور طلبها

حيث الفكرة مره واذا تطبقت سيكون شيء كبير ونتمنى لكم التوفيق ❤️

الله يوفقكم لما يحبه ويرضاه والله يسعدكم وتحمرون بترجمكم

مثل اللي قلتكم واتمنى انها تفديكم 😊 وبالتفريق مقدماً للمشروع اللي حالياً تسونوه ❤️

هادي وحلو من جميع التواهي واحبكم

مطعم يحمل طقم عمل سعودي بالكامل قائم على اعادة تدوير بمعنى ان كل ما يحمله المطعم هو اشياء معد تدويرها ✨

COPYRIGHT FORM

Project Title: VIBES

Term of the Project: 2nd Semester

Academic Year: 1439-1440 Higri, 2018-2019 Gregorian

We the team (Rola Al.Dhwayan , Rand Al.Dhwayan , Reem Al.Gumaizi , Shahad Bin.Dakhe , Meznah faleh Al.Rasheed)

Student's Name		Student's University ID Number	Student's Signature
1	<i>Rola saud Al.Dhwayan</i>	436200047	
2	<i>Rand abdulaziz Al.Dhwayan</i>	436200370	
3	<i>Reem saleh Al.Gumaizi</i>	436200324	
4	<i>Shahad majed Bin.Dakhel</i>	436200024	
5	<i>Meznah faleh Al.Rasheed</i>	436201211	

Hereby assign our copyright of this report and of the corresponding executive summary to the Information System (IS) Department of King Saud University. We also hereby agree that the report or demo from our oral presentations is becoming full property of the IS Department. Publication of this report does not constitute approval by King Saud University, the IS Department or its faculty members of the findings or conclusions contained herein. It is published for the exchange and stimulation of ideas.