EXPO 2020: Connecting the World

Author
Hossam GamalEldein M. N. AbdelMeguid
H00263235

Supervisor
Dr. Hind Zantout

A thesis submitted in fulfillment of the requirements for the degree of MSc. IT (Business).

In the

SCHOOL OF MATHEMATICAL AND COMPUTER SCIENCES

August 2018
Declaration

I, Hossam GamalEldein Mohamed Nassef AbdelMeguid
confirm that this work submitted for assessment is my own and is expressed in my own words. Any uses made within it of the works of other authors in any form (e.g., ideas, equations, figures, text, tables, programs) are properly acknowledged at any point of their use. A list of the references employed is included.

Signed: Hossam GamalEldein M. N. AbdelMeguid
Date: 16th August 2018
Abstract

EXPO 2020 – Dubai is a global exhibition for sharing innovations and finding solutions for human challenges (Vicente, 2012). Dubai chose “Connecting Minds, Creating the Future” as central theme to tackle the rising challenges under 3 sub-themes: Sustainability, Mobility and Opportunity. This thesis is concerned with developing a web application that enable users to connect & share ideas together, find interesting events which fill their needs & curiosity, reach organizations to build fruitful partnerships, and the ultimate goal to foster the collaboration between all participants during EXPO 2020 - Dubai. The application is developed in ASP.NET using C# programming language along with Neo4j graph database, in addition to using Extensible Messaging and Presence Protocol (XMPP) to enable instant messages between users. The project is managed using agile methodology to accommodate the changes raised during the project lifetime.

Keywords: EXPO, Event, Graph Database, Instant Messaging
Acknowledgement

I would like to thank my supervisor (Dr. Hind Zantout) for her ongoing support, time and valuable advices throughout the course of this dissertation, without her guidance this thesis would not have been possible.

Special thanks to (Dr. Idris Ibrahim) for his valuable suggestions and advices, are greatly appreciated.

Last but not least, I would like to thank my family for their continuous support and patience during my programme study.
# Contents

Declaration ............................................................................................................................................ i

Abstract .................................................................................................................................................... ii

Acknowledgement ....................................................................................................................................... iii

Contents ........................................................................................................................................................ iv

List of Figures .................................................................................................................................................... viii

Abbreviations ................................................................................................................................................. 1

1 Introduction .................................................................................................................................................... 2
   1.1 Overview ..................................................................................................................................................... 2
   1.2 EXPO: History and Future ....................................................................................................................... 2
   1.3 Aims & Objectives ..................................................................................................................................... 4
   1.4 Report Outline ......................................................................................................................................... 5

2 Literature Review ......................................................................................................................................... 6
   2.1 Overview ..................................................................................................................................................... 6
   2.2 Event Aspects ........................................................................................................................................... 7
   2.3 Event Categories ....................................................................................................................................... 8
   2.4 Existing Products ..................................................................................................................................... 9
   2.4.1 Eventbrite ........................................................................................................................................... 10
   2.4.2 SCHED ............................................................................................................................................... 11
   2.4.3 Comparison of Event Management Applications .................................................................................. 12
   2.5 Technologies .......................................................................................................................................... 14
   2.5.1 Microsoft ASP.NET ............................................................................................................................ 14
   2.5.2 Database ............................................................................................................................................ 16
   2.6 Instant Messaging ................................................................................................................................... 19
   2.6.1 XMPP ................................................................................................................................................... 20
   2.6.2 Openfire Server .................................................................................................................................. 22
   2.7 Summary .................................................................................................................................................. 22
3  Methodology ...................................................................................................................... 23
   3.1  Overview...................................................................................................................... 23
   3.2  Waterfall Methodology ............................................................................................ 23
   3.3  Agile Methodology ..................................................................................................... 24
   3.4  The Project Methodology is AGILE ........................................................................ 25
   3.5  Evaluation Methodology ........................................................................................... 26
       3.5.1  Purpose of Evaluation ....................................................................................... 26
       3.5.2  Evaluation Types .............................................................................................. 26
4  Requirements Analysis ....................................................................................................... 28
   4.1  Overview..................................................................................................................... 28
   4.2  Description of the Application .................................................................................. 28
   4.3  Project Requirements ............................................................................................... 30
       4.3.1  Overview ......................................................................................................... 30
       4.3.2  Functional Requirements ............................................................................... 30
       4.3.3  Non-Functional Requirements ....................................................................... 33
   4.4  Use Cases .................................................................................................................. 33
       4.4.1  Overview ......................................................................................................... 33
       4.4.2  Use Case Index .............................................................................................. 35
       4.4.3  Application Use Cases ................................................................................. 35
   4.5  Graph Data Model .................................................................................................... 46
       4.5.1  Connecting People ......................................................................................... 48
       4.5.2  Connecting Persons with Events ................................................................. 48
       4.5.3  Connecting Persons with Organizations ..................................................... 49
   4.6  Instant Messaging Platform ....................................................................................... 49
   4.7  Project Components .................................................................................................. 50
5  Professional, legal, ethical, and social issues ................................................................... 51
   5.1  Professional Issues ..................................................................................................... 51
   5.2  Legal Issues ............................................................................................................... 51
   5.3  Ethical Issues ............................................................................................................ 52
   5.4  Social Issues ............................................................................................................. 52
6  System Design ................................................................................................................... 53
   6.1  Overview..................................................................................................................... 53
Appendix B ......................................................................................................................................... 91
  1. Screen-Shots.................................................................................................................................. 91
Appendix C ......................................................................................................................................... 98
  1. Code-Snippets............................................................................................................................... 98
Appendix D ......................................................................................................................................... 106
  1. Introductory Questionnaire........................................................................................................... 106
  2. Tasks List...................................................................................................................................... 107
  3. Post-Tasks Questionnaire............................................................................................................ 108
  4. Results......................................................................................................................................... 109
List of Figures

3-Tiers Architecture .................................................................................................................. 15
Entities Graph Pattern ............................................................................................................... 18
Receiving messages over XMPP servers ..................................................................................... 20
Waterfall Methodology ............................................................................................................... 24
Agile Methodology ..................................................................................................................... 25
Top Level Use Case Diagram ..................................................................................................... 34
Register Use Case Diagrams ...................................................................................................... 36
Login Use Case Diagram ............................................................................................................. 37
Browse Events Use Case Diagram ............................................................................................. 38
View Favored Events Use Case Diagram ..................................................................................... 40
Find People Use Case Diagram .................................................................................................. 41
View Followed People Use Case Diagram ................................................................................... 42
Browse Organizations Use Case Diagram .................................................................................... 44
Followed Organizations in Use Case Diagram ............................................................................. 45
Graph Data Model ...................................................................................................................... 47
Class Diagram for Events ........................................................................................................... 54
Class Diagram for Persons .......................................................................................................... 55
Class Diagram for Companies ................................................................. 56
Graph Model of Main Entities ................................................................. 59
ASP.NET Master Page Layout ................................................................. 60
Graph Database ....................................................................................... 64
Event Details ............................................................................................ 66
Saved Event ............................................................................................... 68
Person Details ........................................................................................... 68
Followed Person ....................................................................................... 70
Organization Details ................................................................................ 71
Followed Organization ............................................................................ 71
Activity Log ............................................................................................... 72
Instant Messaging Window ........................................................................ 74
# Abbreviations

<table>
<thead>
<tr>
<th>Abbreviation</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>AJAX</td>
<td>Asynchronous JavaScript And XML</td>
</tr>
<tr>
<td>BIE</td>
<td>Bureau International des Expositions</td>
</tr>
<tr>
<td>BLL</td>
<td>Business Logic Layer</td>
</tr>
<tr>
<td>CLR</td>
<td>Common Language Runtime</td>
</tr>
<tr>
<td>DAL</td>
<td>Data Access Layer</td>
</tr>
<tr>
<td>IM</td>
<td>Instant Messaging</td>
</tr>
<tr>
<td>SDK</td>
<td>Software Development Kit</td>
</tr>
<tr>
<td>SDLC</td>
<td>Software Development Life Cycle</td>
</tr>
<tr>
<td>UI</td>
<td>User Interface</td>
</tr>
<tr>
<td>UX</td>
<td>User Experience</td>
</tr>
<tr>
<td>XMPP</td>
<td>Extensible Messaging and Presence Protocol</td>
</tr>
</tbody>
</table>
1 Introduction

1.1 Overview

No doubt that hosting a tremendous exhibition like EXPO 2020 – Dubai which extends to 6 months with daily numerous events is a tough mission. It requires collaboration and continuous efforts from all involved parties to bring out a successful event (Werner et al, 2016). In addition, managing the visitors of this massive exhibition and organizing the relevant events are motivational challenges in front of the steering committee.

One of the main factors to make a successful event, provide all cutting-edge technologies and facilities needed by the exhibition’s visitors to relish their entire experiment. (Alon, 2017) thinks that Communicating to the right audience and using all-in-one event management software will optimize the efforts.

Providing a tool that assist the visitors to find events, communicate with professional people and reach the registered organizations is one of the essential facilities required throughout EXPO 2020 - Dubai. This thesis introduces a solution that undertake these requirements to implement an application. The project integrates the main elements of EXPO 2020 – Dubai (events, people and organizations) all into one system that meet the participants’ needs. The application will have a bundle of features to enrich the user experience and cover key functions.

1.2 EXPO: History and Future

World Expo is one of the largest exhibitions around the globe that will be hosted in Dubai, United Arab Emirates between October 2020 and April 2021 (Expo2020dubai, 2018). The selection of the city was announced in November 2013 by the member
countries of Bureau International des Expositions (BIE). The honor of selection came after harsh rounds of voting by the nations' representatives for the bidding cities from these countries: Turkey, Russia, UAE and Brazil (Bureau International des Expositions, 2018).

The exhibition aims to introduce new innovations in various fields that help people working in different sectors to showcase their creative achievements (Expo 2020 Dubai Annual Review, 2016). In the past events, EXPO introduced a lot of significant innovations and technologies that have a great impact on human being.

The world-class festive have a distinctive history that started in 1851, in London, UK (Vicente, 2012). The distinguished exhibitions happens every 5 years and extend up to 6 months at most. It gather all the experts and technology specialists in various business domains and industries to communicate and sharing their interests, best practices and challenges of the future.

The previous universal exhibition was hosted by Milan, Italy between May - October 2015 with participations from more than 140 countries (Learn More about Expo Milano, 2015). The host city chose the theme “Feeding the Planet, Energy for Life” to focus on our planet resources and food issues. According to (Sainaghi & Mauri, 2018), The Milan EXPO 2015 attracted more than 20 million visitors. It also had a great impact on the performance of tourism and hotels occupancy.

**EXPO 2020 Dubai: 'Connecting Minds, Creating the Future'**

From the first day that Dubai been selected to organize and host the event, Dubai government form a committee to manage the mega event and all related programs in order to accomplish their ambitious goals.

According to (Expo 2020 Dubai Annual Review, 2016), the world exhibition expected to host participants from 180 countries and be visited by 25 million people. Under the theme 'Connecting Minds, Creating the Future', Dubai work to host the mega event and establish a strong platform where visitors and participants gather to innovate, learn, enjoy and sharing thoughts around their common goals. They engage all people who have talents together to provide solutions that serve the global community. EXPO 2020
– Dubai will provide the opportunity to find industry’s professionals and experts from all countries and business sectors all in one place.

EXPO 2020 Dubai will open 24 hours per day, 7 days a week through the period of the exhibition that will be extended for 6 months. It is expected to have 300,000 visitors per day (Expo 2020 Dubai Annual Review, 2016). It will be a real challenge to manage and connect this big number of visitors.

In order to engage people with the events conducted throughout the exhibition, there should be an application that use the information collected from all participants, and utilize it to create a kind of relational network of the planned events, professional people and registered organizations.

### 1.3 Aims & Objectives

This project aims to provide a solution for the visitors, participants and all organizations’ representatives of EXPO 2020 - Dubai who share the same interests or common thoughts in order to get connected and communicate together. It will enable users to find interesting events, reaching professional people and communicating with esteemed organizations.

This project undertakes the goals of EXPO 2020 – Dubai as leading motivations to build an application that combines the key elements of the exhibition and integrate into All-in-one application. The intended solution consists of a network of Events, Persons and Organizations which are integrated together to foster the connectivity between people and boost collaboration between them. The project will be developed as a web-based application that can be accessed through the web browsers, and could be developed also for mobile platforms in the future.

As the number of events and activities to be conducted during the exhibition period are expected to be enormous, it will be difficult for visitors to determine the suitable events and pick out the useful ones to attend. The proposed application will classify the events based on EXPO 2020 - Dubai sub-themes (Opportunity, Mobility and Sustainability).
The proposed solution will collect the user’s feedback on events, then suggest them accordingly some events that would be of their interest. It will also recommend events based on their friends or friends of friends’ interests. The project targets to replace the traditional methods used to collect feedback either through a form or electronic surveys, and alternatively embed it into the application.

In addition, the project will enable users to find people working in various industries, build connections with professional persons, and facilitate communications between them. It will utilize the instant messaging technology to empower the application with the tool needed to get people connected.

Finally, this project will also enable users to browse registered organizations and reaching their decision makers. Users could add any organization to their favorite list or even connect directly with any of their representatives.

1.4 Report Outline

The first chapter of this thesis includes an introduction to the topic, identifying the problem, and the main objectives of the project. The second chapter will explore the event aspects and categories, and compare the proposed solution against two of the existing products. It will go through the technologies and features which will be used in the application. The third chapter will identify the methodology which will be used to manage and evaluate the project results. The fourth chapter will analyze the system requirements. The fifth chapter will explore the type of issues that could be encountered or even resulted from the project.

The sixth chapter is concerned with the system design aspects, then followed by the system development and implementation which is explained in chapter seven. The system evaluation results will be discussed and analyzed in chapter eight. The ninth chapter will contain a conclusion of the entire project and the future work.
2 Literature Review

2.1 Overview

Guests of exhibition often need to know the agenda at least four months prior their visit to determine the events which coincide with their interests, or that may match with their business needs (billetto, 2018). Although exhibition organizers usually announce the agenda of the entire event earlier, however some people may not have the chance to look at the activities. With the expected number of events that will be carried out during EXPO 2020 - Dubai, it is essential to have an application that organize the events and connect visitors to the activities that would be of their interest. It will be an overwhelming effort for visitors to find their preferred events in the hustle of exhibitions.

The rapid changes of technology requires from the organizers of exhibitions to adapt their communication with the visitors, and utilize technologies that give their visitors a memorable experience (Sandra & Nataša, 2014). Providing a platform for the enthusiastic visitors in which they can view all events, find ones related to their domain, and listing the top rated events by their fingertip all are worthwhile features for thrilling their experience.

Prioritizing events based on the feedback received from the visitors, showing events distinguished by EXPO 2020 sub-themes (Opportunity, Mobility and Sustainability), and enabling users to search events by the level of connection they have with other persons all are considerable diversity in the features of the proposed application that would be needed by the visitors of EXPO 2020 - Dubai.
2.2 Event Aspects

Event is a blend of activities that aim to achieve the goals of the event and meet customer needs (Ian Yeoman et al, 2004). EXPO 2020 - Dubai is a mega exhibition that have numerous events will be hold during the period of exhibition extended to 6 months. Each event should achieve the goals of the organizer, however all the events should complement each other to attain the ultimate objectives beyond the entire exhibition.

In order to come up with a successful event and make visitors satisfied, (Ian Yeoman et al, 2004) list some considerations that should be taken into account when organizing an event:

- **Size and Capacity:** the reserved spaces should comfortably accommodate the needs of visitors, exhibitors, equipment and all facilities that would underpin the success of the event.
- **Facilities:** include all services and tools that are vital to accomplish the work. It is wide to include technical work, equipment, communications, power, food, devices and safety factors.
- **Visibility:** a crucial element of the success of any event to reach the target audience and attract them through various media campaigns and all suitable communication channels. Marketing campaigns may be associated with a financial constraints.
- **History:** can be a major factor on the success or failure of any event. The history of a place or past events may influence negatively or positively on the visitors. The reputation of particular event could control the level of trust and expectations of people who plan to attend.
- **Timing:** choosing a suitable timing to organize the event play a vital role. An event should be scheduled at a suitable season with no overlap with other similar occasions.
• Ticket cost: the cost of ticket should be carefully studied before offering to the customers. The price of ticket control the decision of people whether to visit the venue or not.

2.3 Event Categories

EXPO 2020 Dubai will attract millions of enthusiastic visitors to explore the venues and participate in events that would be of their interest or may fill part of their curiosity. The tremendous exhibition will host a lot of events and activities in various disciplines. EXPO 2020 Dubai events will come under three main themes (Opportunity, Mobility and sustainability) which reflect most of our current lifetime issues.

According to (Julia, 2011) as cited by (Peter & Pascal, 2013) events are classified into the below categories:

• Business & Corporate Events: is an event that foster the organization performance by supporting the strategic objectives. It includes all events that could impact on the corporate internal and external processes such as marketing, customer relations, communications and employee relations.
• Cause-Related & Fundraising Events: is an event that aims to receive money or donations for charity or revenue purposes.
• Exhibitions, Expositions & Fairs: is an event that facilitate and connect the buyers & sellers of products or services. The event may be dedicated for particular industry or general.
• Entertainment & Leisure Events: these kind of events are organized for entertainment purposes. It may be one-time or periodic basis, with tickets or free. For example: concerts, live shows.
• Festivals: it is a cultural event that may be celebrated & organized by the community for religion, traditional or cultural origins. For example: national day, agriculture.
• Government & civic Events: is an event created for a particular entity or political party to boost the reputation and public outlook on the organizer.
• Marketing Events: any event that bring all sellers & buyers in one place in order to promote particular products or services. It aims to increase the awareness of people around the offered product/service to increase revenues or market proportion.

• Conferences, Workshops, Meetings, and Convention Events: people meet to share information around specific or multi topics. It aims to make a discussion, debate, consensus, networking and exchanging knowledge on a particular subject.

• Social/Life-Cycle Events: according to (Timo Reuter et al, 2013) social events are defined as “events that are organized and attended by people and are illustrated by social media content created by people”. It is a private occasion for religious, cultural or social purpose. Usually it is a repetitive events for sharing moments with others.

• Sports Events: these events usually have a large number of participants to spectate a competitive sporting event.

It is obvious that many of the mentioned categories could describe the class of EXPO 2020 – Dubai. Although EXPO 2020 – Dubai is identified as a global unique exhibition that aims to share innovation and foster cooperation (Bureau International des Expositions, 2018), it is extensively broad to be categorized under all the following types: (Business & Corporate Events), (Exhibitions, Expositions & Fairs, Entertainment & Leisure Events), (Festivals), (Marketing Events) and (Conferences, Workshops, Meetings and Convention Events).

2.4 Existing Products

There are numerous web applications available online that enable organizers to create accounts and setup events. Most of these websites are commercial-oriented which aims to provide the customers with different packages that suit their needs.

The existing websites usually target users within a specific country, city or area. They promote their events through various advertisement channels to reach and attract them
to join the event. According to (Katie, 2018), People use discovery sites like neighborhood guides, newspaper and social media posts to find out events and activities. Many Events websites are providing a complete event management system with most of aspects, starting from creating an event, ticketing, promoting and incorporate all the event details.

This report will focus on 2 of the top leading event management service providers, where they acquire a big proportion of the market and have a good reputation and gain their customers trust.

Although the following sample websites are more generalized and could suit all types of events, the intended application will be more concentrated on a single mega exhibition which have enormous embedded events that complement the strategic objectives of the exhibition (EXPO 2020 - Dubai).

### 2.4.1 Eventbrite

Eventbrite is a complete event management application to create, design, manage and analyze all the event lifecycle. The website is designed to manage all types of event starting from a dinner to largest festival. They run over 2 million events every year (Eventbrite, 2018). They have a wide range of tools that enrich the experience of both the organizer and the event attendees.

The application provides easy access to user account from any device. They provide set of options for the event creator to facilitate all event aspects and make customers focus on their core business. They provide a dash board to manage the event, and marketing tools to help organizers reach their audience on social networks. They are integrating with many promotion sites (event distribution) to sell their tickets. According to (Katie, 2018), selling tickets directly on Facebook drives sales to the double.

The website offer many packages for the organizers to choose the suitable one as per own business needs. They also offer free account for no ticketing events. They have an analytical tool to manage the sales and marketing campaigns (Eventbrite, 2018). They provide widgets to integrate the event on customers’ websites through embedding HTML code into the pages.
Eventbrite website offer various coloring themes & designs options for the organizer to choose the suitable appearance. With multiple subscription packages, organizers can choose the features that fit their business needs. They offering many payments methods for both the organizer and attendees (Eventbrite, 2018). They facilitate the purchasing process and transfer the collected money to the organizer as per the chosen plan. They avail many payment options for organizers & attendees: Eventbrite Payment Processing, PayPal, and Authorize.Net.

Briefly, although this website is comprehensive event management solution with most of features that event organizers might be looking for, but it is not suitable for a particular mega exhibition like EXPO 2020 Dubai with many expected events to be held every day across the hosting period which extends to 6 months. The needs of building correlation between events, giving high level of user control, and engaging attendees efficiently with the activities all lead to have a new customized solution that can handle the enormous events, connect people together and solicit feedback from visitors all in one reliable application.

2.4.2 SCHED

SCHED is a distinguished event management website with superior features that meet the requirements of any event scale. They offer the facilities of synchronizing the event details with all of the personal calendars, social networks, digital signage and paper guides (SCHED, 2018). They provide the attendees with a profile page to ease the communications between them and find each other. They can customize a mobile app for the event to let all attendees updated.

They provide the event’s speakers more features to manage the sessions with full control of the session details. They facilitate adding & editing the slides on the speaker profile. With easy access to the attendees list, speakers can communicate with the attendees by email to keep them up to date. Using the outstanding email tools, Organizers can reach the attendees during the event, invite them to participate and update the attendees with any coming up announcements.
One of the main features they provide to their customers is branding the event page as per the customer's website colors in order to give the same look & feel. They empower the provided service with many kinds of reports (e.g. Capacity report, session report, Feedback report) to enrich the organizer experience and facilitate managing the audience efficiently (SCHED, 2018).

They have various pricing plans to meet the organizer business requirements. The offered packages are varying the features based on the customer budget. Some of the advantages they provide: printing the attendees list and schedule, In addition to native mobile Applications for iOS & android.

Overall, the website is suitable for managing a particular event with single-time attendance. It is not specified to manage huge exhibition like EXPO 2020 Dubai with its numerous daily events.

### 2.4.3 Comparison of Event Management Applications

As we have just explored the existing products, it is obvious that websites are great for organizers when they have an event running within limited time and specific activities. They have powerful features that cover most of organizer needs starting from a tiny party or meeting to big festivals.

However these kind of event management systems are not designed to handle massive exhibition extended to six months with many embedded events, conferences and activities that will run every day in several disciplines. Table (2.1) illustrates the key differences between the proposed solution and the existing products.

<table>
<thead>
<tr>
<th>Features</th>
<th>Applications</th>
<th>EXPO 2020 - Dubai</th>
<th>Eventbrite</th>
<th>SCHED</th>
</tr>
</thead>
<tbody>
<tr>
<td>Accessibility</td>
<td>• Web-based application</td>
<td>• Web-based Application</td>
<td>• Web-based Application</td>
<td>• Web-based Application</td>
</tr>
<tr>
<td></td>
<td>• Mobile Apps</td>
<td></td>
<td>• Mobile Apps</td>
<td>• Mobile Apps</td>
</tr>
<tr>
<td>Event Suggestions</td>
<td>• Recommend Events based on user Connections</td>
<td>• General Recommendations</td>
<td></td>
<td>• General Recommendations</td>
</tr>
</tbody>
</table>
As demonstrated in the comparison table, the existing products are not proposing a solution to solicit feedback directly from visitors at the venues of EXPO 2020 Dubai. There is no method to correlate and manage the events rate and feedback from the visitors in order to improve the performance during the exhibition. While the existing products are not providing a connection between the expected events, it will be difficult to recommend the visitor more of the activities, events or conferences that would be of their interests. The existing applications are not providing a communication channel or a way to make the participants of a particular event to get connected.
2.5 Technologies

In order to resolve the underlying problems and fill the gap between the existing products and the raised requirements for EXPO 2020 - Dubai, this section explains the technologies that will be used for implementing the project in order to address the identified issues. It goes through the key features of each technology intended to be used, and explaining why it is suitable for this application.

2.5.1 Microsoft ASP.NET

ASP.NET is a Microsoft programming language used to develop a variety of applications (Matthew, 2012). It is web platform & toolkit based on .NET framework to develop web applications and services. Developers can write the code through by in many programming languages like VB.NET or C#.

There are several templates available on Visual Studio IDE to choose the one suit our need, for example: ASP.NET web forms site, Windows Forms Application, Class Library, etc.

ASP.NET is a server-side programming language, all code are executed at the server side, then render the results at the client’s web browser in HTML format.

According to (Nash, 2010), Common Language Runtime (CLR) is the engine that run the code at the server. CLR is part of .NET framework architecture to execute the intermediate language (compiled code) on the fly using Just In Time (JIT) compiling, In addition to providing some services based like memory management and security.

ASP.NET have 3 frameworks used for developing web applications (ASP.NET Overview, 2010). The frameworks are: web forms, MVC and web pages. Each of them has their own development style, the developer choose the approach based on the application type, the level of experience and development skills.
A standard web application has layers architecture to manage the complexity of the code and placing application code into organized layers based on the functionalities & responsibilities. It helps to isolate each layer and make it more manageable and easier to change with the limited impact. The typical application layers are: User Interface Layer (UI), Business Logic Layer (BLL) and Data Access Layer (DAL). Figure (2.1) illustrates the 3-tiers architecture:

![3-Tiers Architecture](image)

The user send the request to User Interface (UI) layer, then the request is forwarded to Business Logic Layer (BLL) for processing where the logic of application exists. In this context, there is no direct interaction between UI layer and Data Access Layer (DAL), all requests only go through BLL to control the application flow and check the necessity of accessing database via DAL or not.
2.5.2 Database

2.5.2.1 NoSQL Databases

NoSQL is a new generation of databases to deal with the rapid demand of huge data sets and accommodate the different aspects of data models. It is designed to handle the scalability issues of relational database that affect the data management performance. (Celko, 2013) says that traditional RDBMS model is not suitable to handle data placed on multiple machines or distributed across many networks. It is not adequate to process big data with unstructured format.

NoSQL originally created to process the massive data stored on several distributed servers or in the cloud and accessed by millions of users. NoSQL is fostered with several features to address the downsides of traditional databases, According to (Roopendra, 2017) the key benefits of NoSQL are:

- **Schema agnostic:** is one of the main advantage over the relational databases, database has no schema, no relations required for data objects which make it more flexible to handle structured and unstructured data.
- **Scalability:** load on databases are increasing fast, distributing databases across several distributed hosts using scale out technique shows more performance efficiency and cost effective approach than scaling up which basically require manual sharding and adding more hardware with expensive cost.
- **Performance:** Adding more servers for NoSQL databases optimize the performance, and get benefit of the automatic sharding feature which consider an overhead cost in the relational databases.
- **High availability:** NoSQL distribute data equally across all nodes which eliminate the complexity of relational database architecture which consist of a primary and secondary nodes.
- **Global availability:** the distributed data architecture make NoSQL more reliable, minimize the latency and enhance the performance. Users can access data from anywhere across the automated replicated servers or through clouding with the lower level of administration.
According to (Sullivan, 2015), NoSQL has four widely used database types:

1- Key-Value pair database: Data stored within a big hash table in key-value format.
2- Document database: Flexible data structures organized in a set of collections.
3- Column family store database: Cells of data within columns that are grouped in column families.
4- Graph database: Storing data in entities (nodes) linked by relationships (edges).

As this Project adopts building connections between the key entities (Persons, Events and Organizations), so the database which mostly suits the requirements is graph database. The next section explains the graph database, and demonstrates how it will be utilized to present the graph data model.

2.5.2.2 Graph Database

“A graph database management system is a database management system with Create, Read, Update, and Delete (CRUD) methods that expose a graph data model” (Ian, Jim & Emil, 2015).

According to (Chris, 2015), Graph database consists of nodes, each node store information about particular entity, each node has properties that define the characteristics of that entity. Edges are used to connect the nodes to shape out the graph model. Edges could also have some properties which used to correlate the data of the nodes.

Graph database is a suitable choice for any system that has a hierarchical data or interconnected relationship architecture. It is useful for applications which use graph-like queries (Peter et al, 2013). For some use cases where data patterns are correlated, putting data in a specific order within a graph model improve the performance of data retrieval and reduce the latency in comparison with transactional queries.

Performance is a significant factor for graph databases, the mechanism used to deal with the connected data is sophisticated than the one used in relational database where dataset is built upon a set of joint-queries and performance get degraded as the data size grow up. In contradiction, creating dataset in a graph database is built on a
projection of a specific part of the graph, which reduce the execution time of any query by focusing on a portion of the graph only.

Graph database is very flexible to adapt with any changes emerged, it can accommodate the new requirements of adding a new relation, new nodes or any subgraph without affecting the entire application. With no schematic architecture, graph database is more agile to evolve with the changes raised throughout the application development lifecycle.

With all the mentioned powerful features, graph database have much more advantages that make it a suitable choice for this project. With the benefits of connecting data through a relational map, graph database will provide a proper solution to connect the events that will be carried out through EXPO 2020 Dubai exhibition, and make it the ideal database to use in this project.

Figure 2.2: Entities Graph Pattern

Figure (2.2) illustrates how the graph database will be used to connect the main entities of the system. The entities (nodes) are linked through connections (edges) which describe the type of relationships between nodes. Each node has a set of attributes in key-value pairs. Each relationship (Edge) might have a direction or attributes presenting the relationship properties (Chris, 2015).
Neo4j is one of the graph database management systems with native graph processing and storage (Robinson I. et al., 2015). Neo4j is empowered with rich features that support rapid traversals for graph algorithms.

This project will use Neo4j with one of the most widely used graph query language called Cypher. Cypher is used to retrieve data from graph database matching a specific pattern. The below cypher query illustrate how to return data of all events for a particular person:

```
match (P:PERSON{user_id:"hossam"})-[R:Interest_in]-(E:EVENT) return P,E
```

### 2.6 Instant Messaging

Instant Messaging (IM) is defined as a method of communication between two or more persons in real time. IM is used in many areas where immediate contact is required between persons or groups. (Edney & Maximo, 2007) stated that 400 million instant messages are exchanged every day. Because of the features and collaboration functionalities provided by IM, it has become increasingly popular and used by 75% of all teens in US (Amanda, 2012).

The messages usually are exchanged through a client software or over the web using non-standard protocols. The protocols which are used to transfer the text contents between peers also capable of transferring files, images, voice and live videos. A communication protocol is defined as behavior convention which specify the format of messages and order of interactions between the peer entities (König & Këonig, 2012).

Although the great benefits that could be achieved from using IM in organizations, (Joepen, 2005) thinks Instant Messaging could cause severe security risks for some firms which can’t control their data flow. The protocols used to transfer messages could carry scripts, macros or attachments that may be infected with viruses. So using a secured messaging server to keep all messages safely, and choosing an appropriate messaging protocol are substantial before starting this project.
2.6.1 XMPP

Extensible Messaging and Presence Protocol (XMPP) is one of the competing instant messaging standards. It is based on Jabber open-source community, and was adapted by Internet Engineering Task Force (IETF) in 2002 to provide the functionalities of instance messaging and presence needed to build IM applications (Jennings et al., 2006). Some examples of the XMPP applications are Whatsapp and Google talk.

XMPP is basically streaming XML over the network, it was designed to introduce a decentralized alternative to services like MSN and ICQ (XMPP, 2018). Some of the core technologies of XMPP include the presence which provides information about network availability, strong authentication and contact lists.

Unlike other IM standards which require many unnecessary requests to introduce the messages, XMPP is designed with real-time pushing mechanism to send all messages instantly between clients efficiently (XMPP, 2018).

XMPP is flexible to enable any user registered on XMPP server to communicate with another XMPP server (Sharma, 2018). The client usually receives the message directly from the server if both are connected to the same XMPP server, however if the recipient is connected to another XMPP server, the message is sent to the recipient’s XMPP server, then the message is forwarded to the receiver client (Bazara & Fatma, 2010). Figure (2.3) demonstrates how clients are receiving messages over XMPP servers.

Figure 2.3: Receiving messages over XMPP servers - (Bazara & Fatma, 2010)
XMPP is completely federated and supports high availability. So if one user is connected through a particular XMPP server and that server get failed, all users who are connected on other XMPP servers will retain reachable and connected (Marcel, 2016). Moreover, All XMPP clients support the key features of instant messaging, so users can choose the client program they prefer to connect with XMPP server.

XMPP will be utilized in this project to connect people and facilitate the communications between them during EXPO 2020- Dubai. It will enable the users to reach people they are following, foster their connectivity and promote the cooperation between all participants. It will support the ultimate objective of EXPO 2020 which aims to connect minds and elevate innovation between people.

This project will use XMPP technology to enable users communicating from the application over the web using a chatting window. User could send instant messages only to the persons they are following and already added to their favorite list. Likewise, users could initiate textual conversation with a friend by opening the messaging window of that friend.

Each user must have a unique Jabber ID which identify the user address on XMPP server. Jabber Id is required to present the availability status of users on the server and setup the communication channel between peers. The application will use the same user ID created by each user at the registration to be identical with their own Jabber Id.

This project will also utilize the presence feature of XMPP to give users the control to show their availability status for their peers, besides controlling the visibility state for a particular user.

This project will use agsXMPP library to integrate the web application with the functionalities of XMPP server. agsXMPP SDK is open-source library for XMPP client and server development. It is fast and lightweight tool which is dedicated to Microsoft .NET framework (agsXMPP SDK, 2018). SDK stands for Software Development Kit which contains a set of libraries and tools for developing applications on certain platforms.
2.6.2 Openfire Server

XMPP server plays a significant role to provide the features required by client programs. Openfire is an open-source XMPP server that provide messaging and presence features. It is compatible with many platforms like Linux, macOS, Solaris and Windows (XMPP, 2018). Openfire has an easy to use web-based interface. By default, it has embedded database to store user details and messages. However it supports any external databases like MySQL or oracle (Openfire, 2018).

This project will use Openfire server to handle the instant messages between users and manage the presence status of clients over XMPP. Openfire will store Jabber IDs of all registered users, and validate their credentials to authenticate them on the server.

2.7 Summary

EXPO 2020 - Dubai is a mega exhibition with numerous events running daily for 6 months. Millions of visitors are expected to participate in this world festival. Managing these events efficiently, connecting all participants and reaching people with the registered organizations all require placing a proper strategy to attain them successfully.

This project is concerned with building a web application to connect people together during EXPO 2020 – Dubai, collect feedback from the visitors and recommend events accordingly. The proposed application aims to collect the visitor’s feedback constantly, evaluate the satisfaction of participants in order to get an understanding about the visitors experience and enhance the quality of the provided services. Recommending events to the visitors based on the collected feedback will enrich the visitor experience and contribute to the exhibition success.

There are several events management tools available online with wide variations in the features they provide to their customers. Most of them are appropriate to the events which run for a short limit period, but they are not suitable solutions for a mega exhibition like EXPO 2020 - Dubai. Developing a new application to fill the gap is the main objective of this project.
3 Methodology

3.1 Overview

No doubt that a successful project requires a proper project management plan. One of the key factors to develop a new software efficiently is choosing the appropriate software development methodology that meets the requirements of the project.

There are several methodologies used to control and manage the entire software development lifecycle. This chapter explores two of the common methodologies used for developing applications. It goes through the key features of each methodology, highlighting some of the key principles and offered advantages, and then select one methodology that mostly suits the project requirements & conditions.

This chapter also discuss the evaluation methodology used throughout the project progress, In addition to the user evaluation approaches used after the software development completion.

3.2 Waterfall Methodology

Waterfall methodology is a traditional approach for managing software development lifecycle. Waterfall model have 5 principles & guidelines set that are essential to develop new software (Isaias & Issa, 2015). The processes are flowing sequentially starting from the requirements analysis till the deployment & maintenance stages. Each stage can’t start till the predecessor stage finishes its activities completely without overlapping with other stages. Figure (3.1) demonstrates the main stages of waterfall methodology.
According to (Stober, 2010) The Waterfall methodology are summarized in 5 phases:

- **Requirements Analysis:** All requirements of the system are gathered and documented after deep discussions with all involved stakeholders. These requirements include functional and non-functional requirements. Stakeholders includes any party whether individuals or groups that can have direct or indirect impact on the project.

- **Design:** All system designs are established in this stage. A set of charts & diagrams are used to shape out the sequences and functionalities of the processes.

- **Implementation:** System designs are interpreted into units of code, each unit implement a specific function in the application.

- **Testing:** In this stage, all the developed units are integrated together and tested against the planned requirements.

- **Support:** After the software delivered to the customer, the support stage starts according to the level of agreement with the customer.

### 3.3 Agile Methodology

Agile methodology aims to reduce the risks associated with the projects through breaking down the deliverables into several iterations. Each iteration has several tasks that last with a tangible functionality lead to the final product or service. Figure (3.2) demonstrates the iterations processes in agile methodology.
According to (Beck K. et al, 2001), Agile methodology allocate higher priority to people over the tools and processes, by focusing more on customer satisfaction and delivering products with high quality and proper functionalities. It is flexible, adaptive and welcome any new changes from the customer even at late stages. It promotes direct communications between the development team. It Gives preference for face to face interactions over prepared documents, and encourages the development team to work together.

The main benefits of using agile methodology through this project are the flexibility to accommodate any change requests during the application development lifecycle. It will influence the development paces to deliver the software with high quality, more frequency and at shorter time.

### 3.4 The Project Methodology is AGILE

This project will follow the agile methodology to develop the required application. Although waterfall methodology have many significant features, it is obvious that agile methodology has many advantages over waterfall methodology in respect to the software development aspects as well as the needs of this project.

Through the iterative planning and feedback processes, this project will easily adapt the software development with any changes in the requirements. It will ensure that project deliverables are aligned with the defined requirements and meet user expectations.
3.5 Evaluation Methodology

This section is mainly concerned with explaining the mechanisms used to evaluate the application. Understanding the evaluation methodologies will influence positively on the undertaken evaluation of this project and will ensure high quality of research results.

According to (Nkwake & Mayne, 2015), Evaluation is a set of tools and skills applied through social science research to determine the success of interventions within fields of practices. The role of evaluator is essential to help practitioners make better decisions on further developments and improve the ongoing processes of the project as well.

3.5.1 Purpose of Evaluation

The main purposes of performing evaluation in this thesis to get feedback from the participants on the successes, failures and lessons learned, as well as enhancing the functionalities of application, and address any defects in the future. In addition, it examines how far the project objectives have been accomplished. It is critical to identify the strengths and weaknesses of the application to improve the performance. Furthermore, it will reduce the risks that could negatively affect the application.

The evaluation of this application aims to check feasibility of the system, validate the functionalities and determine whether the application performing up to the expectations of users. It also examines the familiarity and acceptance of user interface.

3.5.2 Evaluation Types

According to (Chen, S. et al., 2011) as acknowledge by (Clarke, 1999), There are two common approaches used in evaluation:

1- Formative Evaluation:

The purpose of this approach to acquire feedback during the implementation and development processes. It is two way interactions which aims to improve the
development processes while it happens. It helps to monitor the progress of project toward achieving the goals (Tracy, 2015).

2- Summative Evaluation:
This Type of evaluation is carried out after the completion of the development processes. It is one way communication to gather the feedback from users to measure the impact and assess the effectiveness of the system (Chen, S. et al., 2011).

Overall, both of the evaluation types have some similarities and significant differences. The formative evaluation provides insights into the performance of functionalities, it comes with qualitative evidences needed to improve the system, while the summative evaluation will provide quantitative outcome measures.

This thesis adopts a combination of both the evaluation methodologies to collect results from participants who represent potential users of this application:

- Formative Evaluation: examine certain set of system functionalities, familiarity of user interface (UI) and User Experience (UX). Although it is highly recommended to consult subject-matter experts for some designing options, It may be difficult during the implementation due to the time constraint. However, This project will use a criteria-based evaluation which is conducted through a checklist of tasks, the users will be asked to accomplish them on the system.

- Summative Evaluation: In this evaluation, users will be asked a set of questions through a questionnaire after the project development completion to judge the worth of application and appraise the project success.
4 Requirements Analysis

4.1 Overview

System Requirements Analysis is a methodology used to identify the resources & characteristics required to fulfill all the system needs (Grady, 2014). During this phase, all the required tasks are encompassed to transform the customer needs into technical specifications of the system. Performing the following major activities in a proper manner are part of this phase in order to meet the user expectation, and avoiding any error that could impact the software success at later stages:

1- Eliciting requirements: gathering requirements from key stakeholders.
2- Analyzing requirements: validating the collected requirements and eliminate any ambiguity.
3- Modeling of requirements: articulating the collected requirements in several models like use cases or user stories.
4- Reviewing: improving the collected requirements through several iterations is needed to ensure providing the desired benefits.

4.2 Description of the Application

The scope of this project is developing a web application to help the visitors of EXPO 2020 - Dubai to connect together & find the planned events and activities during the exhibition period. The world EXPO extends up to 6 months with millions of visitors expected. The application will work as a guide for the visitors to provide them with the events details they need, in order to help them making a decision whether to attend or
not. In addition, it facilitates the communications between individuals as well as organizations to foster the collaboration between all participants.

As connecting people is the ultimate objective of EXPO 2020 - Dubai, all features provided by this application are underpinning this goal through a set of functionalities which empower the application. Some of these functions are: recommending events, enabling the users to find people who work in the same field, reaching them by instant messages, communicating directly with leading organizations via their representatives.

The application works to collect the visitor feedback as well as the rate on the events they participate in. Accordingly, the users can view a list of recommended events based on their rates, or get events favored by friends as well as co-friends. The application also provides filtering by industry, in which the events list is filtered by the same field. For example, if the visitor is working in the IT industry, and may be interested to get a list of all IT related events, then by a single click all events only related to IT field will be displayed.

This project is managed to create a web application based on Microsoft ASP.NET technology along with NoSQL graph database to store all of events, people and companies details.

The graph database is designed to handle effectively all queries requested by large number of users. The database is efficiently process all queries in high performance with no latency. This project uses the native graph structured model of the Neo4j database to build connections between people, organizations and the events they are interested in.

Graph database consists of entities (nodes) connected by relationships (edges), and both may have properties in name-value pairs. Graph database set properties for the nodes and relationships containing the necessary attributes. Example of these relationship types between one person and an event are: Review, Rate and Interest.
4.3 Project Requirements

4.3.1 Overview

A requirement is a desire or need by a particular product or service to be satisfied (Koelsch, 2016). It is a singular statement that focuses on a specific physical or functional need. It consists of three elements: function, verb and need. To bring out a successful project, all requirements of the project should be clear, understandable and achievable. According to (Stober, 2010), there are two major types of requirements for any application: functional requirement which describes a function through a user scenario, and non-functional requirement which sets constraints on system behaviors and specifies quality attributes.

This section identifies the functional and non-functional requirements that are required to be fulfilled at the implementation stage of this project.

4.3.2 Functional Requirements

Table (4.1) lists all the functional requirements of this project:

<table>
<thead>
<tr>
<th>ID</th>
<th>Type</th>
<th>Function / Requirement</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td><strong>Splash Screen</strong></td>
</tr>
<tr>
<td>1</td>
<td>Mandatory</td>
<td>Application shall have a splash screen with links to login &amp; registration pages.</td>
</tr>
<tr>
<td></td>
<td></td>
<td><strong>Login</strong></td>
</tr>
<tr>
<td>2</td>
<td>Mandatory</td>
<td>Application shall use form authentication on login page</td>
</tr>
<tr>
<td>3</td>
<td>Mandatory</td>
<td>User shall login simultaneously on the application and the messaging server</td>
</tr>
<tr>
<td></td>
<td></td>
<td><strong>Registration</strong></td>
</tr>
<tr>
<td>4</td>
<td>Optional</td>
<td>Registration form shall validate user entries</td>
</tr>
<tr>
<td>5</td>
<td>Mandatory</td>
<td>User shall create an account with all needed details on the registration page.</td>
</tr>
</tbody>
</table>
### Chapter 4. Requirements Analysis

<table>
<thead>
<tr>
<th></th>
<th>Mandatory</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>6</td>
<td></td>
<td>Registration Page shall allow only unique user ID</td>
</tr>
<tr>
<td>7</td>
<td></td>
<td>User ID shall be same on application &amp; messaging server</td>
</tr>
</tbody>
</table>

#### User Profile

<table>
<thead>
<tr>
<th></th>
<th>Mandatory</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>8</td>
<td></td>
<td>User shall update profile information</td>
</tr>
<tr>
<td>9</td>
<td>Optional</td>
<td>Page shall use both required &amp; expression fields validators when needed</td>
</tr>
</tbody>
</table>

#### Find Events

<table>
<thead>
<tr>
<th></th>
<th>Mandatory</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>10</td>
<td></td>
<td>User shall find a list of available events</td>
</tr>
<tr>
<td>11</td>
<td></td>
<td>User shall have a filtering option by event category</td>
</tr>
<tr>
<td>12</td>
<td></td>
<td>Page shall enable user to find preferred events of friends</td>
</tr>
<tr>
<td>13</td>
<td></td>
<td>Find event page shall allow searching connections up to level 5.</td>
</tr>
<tr>
<td>14</td>
<td></td>
<td>Every event shall show given rate when available</td>
</tr>
<tr>
<td>15</td>
<td></td>
<td>Every Event shall show the EXPO theme it belongs to</td>
</tr>
<tr>
<td>16</td>
<td></td>
<td>Events Page shall have the option to find friends of friends’ preferred events.</td>
</tr>
<tr>
<td>17</td>
<td></td>
<td>User shall add any event to events favorite list (Saved events page).</td>
</tr>
<tr>
<td>18</td>
<td></td>
<td>User shall have the option to remove the filter key</td>
</tr>
<tr>
<td>19</td>
<td></td>
<td>User shall filter the events list by metatag key.</td>
</tr>
<tr>
<td>20</td>
<td></td>
<td>Every event shall display a list of friends interested in that event</td>
</tr>
<tr>
<td>21</td>
<td>Optional</td>
<td>Every event shall show the event organizer</td>
</tr>
<tr>
<td>22</td>
<td>Optional</td>
<td>Page shall use AJAX for any update on user interface</td>
</tr>
</tbody>
</table>

#### Saved Events

<table>
<thead>
<tr>
<th></th>
<th>Mandatory</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>23</td>
<td></td>
<td>User shall view a list of saved events</td>
</tr>
<tr>
<td>24</td>
<td></td>
<td>Every Event shall contain the same details as shown in the find event page</td>
</tr>
<tr>
<td>25</td>
<td></td>
<td>User Shall remove the event from the saved (Favorite) list.</td>
</tr>
<tr>
<td>26</td>
<td></td>
<td>User shall rate the event within the saved list only</td>
</tr>
<tr>
<td>27</td>
<td></td>
<td>User shall review an event on the saved events list</td>
</tr>
<tr>
<td>28</td>
<td></td>
<td>Page shall be updated automatically once any event removed</td>
</tr>
</tbody>
</table>

#### Find People
<p>| | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>29</td>
<td>Mandatory</td>
<td>User shall find people and follow from a list</td>
</tr>
<tr>
<td>30</td>
<td>Mandatory</td>
<td>Person details shall be displayed with contact details</td>
</tr>
<tr>
<td>31</td>
<td>Optional</td>
<td>Page shall use AJAX for any update on the page</td>
</tr>
<tr>
<td></td>
<td></td>
<td><strong>Followed People</strong></td>
</tr>
<tr>
<td>32</td>
<td>Mandatory</td>
<td>User shall unfollow people from followed people list.</td>
</tr>
<tr>
<td>33</td>
<td>Mandatory</td>
<td>User shall communicate with any person by instant messages</td>
</tr>
<tr>
<td></td>
<td></td>
<td><strong>Search Companies</strong></td>
</tr>
<tr>
<td>34</td>
<td>Mandatory</td>
<td>User shall browse list of companies</td>
</tr>
<tr>
<td>35</td>
<td>Mandatory</td>
<td>User shall follow any company in the list</td>
</tr>
<tr>
<td>36</td>
<td>Mandatory</td>
<td>The list shall display company details and their industry</td>
</tr>
<tr>
<td>37</td>
<td>Mandatory</td>
<td>User shall filter company list by industry</td>
</tr>
<tr>
<td>38</td>
<td>Mandatory</td>
<td>User shall reach the company representatives and communicate with them</td>
</tr>
<tr>
<td></td>
<td></td>
<td><strong>Saved Companies</strong></td>
</tr>
<tr>
<td>39</td>
<td>Mandatory</td>
<td>User shall browse all followed companies</td>
</tr>
<tr>
<td>40</td>
<td>Mandatory</td>
<td>User shall unfollow from any company</td>
</tr>
<tr>
<td></td>
<td></td>
<td><strong>Activity Log</strong></td>
</tr>
<tr>
<td>41</td>
<td>Mandatory</td>
<td>User shall view all the activities done on the account</td>
</tr>
<tr>
<td>42</td>
<td>Optional</td>
<td>User shall delete any activity from activity log</td>
</tr>
<tr>
<td></td>
<td></td>
<td><strong>Instant Messages</strong></td>
</tr>
<tr>
<td>43</td>
<td>Mandatory</td>
<td>User shall update chat presence</td>
</tr>
<tr>
<td>44</td>
<td>Mandatory</td>
<td>Application shall use XMPP to enable instant messages</td>
</tr>
<tr>
<td>45</td>
<td>Mandatory</td>
<td>User Shall receive the message instantly through a pop-up window</td>
</tr>
<tr>
<td></td>
<td></td>
<td><strong>Database</strong></td>
</tr>
<tr>
<td>46</td>
<td>Mandatory</td>
<td>Application shall use NoSQL graph database</td>
</tr>
<tr>
<td></td>
<td></td>
<td><strong>Admin Control Panel</strong></td>
</tr>
<tr>
<td>47</td>
<td>Mandatory</td>
<td>Administrator shall add new event</td>
</tr>
<tr>
<td>48</td>
<td>Mandatory</td>
<td>Administrator shall add new company</td>
</tr>
</tbody>
</table>

*Table 4.1: Functional Requirements*
4.3.3 Non-Functional Requirements

Table (4.2) list all non-functional requirements of the project:

<table>
<thead>
<tr>
<th>ID</th>
<th>Type</th>
<th>Area / Requirement</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td><strong>User Interface</strong></td>
</tr>
<tr>
<td>49</td>
<td>Mandatory</td>
<td>Application shall be in English language</td>
</tr>
<tr>
<td>50</td>
<td>Mandatory</td>
<td>Application shall have user friendly interface with appropriate coloring style</td>
</tr>
<tr>
<td>51</td>
<td>Mandatory</td>
<td>All application functions shall go through a proper test</td>
</tr>
<tr>
<td>52</td>
<td>Optional</td>
<td>All application CSS files shall be placed in one directory</td>
</tr>
<tr>
<td>53</td>
<td>Optional</td>
<td>All application Java Script files shall be placed in one directory</td>
</tr>
<tr>
<td></td>
<td></td>
<td><strong>Find Event</strong></td>
</tr>
<tr>
<td>54</td>
<td>Mandatory</td>
<td>Show saved event icon in different color</td>
</tr>
<tr>
<td>55</td>
<td>Optional</td>
<td>Page shall use indicative icons on the events</td>
</tr>
<tr>
<td>56</td>
<td>Optional</td>
<td>Every event shall show different background color when mouse go over</td>
</tr>
<tr>
<td>57</td>
<td>Mandatory</td>
<td>Event Cost shall be displayed with 2 decimals at most</td>
</tr>
<tr>
<td></td>
<td></td>
<td><strong>Find People</strong></td>
</tr>
<tr>
<td>58</td>
<td>Optional</td>
<td>People list shall display person picture if available</td>
</tr>
<tr>
<td></td>
<td></td>
<td><strong>Registration</strong></td>
</tr>
<tr>
<td>59</td>
<td>Optional</td>
<td>Application shall only apply complicated password policy.</td>
</tr>
<tr>
<td></td>
<td></td>
<td><strong>Search Companies</strong></td>
</tr>
<tr>
<td>60</td>
<td>Mandatory</td>
<td>Company Details shall be displayed on mouse over</td>
</tr>
</tbody>
</table>

*Table 4.2: Non-Functional Requirements*

4.4 Use Cases

4.4.1 Overview

Use case is a descriptive text of actions that explain the sequenced interactions between the users and a system (Souza, 2016). Use cases are using a natural language to represent the detailed description for each of the system functionalities.
Use case is one of the significant approaches used to present the information in a model that aid in the software development, and provide better understanding of the user requirements as well. Functional and Non-Functional requirements in the previous section are used here as the foundation for creating use cases which cover different aspects of the system requirements.

Figure (4.1) shows a top level UML use case diagram of the entire application which explain the relations between actors and the system. According to (Souza, 2016), a detailed description of each use case diagram must be represented to explain the underlying interactions between the actor and the system. Therefore, the next sections drill down in each use case to provide further details along with the use case diagram pertained to the underlying functionality.

**UML Tool: UMLET**

![Figure 4.1: Top Level Use Case Diagram](image-url)
4.4.2 Use Case Index

Table (4.3) lists the index of main use cases of the project:

<table>
<thead>
<tr>
<th>Use Case ID</th>
<th>Use Case Name</th>
<th>Primary Actor</th>
<th>Scope</th>
<th>Complexity</th>
<th>Priority</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Register New User</td>
<td>New User</td>
<td>In</td>
<td>Medium</td>
<td>3</td>
</tr>
<tr>
<td>2</td>
<td>Login Registered User</td>
<td>Registered User</td>
<td>In</td>
<td>Medium</td>
<td>3</td>
</tr>
<tr>
<td>3</td>
<td>Browse Events Registered User</td>
<td>Registered User</td>
<td>In</td>
<td>High</td>
<td>1</td>
</tr>
<tr>
<td>4</td>
<td>View Favored Events Registered User</td>
<td>In</td>
<td>High</td>
<td>2</td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>Find People Registered User</td>
<td>Registered User</td>
<td>In</td>
<td>High</td>
<td>1</td>
</tr>
<tr>
<td>6</td>
<td>View Followed People Registered User</td>
<td>In</td>
<td>High</td>
<td>2</td>
<td></td>
</tr>
<tr>
<td>7</td>
<td>Browse Organizations Registered User</td>
<td>In</td>
<td>High</td>
<td>1</td>
<td></td>
</tr>
<tr>
<td>8</td>
<td>View Followed Organizations Registered User</td>
<td>In</td>
<td>High</td>
<td>2</td>
<td></td>
</tr>
</tbody>
</table>

*Table 4.3: Use Case Index*

4.4.3 Application Use Cases

The actual use case as stated by (Darren, 2014) is identifying the components used in each use case through a textual representation, in which the sequence events are explained to illustrate the underlying functions and provide a clear understanding.

The following sections focus on the main use cases diagrams and their components to grasp the project details:

4.4.3.1 Register Use Case

Figure (4.2) shows the Register use case diagram, which demonstrates the interactions between the key actors and the system.
Figure 4.2: Register Use Case Diagrams

<table>
<thead>
<tr>
<th>Use Case ID</th>
<th>1</th>
</tr>
</thead>
<tbody>
<tr>
<td>Use Case Name</td>
<td>Register Use Case</td>
</tr>
<tr>
<td>Use Case Description</td>
<td>Collecting user details are required to start using and get benefits of the functions provided by the application. This information is used to connect people together, and suggest some events accordingly.</td>
</tr>
<tr>
<td>Use Case Goal</td>
<td>To Register user on the application</td>
</tr>
<tr>
<td>Preconditions</td>
<td>No other users registered in the database with the same user ID</td>
</tr>
<tr>
<td>Primary Actor</td>
<td>New User</td>
</tr>
</tbody>
</table>
| Success Scenario | 1- Fill all fields of the registration form.  
2- The system check the validity of the given information.  
3- The system check the user ID against the database to ensure no duplication.  
4- The system create a new node (Record) in the database with user details.  
5- The system create a new user in the messaging server. |
| Exceptions | 1- User failed to enter valid data at client side.  
2- Messaging Server is not running.  
3- Database service is not running. |

4.4.3.2 Login Use Case

Figure (4.3) shows Login Use Case diagram, which demonstrates the interactions between the key actors and the system.
Chapter 4. Requirements Analysis

Figure 4.3: Login Use Case Diagram

<table>
<thead>
<tr>
<th>Use Case ID</th>
<th>2</th>
</tr>
</thead>
<tbody>
<tr>
<td>Use Case Name</td>
<td>Login Use Case</td>
</tr>
<tr>
<td>Use Case Description</td>
<td>Checking the validity of the user name &amp; password against the database to ensure the user is registered. The user get access to the application functions, and sign on the messaging server. Administrator can also login on the same page using the admin credentials, then will be redirected to admin control panel.</td>
</tr>
<tr>
<td>Use Case Goal</td>
<td>To sign user on the application and the messaging server.</td>
</tr>
<tr>
<td>Preconditions</td>
<td>User should have an account on the application.</td>
</tr>
<tr>
<td>Primary Actor(s)</td>
<td>Registered User.</td>
</tr>
<tr>
<td></td>
<td>Admin User.</td>
</tr>
<tr>
<td>Success Scenario</td>
<td>1- Enter the user name and password.</td>
</tr>
<tr>
<td></td>
<td>2- The system check the validity of the given information.</td>
</tr>
<tr>
<td></td>
<td>3- The system sign the user on the messaging server.</td>
</tr>
<tr>
<td></td>
<td>4- The system redirect the user to a landing page.</td>
</tr>
<tr>
<td></td>
<td>5- If the user is an administrator, user will be redirected to the administrator control panel</td>
</tr>
<tr>
<td>Exceptions</td>
<td>1- User failed to enter valid data at client side.</td>
</tr>
<tr>
<td></td>
<td>2- Messaging Server is not running.</td>
</tr>
<tr>
<td></td>
<td>3- Database service is not running</td>
</tr>
</tbody>
</table>
4.4.3.3 Browse Events Use Case

Figure (4.4) shows Browse Events Use Case diagram, which demonstrates the interactions between the key actor and the system.

![Browse Events Use Case Diagram](image)

*Figure 4.4: Browse Events Use Case Diagram*

<table>
<thead>
<tr>
<th>Use Case ID</th>
<th>3</th>
</tr>
</thead>
<tbody>
<tr>
<td>Use Case Name</td>
<td>Browse Events Use Case</td>
</tr>
<tr>
<td>Use Case Description</td>
<td>It shows some interactions on the browse events page which enable the registered user to browse a list of events with all details to choose the interesting one, then add to the favorite list. It also shows a user can filter the list with the event field (category) as well as filtering the list with a metatag in which the event belongs to. Users can find all the events preferred by their friends or by friends of friend (Co-friends) by a single click.</td>
</tr>
<tr>
<td>Use Case Goal</td>
<td>User can search events preferred by others up to 5 connection levels.</td>
</tr>
<tr>
<td>--------------</td>
<td>---------------------------------------------------------------</td>
</tr>
<tr>
<td>Preconditions</td>
<td>User should have an account on the application.</td>
</tr>
<tr>
<td>Primary Actor</td>
<td>Registered User.</td>
</tr>
</tbody>
</table>
| Success Scenario | 1- Browse the events page.  
2- Click add to favorite button on any event.  
3- Click the field button to filter the events list by category.  
4- Click the metatag key to filter the events list by metatag.  
5- View the friends who are interested in any event.  
6- Click the buttons of events preferred by friends or by co-friends.  
7- Search up to 5 levels of connections with other people. |
| Exceptions | 1- Database service is not running. |
**Use Case ID** | 4  
---|---  
**Use Case Name** | View Favored Events Use Case  
**Use Case Description** | It shows some interactions on favored events, in which the registered user can view all the favorite events, remove any of the displayed events and filter the list with the event field (category) as well as the metatag in which the event belongs to. Users can rate the events they attend, In addition to review it with any comments.  
**Use Case Goal** | Demonstrate browse favorite events function, In addition to the actions the user can do on the page.  
**Preconditions** | User should have an account on the application. At least one event should have been added to the favorite list.  
**Primary Actor** | Registered User.  
**Success Scenario** |  
1- Browse the favored events page.  
2- Click remove button to Un-favorite any event.  

*Figure 4.5: View Favored Events Use Case Diagram*
Chapter 4. Requirements Analysis

3- Click the field button to filter the favored events list by category.
4- Click the metatag key to filter the favored events list by metatag.
5- View the friends who are interested in any event.
6- Click the rate button.
7- Enter a comment on event.

Exceptions
1- Database service is not running.

4.4.3.5 Find People Use Case

Figure (4.6) shows Find People Use Case diagram, which demonstrates the interactions between the key actors and the system.

![Find People Use Case Diagram](image)

**Figure 4.6: Find People Use Case Diagram**

<table>
<thead>
<tr>
<th>Use Case ID</th>
<th>5</th>
</tr>
</thead>
<tbody>
<tr>
<td>Use Case Name</td>
<td>Find People Use Case</td>
</tr>
<tr>
<td>Use Case Description</td>
<td>Find People is a list of people with their details, in which the registered user can follow any of those persons, filter the list with the field of a particular person as well as the organization they belong to.</td>
</tr>
<tr>
<td><strong>Use Case Goal</strong></td>
<td>It shows the interactions between the registered user and Find people list. Once the user follow someone, then the followed people list will automatically updated.</td>
</tr>
<tr>
<td>------------------</td>
<td>--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td><strong>Preconditions</strong></td>
<td>User should have an account on the application.</td>
</tr>
<tr>
<td><strong>Primary Actor</strong></td>
<td>Registered User.</td>
</tr>
</tbody>
</table>
| **Success Scenario** | 1- Browse people from various fields.  
2- Click Follow button to add person to favorite list.  
3- Click the field button to filter the list by industry.  
4- Click the metatag key to filter the people list by metatag.  
5- View friends who are have connection with a particular person. |
| **Exceptions**    | 1- Database service is not running.                                                                                                                                                                |

### 4.4.3.6 People I am following Use Case

Figure (4.7) shows People I am following (Followed People) Use Case diagram, which demonstrates the interactions between the key actors and the system.

![View Followed People Use Case Diagram](image)
Chapter 4. Requirements Analysis

Use Case ID | 6
---|---
Use Case Name | Followed People / People I am following Use Case
Use Case Description | A list of favored persons (Friends) with their details, in which the registered user can unfollow any of those persons, filter the list with the field of a particular person as well as the organizations they belong to.
Use Case Goal | It shows the interactions between the registered user and the followed person, In addition to the communication with a friend through the messaging server.
Preconditions | User should have an account on the application.
Primary Actor | Registered User.
Success Scenario | 1- Browse people from followed list.
2- Click Unfollow button to remove person to your favorite list.
3- Click the field button to filter the list by industry.
4- Click the metatag key to filter the people list by metatag.
5- View friends who are have connection with a particular person.
6- Click Communicate icon on any person to start sending instant messages.
Exceptions | 1- Database service is not running.
2- Messaging server is not working.

4.4.3.7 Browse Organizations Use Case

Figure (4.8) shows Browse Organizations Use Case diagram, which demonstrates the interactions between the key actors and the system.
### Use Case Description

A list of organizations with their details, in which the registered user can follow any of those organizations, filter the list with the field of a particular organization as well as the countries they belong to. Communicate with organization representatives using instant messages as well as emails. User can view friends who are working in any particular organization.

### Use Case Goal

It shows the interactions between the registered user and the organizations list, In addition to the communication with organization representatives through the messaging server.

### Preconditions

User should have an account on the application.

### Primary Actor

Registered User.

### Success Scenario

1. Browse organizations.
2. Click follow button to add organization to your favorite list.
3. Click the field button to filter the list by industry.
4. Click the country button to filter the list by country.
5. View friends who are working in a particular organization.
4.4.3.8 Organizations I am Interesting in Use Case

Figure (4.9) shows Organizations I am interesting in (Followed Organizations) Use Case diagram, which demonstrates the interactions between the key actors and the system.

<table>
<thead>
<tr>
<th>Use Case ID</th>
<th>8</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Use Case Name</strong></td>
<td>Followed Organizations/Organizations I am Interesting in Use Case</td>
</tr>
<tr>
<td><strong>Use Case Description</strong></td>
<td>A list of followed organizations with their details, in which the registered user can unfollow any of those organizations, filter the list with the field of a particular organization as well as the countries they belong to. Communicate with organization representatives using instant messages as well as emails. User can view friends who are working in any particular organization.</td>
</tr>
</tbody>
</table>
Use Case Goal

It shows the interactions between the registered user and the followed organizations list, in addition to the communication with organization representative through the messaging server.

Preconditions

User should have an account on the application.

Primary Actor

Registered User.

Success Scenario

1- Browse followed organizations.
2- Click unfollow button to remove organization from your favorite list.
3- Click the field button to filter the list by industry.
4- Click the country button to filter the list by country.
5- View friends who are working in a particular organization.
6- Click Communicate icon on any organization to start sending instant messages to their representatives.

Exceptions

1- Database service is not running.
2- Messaging server is not working.

4.5 Graph Data Model

From the project requirements explained in the previous sections, it is obvious that the main objects in this project are required to be connected through direct and indirect relationships. Graph database plays a vital role in the project to link these entities and handle traverse queries easily & efficiently. One of the graph database key strengths is retrieving complex data which has indirect relationships by using simple queries rather than using multiple join queries as in relational database.

The following figure (4-10) shows the graph data model of the key entities and their relationships within the application. It demonstrates the interaction types between each node with the others, and the directions of these relationships. Each relationship could have properties that describe the connection between the nodes.
Figure 4.10: Graph Data Model
4.5.1 Connecting People

Finding & connecting people who share the same concerns or working on the same field are substantial to increase the cooperation and foster the communications between people.

The graph database utilized to connect people together based on two types of recommendations:

1- Personalization: In this approach, the user could find people based on some logical recommendations derived from their personal information like their working field, organization. The user has no influence over the displayed recommendations, they are completely built upon the relationships between both of the user and the data saved from other users. However, the user could filter these recommendations list by some of the main fields.

2- Customization: The users view the recommendations based on the relationships they create with other people as well as the events they get interested in. So the users control the recommendations they view based on the persons they follow. For examples, the users could search for people who may have connections with their first-level connections (direct friends), or find persons who have relations with their own friends (co-friends). The users could also view the persons they follow, and have the control to unfollow at any time.

4.5.2 Connecting Persons with Events

Successful exhibitions are keen on making visitors attend the appropriate events that meet their needs and exceed their expectations.

The graph database play a vital role to build relationships between the project entities, and facilitate the data retrieval in a graph structure which modeling the real representation and the actual connections of the persons and events. The following approaches are used to link the persons and events in the applications:
1- **Personalization:** The users view a list of recommended events based on the level of connections they have with other persons or events. They view the details of events as well as the category that events belong to. Users could personalize the list based on the field or metatags of that events.

2- **Customization:** the users control the events items they view based on the events added to their favorite list, as well as the relationships they build with other persons. Users could also control the events list based on the events rate given by users.

### 4.5.3 Connecting Persons with Organizations

Throughout the exhibition period, people are interesting to approach some organizations to have further communications or initiate a new business channel.

The relationships between persons and organizations are implemented through the following two approaches:

1- **Personalization:** when users enter their organization name at the registration process, then this information is used to show friends who work in that organization.

2- **Customization:** The user filter the list using the country or field of any organization. Each organization could have one or two representatives to communicate with people. People could reach one of the representatives using their contact details or sending instant messages.

### 4.6 Instant Messaging Platform

Connecting people through instant messages is one of the significant features in this project to enable users communicating simultaneously. So, it is essential to figure out an appropriate platform which maintain the messages delivery between users. By facilitating instant communications between person to person and person to organization, the application will enrich the user experience and underpin the ultimate objectives of EXPO 2020 – Dubai.
There are several technologies available which support messaging between web clients. This project uses Extensible Messaging and Presence Protocol (XMPP) to enable instants messages between users. XMPP is compatible with Microsoft ASP.NET programming language used in the project. This project uses agsXMPP library, which is an official SDK (Software Development Kit) to integrate Microsoft .NET applications with XMPP functionalities.

The application requires to allow users controlling the presence status shown for others using XMPP. The user could also show a different status for a particular user within the chat window. XMPP is powered by (openfire) server, open licensed server to handle instant messages and chat groups.

4.7 Project Components

This project consists of the following main elements:

1- Application: Web-based application to enable all participants of EXPO 2020 - Dubai to connect together and find events based on their preferences. This web application has many features with user friendly interface that reflect the morale and themes of EXPO 2020 - Dubai.

2- Graph Database: The application is based on graph database. The NoSQL graph database used to connect people, organizations and events all together by native relationships to handle query connections and store data efficiently.

3- Messaging Server (XMPP): Extensible Messaging and Presence Protocol is a protocol to facilitate communications between users. It provides real-time operations for instant messages.
5 Professional, legal, ethical, and social issues

5.1 Professional Issues

Professionals have a significant role to shape the future of society by adhering to the code of ethics when they do their duties (meherchilakalapudi, 2009). Professional issues come out when people disregard the standards and regulatory obligations, or not adhere to the codes of conduct throughout the software development lifecycle.

This project will follow the software development standards with highest quality. The software will be tested properly to ensure its functionalities are working well. Any third party tools used in the project will be licensed and clearly mentioned. All intellectual properties will not be used unless it is approved by the owner.

5.2 Legal Issues

Legal Issues are the legal implications for misusing the technology or violating the copyrights during the software development process. Confidentiality, privacy and intellectual property & distributions all are main legal aspects that should be considered when developing an application (Sophie, 2015).

All programs and third parties software used throughout the development of this project are licensed. Any logos, slogans or contents in this application will be used upon a written approval from the owner. All the users’ details and logging information are stored in the database and used for testing purpose only.
5.3 Ethical Issues

Ethics is a set of principles that govern an individual or a group (Gary, 2006). In this context, there are several issues may be raised at the software implementation stage from wide areas like (Privacy, Encryption, Trust and Intellectual property).

In this project, all code or open-source tools used are comply with the rules of the distribution, copying and modifications. Any person participates in the evaluation process will be informed about the purpose of the application and all their information will be highly confidential.

5.4 Social Issues

University of Minnesota (2010) says that social issue or problem is defined as any social condition that affect negatively with consequences on large number of people, and need to be addressed.

This project doesn't affect or being affected directly with any social issues. It will not exclude a particular culture or groups during the implementation. Moreover, all data collected from users during the evaluation process will be confidential.
6 System Design

6.1 Overview

This Chapter explains how the system is designed according to the functional & non-functional requirements which are presented at the Requirements Analysis chapter. It shows the relationships between the main system classes and how the interactions happening between them. It explained the design of the graph database and the fields of the database entities. It discusses the principles used to design the user interface of the application to meet the user expectations.

6.2 Class Diagrams

Modelling the project main classes using class diagrams illustrates the conceptual design of the application. Class diagrams show the main objects and their relationships along with the key attributes used within each class.

The following sections demonstrate the class diagrams and explain the main classes used for implementing certain functions within the application, it is used to map out the high level processes of the system.

6.2.1 Class Diagram for Events

Figure (6.1) demonstrates a class diagram for the classes used by find events pages.
1- New4jConnection Class: This class is used to initiate the connection with the Neo4j graph database, it has a method to manage the connection through Neo4J driver. All the other classes fully depend on this class to access the graph database to execute queries and retrieve data.

2- Event Class: This class is used as a generic data type by the classes of web pages for storing the details of event objects.

3- Find Events Class: This class is used to read events from the graph database and display the query results to the user. It has many methods to control and filter the events according to the user preferences.

4- Saved Events Class: This class display the events which are saved by the users to their favorite list.

Figure 6.1: Class Diagram for Events
6.2.2 Class Diagram for Persons

Figure (6.2) demonstrates a class diagram for the classes used by find people page.

New4jConnection Class: explained in Section (6.2.1)

1- Person Class: This class used as a generic data type by the classes of web pages for storing the details of person objects.

2- Find People Class: This class is concerned with displaying a list of persons based on some recommendations criteria.

3- Followed People Class: Also called (People I am following), this class manage the list of persons the user follow and add into the favorite list.

Figure 6.2: Class Diagram for Persons
6.2.3 Class Diagram for Companies

Figure (6.3) demonstrates a class diagram for the classes used by browse companies page.

1- New4jConnection Class: explained in section (6.2.1)
2- Company Class: used as a generic data type by the classes of web pages for storing the details of company objects.
3- Search Companies Class: This class is designed to list the organizations along with their general details.
4- Saved Companies Class: Also called (Organizations I am Interesting in), this class basically retrieve the organizations saved by the user from the favorite list.

Figure 6.3: Class Diagram for Companies
6.3 Database Design

This Project uses Neo4j Graph Database to store the data in a graph structure. A graph is set of nodes and relationships that connect them to build semantic context (Robinson I. et al, 2015). The graph database has no fixed schema, which make it ideal choice for our project. It allows adding more nodes and relationships without affecting the original data.

The graph database of this project is consisting of 3 main entities:

1- Person: Every node of this object is used to store information of one person in name-value pairs. The node is tagged with a label PERSON with value of a person name. Below is the list of properties exist within each node along with short description:

- user_id: Unique ID created by user
- password: hashed password of user
- name: combination of First and Last name of user
- fname: first name of user
- lname: last Name of user
- country: country of user
- industry: career field of user
- designation: job Title of user
- organization: company name where the user work
- gender: type of user
- age: how old the user is
- contact: Phone or Mobile number of user
- email: Email address of user

2- Event: a node from this object type contains all the details of an event. The node has a label EVENT with value of an event name, and store data in name-value pairs.
The following list are the properties available in each Event node with short description for each:

- title: The name of event
- description: brief description of the event
- startDate: start date of event
- endDate: end date of event
- location: the venue where event happens
- organizer: the sponsor of event
- field: the business area of event
- category: classification of event
- cost: the price of event ticket (if applicable)
- metatags: keywords used for filtering and searching

3- Organization: A single organization node contains company information in name-value pairs. Every node has a label ORGANIZATION with value of an organization name. Below is the list of properties exist within each node along with short description:

- orgID: unique GUID
- name: name of company
- about: brief description on company
- industry: business area of company
- country: country where company head office located
- website: URL of company public website
- email: official contact email of company
- contact: official contact number of company
- contact_person1: user ID of company representative
- contact_person2: user ID of company representative
- metatags: keywords used for filtering & searching
Figure 6.4 demonstrates a graph representation of the connections between the 3 main entities in the application, and the relationships used to link between nodes.

Figure 6.4: Graph Model of Main Entities

6.4 User Interface Design

User Interface (UI) design plays a significant role in any software success. UI is the access points where users interact with the system. System design should anticipate the user needs & interactions, and provide users with engaging interfaces matching their expected behaviors.
As stated by (Joshua, 2018), Clarity comes on the top aspects required in system design. It helps people to recognize and understand the use of the system and encourage them to use it frequently. Through predicting the actions the user do and providing the suitable interactions, the system will inspire the user with confidence to use it again.

The application is designed to be familiar with the users through many designing aspects: the layouts are consistent in all pages, unified coloring style across pages, using appropriate fonts sizing & coloring, and handing the control to the user at every turn. However, due to the time constraint of this project, further aesthetic aspects of the application were disregarded.

6.4.1 ASP.NET Master Page

ASP.NET Master Page used to create a consistent & centralized layout in all pages. The layout is predefined as a template with all the elements required to be used by the content pages (Evjen et al, 2010). These common UI elements includes images, scripts and CSS files needed to be unified across the content pages which are referenced from the master page. Figure (6.5) illustrates the layout used in the application to build content pages based on a single master page.

![Figure 6.5: ASP.NET Master Page Layout](image)
The master page of the application is used by multiple content pages. The main master page has ContentPlaceHolder Control which reserve a space area for the content pages to render its elements. Changing the common elements inside the master page will dynamically reflect on all content pages referenced from that master page without any intervention from the developer.

On the other hand, the content pages use a Content Control to bind the elements placed inside into the reserved place within the master page. As stated by (Evjen et al, 2010), Each content page requires to specify which master page to use either at page level using inline-coding model in the page directive or at application level in web.config file.

The Navigation bar on the left side of the Master Page is designed using simple HTML code along with Cascading Style Sheets (CSS). The focused design makes the user feels comfortable and pleasant to access any of the main functionalities with a single click.

The header bar is designed to hold a welcome message to the user, and display the current presence status received from the messaging server through a drop down list.
7 System Implementation

7.1 Overview

This chapter will go through the preparations of the development environment required to implement the project, Exploring the Integrated Development Environment (IDE) and tools used through the software development.

The system implementation phase include the coding, debugging, compilation and testing stages. This chapter explains some functions developed within the application pages, and demonstrates some of the code snippets used to build significant functionalities within the system. Most of the code snippets which are used for demonstration throughout this chapter are placed in Appendix C.

This chapter also comes through the cypher query language used to retrieve the data from Neo4j graph database. It also shows some of queries to build relationships between entities.

7.2 Development Tools Setup

Before starting the development phase, there are some tools required to be installed. All the tools used for the software development are open and licensed free.

The following Tools have been installed and configured on the development machine:

1- Visual Studio - Community Edition:
Visual Studio is a fully Integrated Development Environment (IDE) to write, debug and test code. It is designed to create various types of applications based on Microsoft .NET Framework. Microsoft .NET framework run as the compiler platform for the programming languages (Visual Basic and C#). ASP.NET is web
framework with open source license used for websites & web applications development.

2- Neo4j Graph Database – Community Edition:
Neo4j is NoSQL graph database with open source license. It is fully supporting .NET platform through a driver to connect the application with the database using binary protocol (Neo4j, 2018). This project use Neo4j Community Edition version 3.3.5.

3- Openfire Server:
It is open source licensed server used for instant messaging, it is adopting (Jabber) protocol, or also called (XMPP) Extensible Messaging and Presence Protocol (Openfire, 2018). This project uses Openfire version 3.7.1.

4- agsXMPP SDK:
agsXMPP is open source library for XMPP, it is dedicated to integrate .NET applications with XMPP server. It is shipped with a set of features for chatting and handling message events.

7.3 Neo4j Database Setup

The nature of this project is mainly depends on building connections between entities. Neo4j is native graph database management system to build such a graph model. As mentioned earlier, this project uses Neo4j community edition which is released with basic features meet the project requirements.

The database installation & configuration are done straight forward in few simple steps. Creating a new database file and establishing the connection to the application are essential to start inserting data. Finding proper .NET driver of Neo4j graph database is mandatory to integrate ASP.NET with the graph database.
There are three main nodes used throughout the application. All nodes are created dynamically in the application using a cypher query language. All nodes have a label to identify the node type (e.g. PERSON, EVENT or ORGANIZATION). Some edges (Relationships) between nodes may have properties describing the connection. Figure (7.1) demonstrates real data retrieved from the graph database.

![Graph Database Diagram](image)

**Figure 7.1: Graph Database**

### 7.4 System Development

#### 7.4.1 ASP.NET Master Page

ASP.NET Master Page use a predefined layout with a single look and feel across all the content pages (Microsoft Developer Network, 2018). At runtime, it consolidates the content pages with the Master Page layout to produce the output page. The master page could contain one or more ContentPlaceHolder controls. These placeholder controls are reserving space for the content pages.
This project uses ASP.NET Master Page to place the navigation menu and the top bar on all pages consistently.

The below replaceable elements are placed in Master page file to reserve regions where the head & contents will appear.

```html
<asp:ContentPlaceHolder ID="head" runat="server"></asp:ContentPlaceHolder>
<asp:ContentPlaceHolder ID="ContentPlaceHolder1" runat="server"></asp:ContentPlaceHolder>
```

The following element is placed in the content page which reference the placeholder control in the Master Page. All the elements inside the content control will replace the placeholder at runtime.

```html
<asp:Content ID="Content1" ContentPlaceHolderID="ContentPlaceHolder1" runat="server"></asp:Content>
```

The following declaration is required on the top of each content page to reference the master page layout.

```html
<%@ Page Title="" Language="C#" MasterPageFile="~/Main.Master" AutoEventWireup="true" %>
```

### 7.4.2 System Functionalities

#### 7.4.2.1 User Registration

The users of this application are required to register firstly with their basic information. This information is stored in graph database within a node labeled with the user ID. For security purpose, the password is get encrypted before stored in the graph database. The same user name and password provided by the user are also used at openfire server to create a new account for the user.

The code snippet (1.1.1) located in Appendix C, for CreatePerson() method, is creating a new person entity in the graph database from the information collected from the user.

The code snippet (1.1.2) illustrates how the new user account is created at openfire server using XMPP library.
7.4.2.2 User Login

This function uses the credentials provided by the user to validate against the graph database. The login method evaluates the results to authenticate the user or reject. If the user inputs match the data stored in DB, then the user is signed in and redirected to a landing page, otherwise, an error message is displayed to alert the user.

Once the user get authenticated, the system keep the user information throughout the session lifetime to reduce the requests against the database. The code snippet (1.2.1) explains the login function.

At the time the user is authenticated on the application, another authentication process happens on the XMPP server. The code snippet (1.2.2) is authenticating the user on the XMPP server.

7.4.2.3 Find Events

Finding Events is one of the powerful features in the application. It provides users with a list of events along with details needed to help them deciding about it. Figure (7.2) shows a sample event which is displayed on the events list. It shows the friends who are interested in that event, an icon representing the EXPO 2020 Dubai theme in which the event pertained, and a rate which was given by other users for that event.

![Figure 7.2: Event Details](image)

The following code shows a query statement in cypher query language to be executed against the graph database to return all the events.
The users can easily find the events liked by their friends or friends of friends. In addition, Users have facility to search the events based on their connection level with others. The following cypher query is used to retrieve the events liked by friends of a user:

```cypher
match (n:PERSON{user_id:"""" + User.user_id + """"})-[[:Knows]-[Friends]-[:Saved]->_(events:EVENT) return DISTINCT _events
```

The following cypher query used to retrieve the events liked by friends of friends for a particular user:

```cypher
match (n:PERSON{user_id:"""" + User.user_id + """"})-[[:Knows]-[x]-[:Knows]-(CoFriends), (Cofriends)-[:Saved]->_(events:EVENT) return DISTINCT _events
```

The following code shows a query used to search events based on the connection level specified by the user:

```cypher
MATCH (n:PERSON{user_id:"""" + User.user_id + """"})-["1.." + ddllevel.SelectedItem.Text + "]-_(events:EVENT) return DISTINCT _events
```

As the users have full control on Filtering the list with Event fields or by metatags, the code snippet (1.3.1) explains how the command is handled to update the list accordingly.

The code snippet (1.3.2) for a method to get the average rate for a particular event:

### 7.4.2.4 Saved Events

Events I like or Saved Events list is a favorite list of all events the user like and plan to attend during the exhibition. It has most of the features available in the events list, in addition to rating and reviewing events. Users can dislike any of the events available in the list, then the list get updated automatically upon the user request. Figure (7.3) illustrates the main elements of saved event.
The code snippet (1.4.1) shows a cypher query used to delete the relationships between a particular user and the event entities.

The code snippet (1.4.2) create a new relationship between a user with an event entities for the purpose of reviewing & rating respectively.

7.4.2.5 Find People

Connecting people together during the world exhibition EXPO 2020 - Dubai is the ultimate objective of the entire event. Enabling the users of application to find people who have the same thoughts or working in common fields will underpin the connectivity and cooperation between people.

This application enable users to view list of people with brief information about every person, and may follow/add to the favorite list for further communications. The user can filter the list by the field name or organization of any person. Figure (7.4) shows a sample for person details along with the common events between the user and that person.
The following cypher query statement is used to retrieve all persons from the graph database:

```cypher
match (_persons:PERSON) where _persons.user_id <> "+ User.user_id + @"" return _persons
```

The next significant Cypher query is used to recommend some persons based on the common events both the user and those persons are interested in:

```cypher
match (_per:PERSON{user_id:"" + User.user_id + @""})-[[:Saved]]-(_events:EVENT)-[[:Saved]]-(_persons)
where Not (_per)-[:Knows]-(_persons) return distinct _persons
```

One of the functions this application provides to the users is searching for people by connection level. The user can view all people who have any kind of relationships within the specified connection level. The below cypher query shows an example:

```cypher
MATCH (n:PERSON {user_id:"" + User.user_id + @""})-[[:Knows *1.."" + ddlLevel.SelectedItem.Text + @"]]-(_persons:PERSON) where _persons.user_id <> "+ User.user_id + @"" return DISTINCT _persons
```

The block of code (1.5.1) for a method used to return the common events between the user and a particular person.

For every person displayed on the people list, an icon of common friends might be shown to indicate that some persons are shared friends between the user and that person. The block of code (1.5.2) for a function used to check the common friends between the user and other persons.

In order to follow a particular person from the list, the user needs to select the person and click add to favorite. The next cypher query statement shows how a relationship between 2 persons is created when the user click on add to the favorite.

```cypher
match (_person1:PERSON{user_id:"" + User.user_id + @""}),(_person2:PERSON{user_id:"" + UserToFollow + @""}) create (_person1)-[:Knows]->(_person2)
```

7.4.2.6 People I am Following

This page shows a list of people who have been added to the favorite list. It makes the favored persons to be easily reachable. The list has most of features available in find
people page, however it includes some significant and exclusive functions like instant messaging service. Figure (7.5) shows a sample for the followed person.

![Figure 7.5: Followed Person](image.png)

One of the powerful features available in this application is the instant messaging function. The user can communicate directly with any person from the favorite list through instant messaging using Extensible Messaging and Presence Protocol (XMPP).

The user can unfollow any of the person on the favorite list by clicking unfollow icon on the top right corner. The following cypher query run against the graph database to delete the relationship between and the user and a person.

```cypher
match (_persons:PERSON{user_id:"" + User.user_id + @""})-[r]-(n:PERSON) where n.user_id="" + e.CommandArgument.ToString() + @"" delete r
```

7.4.2.7 Browse Organizations

The user of application may be interested to have further communications with any of the registered companies participating in the exhibition. Providing the users with the organizations information and their contact details they need is worthy and helping them to boost the cooperation and increase business opportunities.

Users can add any organization to their favorite list by selecting Follow button. Figure (7.6) demonstrates a sample of organization details. A contact card is shown when the mouse go over the organization. It contains the contact details for two representatives in case the user needs to communicate with decision makers within the organization.
The following method used to get all organizations from the graph database:

```csharp
void DisplayAllCompany()
{
    using (var session = Neo4JConnection.Neo4jDriver.Session())
    {
        var result = session.Run("match (_org:ORGANIZATION) return (_org)";)
        UpdateCompanyList(result);
    }
}
```

The user can communicate with any of the organization representatives through email or instant messaging.

For some organizations which the user added to the favorite list, it shows different icon to indicate that organization is already followed. The method in code snippet (1.6.1) checks whether the organization is added to the favorite list or not, then return true if the organization is one of the favorite.

7.4.2.8 Organizations I am Interesting in

This page displays the organizations which the users added to their favorite list. It contains almost the same features which are available in the browse organization page, in addition to unfollow button to enable the user to remove any organization from the list. Figure (7.7) shows a sample of an organization the user interested in.
The below cypher query is executed when the user unfollow a particular organization. It deletes the relationship between the person and the organization entities.

```cypher
string DeleteRelation = @"match (_persons:PERSON{user_id:" + User.user_id + "})-[r]-(n:ORGANIZATION{orgID:" + e.CommandArgument.ToString() + "}) delete r";
```

The entire list is automatically updated once the user unfollow any organization. Asynchronous JavaScript And XML (AJAX) is used to update specific contents without causing the whole page to be posted back to the hosting server.

### 7.4.2.9 Activity Log

This page shows a list of all activities the user done on the application. It shows the activity type and name along with an icon expressing the activity type. It also shows a star icon of the rated event on the right side of the record. The user can view the review given for a particular event when the mouse go over the icon placed on the right side. Figure (7.8) shows a snap-shot of user activity log.

![Activity Log](image)

*Figure 7.8: Activity Log*

The following cypher query statement return all relationships for a particular person.

```cypher
match (P:PERSON{user_id:" + User.user_id + "})-[relatedTo]-(n) RETURN P.name, Type(relatedTo), relatedTo,n
```
The results are processed to read the relationship properties which distinguish the type of activity, then display the contents to the user in a proper format. The block of code (1.7.1) identify the type of relationship, then fetch the required information accordingly.

7.4.2.10 User Profile

This Page is developed to allow users updating their profiles. As all the fields in this form are marked mandatory, User has to fill all with correct values. The given information is used from the system to make recommendations of events and persons. The method code (1.8.1) is used to update the user information on the graph database.

7.4.2.11 Instant Messaging

This feature is one of the powerful services in the application. It is utilized as a communication channel to communicate people together and foster the connectivity between them. This feature is based on XMPP (Extensible Messaging and Presence Protocol) which emerged from jabber open-source for streaming XML over network (XMPP, 2018).

While XMPP comes with various open technologies, this application uses only the instant messaging and presence technologies from XMPP. The instant messaging functionality enables people to send & receive messages with other persons over the web application.

The user just clicks the messaging button to open a messaging window for any particular person from the friends list, Once the sender write the message and click send button, the recipient will view a new pop-up window loading with the message. Figure (7.9) shows two persons are messaging over the chat window.
The block of code (1.9.1) is executed when the user click send button.

Users can also control their visibility status shown over the messaging window for a particular person. They can choose one of the available statuses (Available, Away, Busy and Ready to chat).

The Code snippet (1.9.2) shows how the presence status options are handled, then sent to the messaging server to share with the other persons.

### 7.5 System Testing

Testing is one of the significant processes throughout the Software Development Life Cycle (SDLC). One of the main principles to test a software as stated by (Meyer, 2008), is trying to make it fail. It is concerned with finding the faults through operating the system or a particular component under certain conditions to observe the results. Although there are some classifications for testing techniques, (Majchrzak, 2012) thinks that wide range of the available testing methods make it even impossible to describe.
As the project is managed through agile methodology, this project followed incremental testing approach to test the completed tasks after each iteration. The feedback which usually include defects or UI issues were carried over the next iteration to be incorporated with the next iteration tasks. By the completion of iterations for a particular sprint which often represents a feature or major function in the system, then this sprint is integrated and tested with the entire system.

Due to the time constraint in this project, the minimal testing was carried out to ensure the system functionalities are working as per the system requirements. However, the application was tested manually to find out and fix any bugs or defects, assure that system quality is meeting with the user expectations, and verify the system requirements have been completed successfully.

### 7.6 Summary

This chapter explained in detail the system development stage. It explored some of the key functions and web pages available in the system. By the completion of system implementation chapter, the project had considered all the mandatory requirements and most of the optional ones which were agreed upon.

However, some of the new requirements have been raised throughout the project development. The new requirements included adding some new functionalities to the system or doing minor adjustments, e.g. creating a home page, showing different user presence for a particular person over the messaging window, and adjusting user interface. By managing this project using agile methodology, all the new requirements had been incorporated successfully within the project tasks.
8 Evaluation

8.1 Overview

A crucial part of this thesis concerns evaluating the performance of application and ensuring the system functionalities are running properly. Studying the system using common evaluation techniques like checklists and questionnaires leads to high quality of research results and supports any further development effectively.

In order to evaluate the system performance and examine the usefulness of functionalities, two types of evaluations were conducted on 11 subjects with different characteristics who represent the potential users of application. They are consisting of 8 males and 3 females, aged above 21 years, working in various fields, and from diverse nationalities.

8.2 Prior the Evaluation

The system environment was setup on a machine with all needed tools, and configured properly to undertake the evaluation stages. The participants had the facilities to access the system either in-place or remotely.

The participants received a brief on the project as well as the objectives beyond the evaluation. They were informed with all instructions needed to be enrolled in this experiment. They signed a consent form to proceed with the evaluation steps.

As part of conducting the evaluation, participants were asked to take part in online preliminary questionnaire containing general information and few questions. The participants were invited by emails to complete the questionnaire using a link to the survey website. The email also contained a brief of the evaluation process and list of tasks (explained in next section) which were required to be accomplished after answering the preliminary questionnaire. The introductory survey aimed to evaluate the
past experience of participants and assess their knowledge on event management applications.

The preliminary questionnaire with the results received from the respondents can be found in Appendix D – Introductory Questionnaire.

8.3 User Evaluation

8.3.1 Task-based Evaluation

This evaluation was conducted by the end of the software development processes on selected functions of the system. The participants have been asked to accomplish certain tasks on the application. The tasks are covering some of the key features in the application to examine its functionalities. This evaluation aimed to assess the system usability, examine the efficiency of main functionalities and improve user experience. Participants received a list of 12 tasks and they have been asked to complete all of them.

The tasks list which were given to the participants to accomplish on the system can be found in Appendix D – Tasks List.

8.3.2 Post-Tasks Questionnaire

This questionnaire was taken after all participants had finished the tasks list which are mentioned in the previous section. It was carried out after the entire system development completed. All the participants were asked to take part in the questionnaire to provide their input. The questionnaire aimed to evaluate the total user satisfaction on the application, assess the familiarity of user interface and appraise the entire system functionalities success.

The questions and responses received from the participants can be found in Appendix D - Post-Tasks Questionnaire.
8.4 Analysis of Evaluation Results

This section analyzes the results of evaluations that were taken by 11 participants. The evaluations were consisted of an introductory questionnaire, Tasks List and Post-Tasks questionnaire. All graph results can be found in Appendix D – Results.

Results of the introductory survey shows that 73% of the participants were Male and 27% were Female. The range of ages sorted as 45% of the participants were between (25 - 34) years, 36% between (35-44) years and 18% between (45-64) years. Most of participants hold Bachelor degree with 55%, while 18% hold Master degree and similar percentage for Diploma, just 9% hold high school degree. The results shows that 64% of the participants had attended events in the past, while the rest 36% did not. Just 36% of the participants have used event management application in the past. The results shows 73% of the participants thinks that creating a new application dedicated for particular exhibition is worthy, just 9% thinks it does not worth, and the remaining are not sure. Majority of the participants representing 91% are using social or professional network applications, and only 9% are not. 73% of participants plan to visit EXPO 2020 – Dubai, while the rest are not sure yet.

The participants were asked to complete a post-task questionnaire after they had finished the tasks list which is mentioned in the previous section. The results of the post-tasks questionnaire about the evaluation of the entire application shows the following results: 91% of the total participants were able to complete all tasks with no issues, while 9% only encountered some difficulties. 9% only were unable to complete the third task which concerns with adding an event to the favorite list. 64% gave rate 4/5 for the user interface of application, 27% for the highest rate, and 9% evaluate the user interface as average. 64% of the participants found that the application was easy to use, 27% saw it as very easy to use, and 9% felt it as normal. 64% found the find events service is good, 27% rated as very good, and 9% saw it as normal. 36% found that the application gives average control to the user to manage the services, and the same percentage found that it gives a good control, and 27% found the application gives full control to the user. All of the participants who represent 100% found that the application is attractive and innovative, and they would recommend to their friends.
9 Conclusion and Future Work

9.1 Conclusion

Dubai chose “Connecting Minds, Creating the Future” as Main theme for EXPO 2020 to emphasize the importance of communication between people. This project was implemented to provide a solution to boost the connectivity between all of the visitors and participants in EXPO 2020 – Dubai. The project undertook the needs of integrating all the key elements of EXPO 2020 (People, Events and Organizations) to combine all in one-stop application.

The solution consisted of a web-based application which enable users to find people, look for interesting events and browse organizations. The application is based on Neo4j graph database where data stored in a native graph structure. The system is empowered with XMPP to allow users communicating instantly with their peers.

The positive results of the evaluation shows that 91% of the participants are familiar with the application and were able to accomplish a set of selected tasks. 100% of the participants thinks the application is attractive and innovative, and they would recommend to their friends.

Finally, the project was able to fulfill all the mandatory requirements and most of the optional requirements within the project time frame. According to the given results and feedback received from users, the project was able to meet the user expectations and address the underlying objectives successfully.
9.2 Reflection on the Project

Throughout the implementation of this project, there were some pitfalls that interrupted the project progress or caused some delays for certain tasks. The project problems and difficulties are summarized in the following points:

- **agsXMPP is not fully supported**: The Software Development Kit (SDK) library of Extensible Messaging and Presence Protocol (XMPP) is open-source and not fully supported. Only few resources are available on their official website for research, insufficient examples and very limited contents.

- **Neo4j Community Edition has basic features**: Although the community edition was adequate to accomplish the project tasks, but some key features and security options were required to improve and underpin the performance.

- **Handling multiple instant messaging windows**: It was a challenge to handle the instant messages received from multiple users simultaneously over the web browsers. Managing and opening multiple pop-up windows once receiving a new message required writing advanced client script code to handle their events.

- **Learning new technologies**: Within the tight time frame of this project, some of the difficult tasks during the system implementation were required from the author to learn new technologies, acquire new technical skills and investigate in advanced areas in order to accomplish these tasks. Definitely, it was a great opportunity and helped the author to broaden his own knowledge and skills.

9.3 Future Work

Although the developed application is functioning well, there are further development opportunities which could improve the performance, add more significant features and
enhance the system reliability. Below are some recommendations for future development on the system:

- Developing this application for mobile platforms like android, iOS and many other operating systems. As a result, User will have handy solution to manage all the activities of exhibition. The number of users will be increased rapidly and the system will be more accessible. According to (Sandra & Nataša, 2014), 1.2 billion persons were using mobile applications till end of 2012, with expectations to reach 4.4 billion users by end of 2017. Moreover, Developing Mobile Apps will underpin the instant messaging feature and facilitate communication between users.

- Furthermore, allowing users to register & login with their social network accounts (e.g. Facebook accounts) will facilitate the registration process on the system, and enable users to login at their convenience. Reading user information from their social network profiles, and bring their profile pictures into the system will empower the application with accurate data. It even could be utilized to recommend some of the events or persons matching their preferences.

- The application could be developed to show news feed, a list of all posts shared by user’s peers or the organizations they are interesting in, or any raised updates on the events they are following.

- Adding some features to the application like adding pagination to event lists, people lists and organization lists will reduce the time of page loading especially when the application come into action with millions of users.

- Allowing users to search the contents of application (e.g. events, persons), and showing the results that match their key words will be added value.

- The system could be integrated with any mailing server or SMS gateway to notify users on all activities either by emails or SMSs. Users would get notifications on all updates happened on events, or even when they received instant messages from their friends.

- The system could be developed to enable users to view the current presence status of persons they are following all within a single list.
References

- Amanda Lenhart, (2012), Teens, Smartphones & Texting, PEW RESEARCH CENTER, [accessed: 2 August 2018], available online: http://www.pewinternet.org/2012/03/19/teens-smartphones-texting/
- Chris Kemper, (2015), Beginning Neo4j by Chris Kemper, Apress.


Grady, J.O., (2014), System requirements analysis / Jeffrey O. Grady. 2nd ed.

Ian Yeoman, Martin Robertson, Jane Ali-Knight, Siobhan Drummond and Una McMahon-Beattie, (2004), Festival and Events Management: An International Arts and Culture Perspective.

Isaias, P. & Issa, T., (2015), High Level Models and Methodologies for Information Systems by Pedro Isaias, Tomayess Issa.


Katie Sawyer, (2018), The 6 Top Event Promotion Websites You Should Be Using, [accessed: 5 August 2018], available online: https://www.eventbrite.com/blog/event-promotion-websites-ds00/


• Matthew MacDonald, (2012), Beginning ASP.NET 4.5 in VB, Apress, Berkeley, CA
• Neo4j, (2018), Using Neo4j from .NET, [accessed: 21 July 2018], available online: https://neo4j.com/developer/dotnet/
• Robinson I., Jim Webber, Emil Eifrem, (2015), Graph Databases: New Opportunities for Connected Data, O'Reilly, Sebastopol, California.
• Roopendra Vishwakarma, (2017), The Different Types of NoSQL Databases, [accessed: 25 Feb 2018], available online: http://opensourceforu.com/2017/05/different-types-nosql-databases/
• Sandra Medić & Nataša Pavlović, (2014), Mobile Technologies in Museum Exhibitions, TURIZAM Volume 18, Issue 4, pp.166-174
References

## Appendix A

### 1. Project Plan

<table>
<thead>
<tr>
<th>Task Name</th>
<th>Duration</th>
<th>Start</th>
<th>Finish</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Requirements Analysis</strong></td>
<td>20 days</td>
<td>Sat 5/05/18</td>
<td>Sat 2/06/18</td>
</tr>
<tr>
<td>Requirements analysis</td>
<td>10 days</td>
<td>Sat 5/05/18</td>
<td>Thu 17/05/18</td>
</tr>
<tr>
<td>Review Requirements</td>
<td>2 days</td>
<td>Fri 18/05/18</td>
<td>Sun 20/05/18</td>
</tr>
<tr>
<td><strong>UML Design</strong></td>
<td>7 days</td>
<td>Mon 21/05/18</td>
<td>Tue 29/05/18</td>
</tr>
<tr>
<td><strong>System Usability Design</strong></td>
<td>4 days</td>
<td>Wed 30/05/18</td>
<td>Sat 2/06/18</td>
</tr>
<tr>
<td><strong>Sprint 1 (IDE &amp; Database Installation)</strong></td>
<td>4 days</td>
<td>Sun 3/06/18</td>
<td>Wed 6/06/18</td>
</tr>
<tr>
<td><strong>Iteration 1</strong></td>
<td>4 days</td>
<td>Sun 3/06/18</td>
<td>Wed 6/06/18</td>
</tr>
<tr>
<td>Install &amp; configure ASP.NET</td>
<td>1 day</td>
<td>Sun 3/06/18</td>
<td>Sun 3/06/18</td>
</tr>
<tr>
<td>Download &amp; Install Neo4j Graph Database</td>
<td>1 day</td>
<td>Mon 4/06/18</td>
<td>Mon 4/06/18</td>
</tr>
<tr>
<td>Configure &amp; test connection with graph DB</td>
<td>1 day</td>
<td>Tue 5/06/18</td>
<td>Tue 5/06/18</td>
</tr>
<tr>
<td><strong>Sprint 1 Completion</strong></td>
<td>0 days</td>
<td>Wed 6/06/18</td>
<td>Wed 6/06/18</td>
</tr>
<tr>
<td><strong>Sprint 2 (Develop Registration &amp; Login Pages )</strong></td>
<td>13 days</td>
<td>Fri 8/06/18</td>
<td>Mon 25/06/18</td>
</tr>
<tr>
<td><strong>Iteration 1</strong></td>
<td>7 days</td>
<td>Fri 8/06/18</td>
<td>Sun 17/06/18</td>
</tr>
<tr>
<td>Create the Master Page of the application</td>
<td>1 day</td>
<td>Fri 8/06/18</td>
<td>Fri 8/06/18</td>
</tr>
<tr>
<td>Create Application Side Menu</td>
<td>1 day</td>
<td>Sat 9/06/18</td>
<td>Sat 9/06/18</td>
</tr>
<tr>
<td>Design User Registration Form UI</td>
<td>1 day</td>
<td>Sun 10/06/18</td>
<td>Sun 10/06/18</td>
</tr>
<tr>
<td>Develop User Registration Form</td>
<td>2 days</td>
<td>Mon 11/06/18</td>
<td>Tue 12/06/18</td>
</tr>
<tr>
<td>Design User Login Page UI</td>
<td>1 day</td>
<td>Wed 13/06/18</td>
<td>Wed 13/06/18</td>
</tr>
<tr>
<td>Develop User Login Page</td>
<td>2 days</td>
<td>Thu 14/06/18</td>
<td>Fri 15/06/18</td>
</tr>
<tr>
<td>Test User Registration &amp; Login Pages</td>
<td>2 days</td>
<td>Sat 16/06/18</td>
<td>Sun 17/06/18</td>
</tr>
<tr>
<td><strong>Iteration 2</strong></td>
<td>7 days</td>
<td>Sun 17/06/18</td>
<td>Sat 23/06/18</td>
</tr>
<tr>
<td>Fix All bugs found</td>
<td>2 days</td>
<td>Sun 17/06/18</td>
<td>Mon 18/06/18</td>
</tr>
<tr>
<td>Enhance Side Menu Style</td>
<td>1 day</td>
<td>Tue 19/06/18</td>
<td>Tue 19/06/18</td>
</tr>
<tr>
<td>Enhance Pages CSS style</td>
<td>2 days</td>
<td>Wed 20/06/18</td>
<td>Thu 21/06/18</td>
</tr>
<tr>
<td>Integrate the new feature with application</td>
<td>1 day</td>
<td>Fri 22/06/18</td>
<td>Fri 22/06/18</td>
</tr>
<tr>
<td>Add fields validation in the Registration &amp; Login pages</td>
<td>1 day</td>
<td>Sat 23/06/18</td>
<td>Sat 23/06/18</td>
</tr>
<tr>
<td>Iteration 3</td>
<td>2 days</td>
<td>Sun 24/06/18</td>
<td>Mon 25/06/18</td>
</tr>
<tr>
<td>-------------</td>
<td>--------</td>
<td>--------------</td>
<td>--------------</td>
</tr>
<tr>
<td>Design User Profile Page</td>
<td>1 day</td>
<td>Sun 24/06/18</td>
<td>Sun 24/06/18</td>
</tr>
<tr>
<td>Develop User Profile Page</td>
<td>1 day</td>
<td>Mon 25/06/18</td>
<td>Mon 25/06/18</td>
</tr>
<tr>
<td>Design Splash Screen</td>
<td>1 day</td>
<td>Mon 25/06/18</td>
<td>Mon 25/06/18</td>
</tr>
<tr>
<td>Test User Creation on XMPP Server</td>
<td>1 day</td>
<td>Mon 25/06/18</td>
<td>Mon 25/06/18</td>
</tr>
<tr>
<td>Testing Functionalities</td>
<td>1 day</td>
<td>Mon 25/06/18</td>
<td>Mon 25/06/18</td>
</tr>
<tr>
<td>sprint 2 Completion</td>
<td>0 days</td>
<td>Mon 25/06/18</td>
<td>Mon 25/06/18</td>
</tr>
<tr>
<td>Sprint 3 (Browse Events &amp; Favored Events Pages)</td>
<td>10 days</td>
<td>Tue 26/06/18</td>
<td>Sat 7/07/18</td>
</tr>
<tr>
<td>Iteration 1</td>
<td>5 days</td>
<td>Tue 26/06/18</td>
<td>Sun 1/07/18</td>
</tr>
<tr>
<td>Add a bar on the top of master page</td>
<td>1 day</td>
<td>Tue 26/06/18</td>
<td>Tue 26/06/18</td>
</tr>
<tr>
<td>Design Find Events Page</td>
<td>1 day</td>
<td>Wed 27/06/18</td>
<td>Wed 27/06/18</td>
</tr>
<tr>
<td>Develop Find Events Page</td>
<td>2 days</td>
<td>Thu 28/06/18</td>
<td>Fri 29/06/18</td>
</tr>
<tr>
<td>Add Events Filtering options on the page</td>
<td>1 day</td>
<td>Sat 30/06/18</td>
<td>Sat 30/06/18</td>
</tr>
<tr>
<td>Find events of friends and up to level 5</td>
<td>1 day</td>
<td>Sun 1/07/18</td>
<td>Sun 1/07/18</td>
</tr>
<tr>
<td>Iteration 2</td>
<td>2 days</td>
<td>Sun 1/07/18</td>
<td>Mon 2/07/18</td>
</tr>
<tr>
<td>Add Filter details on top of list</td>
<td>1 day</td>
<td>Sun 1/07/18</td>
<td>Sun 1/07/18</td>
</tr>
<tr>
<td>Add rate &amp; Review features to favored events</td>
<td>1 day</td>
<td>Sun 1/07/18</td>
<td>Sun 1/07/18</td>
</tr>
<tr>
<td>Show Friends Interesting on a particular event</td>
<td>1 day</td>
<td>Mon 2/07/18</td>
<td>Mon 2/07/18</td>
</tr>
<tr>
<td>Design Favored Events Page</td>
<td>1 day</td>
<td>Mon 2/07/18</td>
<td>Mon 2/07/18</td>
</tr>
<tr>
<td>Develop Favored Events Page</td>
<td>1 day</td>
<td>Mon 2/07/18</td>
<td>Mon 2/07/18</td>
</tr>
<tr>
<td>Testing Functionalities</td>
<td>1 day</td>
<td>Mon 2/07/18</td>
<td>Mon 2/07/18</td>
</tr>
<tr>
<td>Sprint 3 Completion</td>
<td>0 days</td>
<td>Mon 2/07/18</td>
<td>Mon 2/07/18</td>
</tr>
<tr>
<td>Sprint 4 (Find People &amp; Favored People Pages)</td>
<td>11 days</td>
<td>Tue 3/07/18</td>
<td>Tue 17/07/18</td>
</tr>
<tr>
<td>Iteration 1</td>
<td>3 days</td>
<td>Tue 3/07/18</td>
<td>Thu 5/07/18</td>
</tr>
<tr>
<td>Design Find People Page</td>
<td>1 day</td>
<td>Tue 3/07/18</td>
<td>Tue 3/07/18</td>
</tr>
<tr>
<td>Develop Find People Page</td>
<td>1 day</td>
<td>Wed 4/07/18</td>
<td>Wed 4/07/18</td>
</tr>
<tr>
<td>Add Filter by field and organization</td>
<td>1 day</td>
<td>Thu 5/07/18</td>
<td>Thu 5/07/18</td>
</tr>
<tr>
<td>Iteration 2</td>
<td>7 days</td>
<td>Wed 4/07/18</td>
<td>Thu 12/07/18</td>
</tr>
<tr>
<td>Design Followed People Page</td>
<td>2 days</td>
<td>Wed 4/07/18</td>
<td>Thu 5/07/18</td>
</tr>
<tr>
<td>Develop Followed People Page</td>
<td>3 days</td>
<td>Fri 6/07/18</td>
<td>Tue 10/07/18</td>
</tr>
<tr>
<td>Add filter by field and organization</td>
<td>1 day</td>
<td>Wed 11/07/18</td>
<td>Wed 11/07/18</td>
</tr>
<tr>
<td>Find People by Connection Level</td>
<td>1 day</td>
<td>Thu 12/07/18</td>
<td>Thu 12/07/18</td>
</tr>
<tr>
<td>Iteration 3</td>
<td>3 days</td>
<td>Fri 13/07/18</td>
<td>Tue 17/07/18</td>
</tr>
<tr>
<td>Enhance Pages CSS style</td>
<td>2 days</td>
<td>Fri 13/07/18</td>
<td>Sat 14/07/18</td>
</tr>
<tr>
<td>Add Icon Legend for both pages</td>
<td>1 day</td>
<td>Sun 15/07/18</td>
<td>Sun 15/07/18</td>
</tr>
<tr>
<td>Add AJAX for both pages</td>
<td>2 days</td>
<td>Mon 16/07/18</td>
<td>Tue 17/07/18</td>
</tr>
<tr>
<td>Testing Functionalities</td>
<td>1 day</td>
<td>Tue 17/07/18</td>
<td>Tue 17/07/18</td>
</tr>
<tr>
<td>Sprint 4 Completion</td>
<td>0 days</td>
<td>Tue 17/07/18</td>
<td>Tue 17/07/18</td>
</tr>
<tr>
<td>Sprint 5 (Browse Organizations &amp; Follow Organizations Pages)</td>
<td>7 days</td>
<td>Wed 18/07/18</td>
<td>Thu 26/07/18</td>
</tr>
<tr>
<td>Iteration 1</td>
<td>5 days</td>
<td>Wed 18/07/18</td>
<td>Tue 24/07/18</td>
</tr>
<tr>
<td><strong>Appendix A</strong></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>----------------</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>Design Browse Organizations Page</strong></td>
<td>1 day</td>
<td>Wed 18/07/18</td>
<td>Wed 18/07/18</td>
</tr>
<tr>
<td><strong>Develop Browse Organizations Page</strong></td>
<td>2 days</td>
<td>Thu 19/07/18</td>
<td>Fri 20/07/18</td>
</tr>
<tr>
<td><strong>Design Followed Organizations Page</strong></td>
<td>1 day</td>
<td>Sat 21/07/18</td>
<td>Sat 21/07/18</td>
</tr>
<tr>
<td><strong>Develop Followed Organizations Page</strong></td>
<td>1 day</td>
<td>Sun 22/07/18</td>
<td>Sun 22/07/18</td>
</tr>
<tr>
<td><strong>Design Activity Log Page</strong></td>
<td>1 day</td>
<td>Mon 23/07/18</td>
<td>Mon 23/07/18</td>
</tr>
<tr>
<td><strong>Develop Activity Log Page</strong></td>
<td>1 day</td>
<td>Tue 24/07/18</td>
<td>Tue 24/07/18</td>
</tr>
<tr>
<td><strong>Iteration 2</strong></td>
<td><strong>2 days</strong></td>
<td><strong>Wed 25/07/18</strong></td>
<td><strong>Thu 26/07/18</strong></td>
</tr>
<tr>
<td>Add Ajax for both pages</td>
<td>1 day</td>
<td>Wed 25/07/18</td>
<td>Wed 25/07/18</td>
</tr>
<tr>
<td>Add filter by field and country</td>
<td>1 day</td>
<td>Wed 25/07/18</td>
<td>Wed 25/07/18</td>
</tr>
<tr>
<td>Enhancing UI</td>
<td>1 day</td>
<td>Thu 26/07/18</td>
<td>Thu 26/07/18</td>
</tr>
<tr>
<td>Testing Functionalities</td>
<td>1 day</td>
<td>Thu 26/07/18</td>
<td>Thu 26/07/18</td>
</tr>
<tr>
<td>Sprint 5 Completion</td>
<td>0 days</td>
<td>Thu 26/07/18</td>
<td>Thu 26/07/18</td>
</tr>
<tr>
<td><strong>Sprint 6 (Instant Messaging &amp; Home Page)</strong></td>
<td><strong>3 days</strong></td>
<td><strong>Fri 27/07/18</strong></td>
<td><strong>Tue 31/07/18</strong></td>
</tr>
<tr>
<td><strong>Iteration 1</strong></td>
<td><strong>2 days</strong></td>
<td><strong>Fri 27/07/18</strong></td>
<td><strong>Mon 30/07/18</strong></td>
</tr>
<tr>
<td>Design Messaging Window</td>
<td>1 day</td>
<td>Fri 27/07/18</td>
<td>Fri 27/07/18</td>
</tr>
<tr>
<td>Develop Messaging Window</td>
<td>1 day</td>
<td>Sat 28/07/18</td>
<td>Sat 28/07/18</td>
</tr>
<tr>
<td>Design Chat Presence</td>
<td>1 day</td>
<td>Sun 29/07/18</td>
<td>Sun 29/07/18</td>
</tr>
<tr>
<td>Develop Chat Presence</td>
<td>1 day</td>
<td>Sun 29/07/18</td>
<td>Sun 29/07/18</td>
</tr>
<tr>
<td>Design Home Page</td>
<td>1 day</td>
<td>Sun 29/07/18</td>
<td>Sun 29/07/18</td>
</tr>
<tr>
<td><strong>Iteration 2</strong></td>
<td><strong>2 days</strong></td>
<td><strong>Mon 30/07/18</strong></td>
<td><strong>Tue 31/07/18</strong></td>
</tr>
<tr>
<td>Adding Ajax and timer on chat window</td>
<td>1 day</td>
<td>Mon 30/07/18</td>
<td>Mon 30/07/18</td>
</tr>
<tr>
<td>Enhancing UI</td>
<td>1 day</td>
<td>Mon 30/07/18</td>
<td>Mon 30/07/18</td>
</tr>
<tr>
<td>Testing Functionalities</td>
<td>1 day</td>
<td>Tue 31/07/18</td>
<td>Tue 31/07/18</td>
</tr>
<tr>
<td>Sprint 6 Completion</td>
<td>0 day</td>
<td>Tue 31/07/18</td>
<td>Tue 31/07/18</td>
</tr>
<tr>
<td><strong>Testing</strong></td>
<td><strong>1 day</strong></td>
<td><strong>Wed 1/08/18</strong></td>
<td><strong>Wed 1/08/18</strong></td>
</tr>
<tr>
<td>Testing All Application Functionalities</td>
<td>1 day</td>
<td>Wed 1/08/18</td>
<td>Wed 1/08/18</td>
</tr>
<tr>
<td>Evaluation</td>
<td>2 days</td>
<td>Thu 2/08/18</td>
<td>Fri 3/08/18</td>
</tr>
<tr>
<td>Dissertation Write up</td>
<td>10 days</td>
<td>Sat 4/08/18</td>
<td>Thu 16/08/18</td>
</tr>
<tr>
<td>Dissertation Submission</td>
<td>0 days</td>
<td>Thu 16/08/18</td>
<td>Thu 16/08/18</td>
</tr>
<tr>
<td>Poster Design</td>
<td>12 days</td>
<td>Fri 17/08/18</td>
<td>Tue 28/08/18</td>
</tr>
<tr>
<td>Poster Submission</td>
<td>0 days</td>
<td>Tue 28/08/18</td>
<td>Tue 28/08/18</td>
</tr>
<tr>
<td>Poster Presentation</td>
<td>1 day</td>
<td>Thu 30/08/18</td>
<td>Thu 30/08/18</td>
</tr>
</tbody>
</table>
2. Gantt Chart
3. Risk Management Plan

<table>
<thead>
<tr>
<th>Risk Identification</th>
<th>Risk Probability (1-10)</th>
<th>Risk Impact</th>
<th>Resolution</th>
</tr>
</thead>
<tbody>
<tr>
<td>Database configuration &amp; design requires additional time</td>
<td>5</td>
<td>Medium</td>
<td>Allocate more time for these tasks</td>
</tr>
<tr>
<td>Application UI Design not compatible with other devices</td>
<td>4</td>
<td>Low</td>
<td>Make sure the pages are tested &amp; displayed properly in more than one browser</td>
</tr>
<tr>
<td>Hardware failure and Losing Data</td>
<td>2</td>
<td>High</td>
<td>Consider additional hardware for any unexpected issue. Keep additional backup of all files and databases.</td>
</tr>
<tr>
<td>Further changes required by key stakeholders</td>
<td>8</td>
<td>High</td>
<td>Adapt the project plan to accommodate the new change requests without affecting the project scope or quality</td>
</tr>
<tr>
<td>Messaging server is not reachable or accessible from IDE</td>
<td>5</td>
<td>Medium</td>
<td>Find alternative servers or try another version</td>
</tr>
<tr>
<td>User interface design consumes much time from the project schedule</td>
<td>4</td>
<td>Medium</td>
<td>Allocate sufficient time in project schedule for UI design</td>
</tr>
<tr>
<td>Difficulties to handle multiple instant messages simultaneously</td>
<td>7</td>
<td>High</td>
<td>Set messages controller at a central page to manage received messages</td>
</tr>
</tbody>
</table>
Appendix B

1. Screen-Shots

1.1 Splash Screen

Background Image retrieved from: https://dumielauxepices.net/wallpaper-2348565
1.2 User Registration


1.3 User Login

1.4 Home

1.5 Find People
1.6 People I am Following

1.7 Find Events
1.8 Events I Like

1.9 Browse Organizations
1.10 Organizations I am Interesting in

1.11 Activity Log
1.12 User Profile

1.13 Instant Messaging
Appendix C

1. Code-Snippets

1.1 User Registration

1.1.1 Create Person

```csharp
public void CreatePerson()
{
    string strCreate = "Create (" + user_id + ":PERSON {user_id:" + user_id + ",password:" + password + ",name:" + fname + ",lname:" + lname + ",country:" + country + ",industry:" + industry + ",designation:" + designation + ",organization:" + organization + ",gender:" + gender + ",age:" + age + ",contact:" + contact + ",email:" + email + ");

    using (var session = Neo4JConnection.Neo4jDriver.Session())
    {
        var result = session.Run(strCreate);
    }
}
```

1.1.2 Create user on openfire server

```csharp
xmpp.AutoResolveConnectServer = true;
xmpp.AutoRoster = true;
xmpp.UseSSL = false;
xmpp = new XmppClientConnection();
xmpp.Server = ServerName;
xmpp.Port = Convert.ToInt16(Port);
xmpp.Username = txtUserID.Text;
xmpp.Password = txtPassword.Text;
xmpp.OnRegistered += delegate(object obj) { XmppConnection_OnUserReg(xmpp); };
xmpp.OnRegisterError += new XmppElementHandler(ElementXmppConnection_OnUserRegError);
xmpp.OnBinded += delegate(object obj) { OnUser_Authenticated(xmpp); };
xmpp.OnAuthError += new XmppElementHandler(xmpp_OnUser_authen);
xmpp.RegisterAccount = true;
xmpp.Open();
Session["Xmpp"] = xmpp;
while (RegStatus == 0)
{
    System.Threading.Thread.Sleep(1000);
}
```
Appendix C

1.2 User Login

1.2.1 Login Method

```csharp
byte[] password_bytes = System.Text.UnicodeEncoding.UTF8.GetBytes(txtPassword.Text);
string password = Convert.ToBase64String(password_bytes);
string strCheckUserAccount = @"match (_person:PERSON) where _person.user_id="" + txtName.Text + @"" and _person.password="" + password + @"" return _person";
using (var session = Neo4JConnection.Neo4jDriver.Session())
{
    var users = new List<Person>();
    var result = session.Run(strCheckUserAccount);
    var UserRecord = result.Single();
    if (UserRecord.Keys.Count > 0)
    {
        var UserProfile = (INode)UserRecord[0];
        Person person = new Person();
        person.user_id = UserProfile["user_id"].As<string>();
        person.password = UserProfile["password"].As<string>();
        person.fname = UserProfile["fname"].As<string>();
        person.lname = UserProfile["lname"].As<string>();
        person.designation = UserProfile["designation"].As<string>();
        person.organization = UserProfile["organization"].As<string>();
        person.country = UserProfile["country"].As<string>();
        person.industry = UserProfile["industry"].As<string>();
        person.gender = UserProfile["gender"].As<string>();
        person.age = UserProfile["age"].As<int>();
        person.contact = UserProfile["contact"].As<string>();
        person.email = UserProfile["email"].As<string>();
        Session["Person"] = person; }
```

1.2.2 Authentication on XMPP server

```csharp
xmpp.Server = ServerName;
xmpp.Port = Convert.ToInt16(System.Configuration.ConfigurationManager.AppSettings["Port"]);

xmpp.Username = txtName.Text;
xmpp.Password = txtPassword.Text;
xmpp.AutoPresence = true;
xmpp.UseSSL = false;
xmpp.AutoResolveConnectServer = true;
xmpp.AutoRoster = true;
xmpp.OnAuthError += new XmppElementHandler(xmpp_On_userauth);
xmpp.Open();
```
1.3 Find Events

1.3.1 Filter Events

```csharp
if (e.CommandName == "FieldSearch")
{
    lblFilterName.Text = e.CommandArgument.ToString();
    string strSaveEvent = @"match (_events:EVENT) where _events.field CONTAINS ("" +
    e.CommandArgument + @""") return _events";
    using (var session = Neo4JConnection.Neo4jDriver.Session())
    {
        var result = session.Run(strSaveEvent);
        UpdateEventList(result);
    }
}
if (e.CommandName == "MetaTagSearch")
{
    lblFilterName.Text = "#" + e.CommandArgument.ToString();
    string strSaveEvent = @"match (_events:EVENT) where _events.title CONTAINS ("" +
    e.CommandArgument + @""") or _events.description CONTAINS ("" +
    e.CommandArgument + @""") return _events";
    using (var session = Neo4JConnection.Neo4jDriver.Session())
    {
        var result = session.Run(strSaveEvent);
        UpdateEventList(result);
    }
}
```

1.3.2 Get Average Rate for Event

```csharp
public string GetAverageRate(string EventID)
{
    string strGetAvgRate = @"match (:PERSON)-[_Relation:Rate]-(
    _events{event_id:"" + EventID +
    @""}) return _Relation.Star";
    int TotalRate = 0;
    double RateAvg = 0;
    using (var session = Neo4JConnection.Neo4jDriver.Session())
    {
        var result = session.Run(strGetAvgRate);
        var EventRates = result.ToList();
        if (EventRates.Count > 0)
        {
            foreach (IRecord Rate in EventRates)
            {
                TotalRate += Convert.ToInt16(((List<object>)Rate["_Relation.Star"])[0]);
            }
            RateAvg = (double)TotalRate / EventRates.Count;
        }
        return RateAvg.ToString();
    }
}
```
1.4 Saved Event

1.4.1 Delete a relationship between event and person objects

```csharp
using (var session = Neo4JConnection.Neo4jDriver.Session())
{
    var result = session.Run(DeleteRelation);
}
```

1.4.2 Create a relationship between event and person objects

```csharp
string strCreate_New_relation_Reviewing = @"match (P:PERSON{user_id:"" + User.user_id + @""}),(_event:EVENT{event_id:"" + event_ID + @""}) create (P)[[:Review{Comment:["" + CommentValue + @""}]->(_event);"

string strCreate_New_relation_Rating= @"match (P:PERSON{user_id:"" + User.user_id + @""}),(_event:EVENT{event_id:"" + event_ID + @""}) create (P)[[:Rate{Star:["" + e.Value + "]]->(_event);"
```

1.5 Find People

1.5.1 Return common events between 2 persons

```csharp
List<Event> Get_Common_Events(string PersonID)
{
    if (Session["Person"] != null)
    {
        Person User = (Person)Session["Person"];
        var events = new List<Event>();
        using (var session = Neo4JConnection.Neo4jDriver.Session())
        {
            string strSelectPersons_FavoredEvents = @"match (_persons:PERSON{user_id:"" + User.user_id + @""})-[[:Saved]-(_events:EVENT)]-[[:Saved]-{p:PERSON(user_id:"" + PersonID + @""}) return distinct _events";
            var result = session.Run(strSelectPersons_FavoredEvents);
            foreach (var record in result)
            {
                var node = record["_events"].As<INode>();
                events.Add(new Event
                {
                    event_id = node["event_id"].As<string>(),
                    title = node["title"].As<string>(),
                    desc = node["description"].As<string>(),
                    start_date_str = node["start_date_str"].As<DateTime>().ToString("MMM d"),
                    end_date_str = node["end_date_str"].As<string>(),
                    location = node["location"].As<string>(),
                    organizer = node["organizer"].As<string>(),
                    field = node["field"].As<string>().TrimEnd().Split(' '),
                    category = node["category"].As<string>(),
                    cost = node["cost"].As<double>(),
                    metatags = node["metatags"].As<string>().Split(' '),
                });
            }
        }
    }
}
```
1.5.2 Return common friends between 2 persons

```csharp
List<Person> Get_Friends_of_Person(string PersonID)
{
    if (Session["Person"] != null)
    {
        Person User = (Person)Session["Person"];
        var persons = new List<Person>();
        using (var session = Neo4jConnection.Neo4jDriver.Session())
        {
            var result = session.Run("match (n:PERSON{user_id:" + User.user_id + "} -[:Knows]-\(Friends\)-[:Knows]-\(p:PERSON(user_id:" + PersonID + ") return DISTINCT Friends limit 5")
            foreach (var record in result)
            {
                var node = record["Friends"].As<INode>();
                persons.Add(new Person
                {
                    user_id = node["user_id"].As<string>(),
                    name = node["name"].As<string>(),
                    designation = node["designation"].As<string>(),
                    country = node["country"].As<string>(),
                    industry = node["industry"].As<string>(),
                    gender = node["gender"].As<string>(),
                    contact = node["contact"].As<string>(),
                    email = node["email"].As<string>(),
                    age = node["age"].As<int>(),
                });
            }
            return persons;
        }
    }
    else
    {
        return null;
    }
}
```

1.6 Browse Organizations

1.6.1 Check Interesting organizations

```csharp
bool CheckInterestedCompanies(string Company_ID)
{
    bool FlagValue = false;
    if (Session["Person"] != null)
    {
```
Person User = (Person)Session["Person"]; string strCheckFriends = @"match (_persons:PERSON{user_id:"\"" + User.user_id + "\""})-[:Interest]-(n:ORGANIZATION{orgID:"\"" + Company_ID + "\""}) return n";

using (var session = Neo4jConnection.Neo4jDriver.Session())
{
    var result = session.Run(strCheckFriends);
    foreach (var record in result)
    {
        FlagValue = true;
    }
}
return FlagValue;

1.7 Activity log

1.7.1 Return the type of Relationship

foreach (var record in y)
{
    UserActivity _userActivity = new UserActivity();
    _userActivity.ActivityName = record[0].As<string>();
    _userActivity.ActivityTypeRelatedTo = record[1].As<string>();
    var relation = record[2].As<IRelationship>();
    var node = record[3].As<INode>();
    if (relation.Type == "Rate")
    {
        _userActivity.ActivityRelatedTo = ((List<object>)(relation["Star"]))[0].ToString();
        _userActivity.ActivityDesc = node.Properties["title"].ToString();
    }
    else if (relation.Type == "Saved")
    {
        _userActivity.ActivityDesc = node.Properties["title"].ToString();
    }
    else if (relation.Type == "Knows")
    {
        _userActivity.ActivityDesc = node.Properties["fname"].ToString() + " " + node.Properties["lname"].ToString();
    }
    else if (relation.Type == "Review")
    {
        _userActivity.ActivityDesc = node.Properties["title"].ToString();
        _userActivity.ActivityRelatedTo = ((List<object>)(relation["Comment"]))[0].ToString();
    }
}
1.8 User Profile

1.8.1 Update Person Information

```csharp
public void UpdatePerson()
{
    string strUpdate = @"match(_user:PERSON {user_id:"
        + user_id + @""}) set ";
    if (password != "")
    {
        strUpdate += @"_user.password="" + password + @"";
    }
    strUpdate += @"_user.name="" + fname + " " + lname + @"",_user.fname="" + fname + @"",_user.lname="" + lname + @"",_user.country="" + country + @"",_user.industry="" + industry + @"",_user.designation="" + designation + @"",_user.organization="" + organization + @"",_user.gender="" + gender + @"",_user.age="" + age + @"",_user.contact="" + contact + @"",_user.email="" + email + @""");
    using (var session = Neo4JConnection.Neo4jDriver.Session())
    {
        var result = session.Run(strUpdate);
    }
}
```

1.9 Instant Messaging

1.9.1 Send Message

```csharp
if (txtChatBox.Text != "")
{
    Message msg = new Message(new Jid(UserID), MessageType.chat, txtChatBox.Text);
    msg.Subject = System.Guid.NewGuid().ToString();
    xmpp.Send(msg);
    txtChatBox.Text = "";
}
```

1.9.2 Update Presence

```csharp
Presence userPresence = new Presence();
if (ddlStatus.SelectedItem.Value == "NONE")
{
    userPresence.Show = ShowType.NONE;
}
else if (ddlStatus.SelectedItem.Value == "away")
{
    userPresence.Show = ShowType.away;
}
else if (ddlStatus.SelectedItem.Value == "dnd")
{
    userPresence.Show = ShowType.dnd;
}
else if (ddlStatus.SelectedItem.Value == "chat")
{
    userPresence.Show = ShowType.chat;
}
xmpp.Send(new Presence { Type = PresenceType.available, To = new Jid(UserID), From = xmpp.MyJID, Show = userPresence.Show });
Appendix D

1. Introductory Questionnaire

<table>
<thead>
<tr>
<th>Question 1</th>
<th>IN WHICH AGE RANGE ARE YOU?</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Answers</strong></td>
<td></td>
</tr>
<tr>
<td>19-24</td>
<td>0%</td>
</tr>
<tr>
<td>25-34</td>
<td>45.45%</td>
</tr>
<tr>
<td>35-44</td>
<td>36.36%</td>
</tr>
<tr>
<td>45-64</td>
<td>18.18%</td>
</tr>
<tr>
<td>65+</td>
<td>0%</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Question 2</th>
<th>WHAT IS YOUR GENDER IDENTITY?</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Answers</strong></td>
<td></td>
</tr>
<tr>
<td>Male</td>
<td>72.73%</td>
</tr>
<tr>
<td>Female</td>
<td>27.27%</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Question 3</th>
<th>WHAT IS YOUR HIGHEST DEGREE?</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Answers</strong></td>
<td></td>
</tr>
<tr>
<td>High School</td>
<td>9.09%</td>
</tr>
<tr>
<td>Diploma</td>
<td>18.18%</td>
</tr>
<tr>
<td>Bachelor</td>
<td>54.55%</td>
</tr>
<tr>
<td>Master</td>
<td>18.18%</td>
</tr>
<tr>
<td>Doctorate</td>
<td>0%</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Question 4</th>
<th>DID YOU EVER PARTICIPATE IN ANY KIND OF EVENTS OR EXHIBITIONS?</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Answers</strong></td>
<td></td>
</tr>
<tr>
<td>Yes</td>
<td>63.64%</td>
</tr>
<tr>
<td>No</td>
<td>36.36%</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Question 5</th>
<th>DID YOU USE ANY EVENT MANAGEMENT APPLICATION IN THE PAST?</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Answers</strong></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Appendix D

<table>
<thead>
<tr>
<th>Question 6</th>
<th>Answers</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>DO YOU THINK CREATING NEW APPLICATION DEDICATED FOR MANAGING A PARTICULAR EXHIBITION IS WORTHY?</td>
<td></td>
</tr>
<tr>
<td>Answers</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Yes</td>
<td>36.36%</td>
<td></td>
</tr>
<tr>
<td>No</td>
<td>63.64%</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Question 7</th>
<th>Answers</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>DO YOU USE ANY SOCIAL/PROFESSIONAL NETWORK APPLICATIONS?</td>
<td></td>
</tr>
<tr>
<td>Answers</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Yes</td>
<td>72.73%</td>
<td></td>
</tr>
<tr>
<td>No</td>
<td>9.09%</td>
<td></td>
</tr>
<tr>
<td>Not Sure</td>
<td>18.18%</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Question 8</th>
<th>Answers</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>DO YOU PLAN TO VISIT EXPO 2020 - DUBAI?</td>
<td></td>
</tr>
<tr>
<td>Answers</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Yes</td>
<td>72.73%</td>
<td></td>
</tr>
<tr>
<td>No</td>
<td>0%</td>
<td></td>
</tr>
<tr>
<td>Not Sure</td>
<td>27.27%</td>
<td></td>
</tr>
</tbody>
</table>

2. **Tasks List**

1- Create a new account on the system.
2- On the left menu, Click on Find Events link.
3- Select any event and click Add to your favorite list icon.
4- Click the field name within any event to filter the list.
5- On the left menu, click Find People.
6- Select any person and click Follow Person icon.
7- On the left menu, click Events I Like.
8- Rate any event and write an appropriate review.
9- On left menu, click browse organization.
10- Select any two organizations, and click Add to my favorite organizations icon on each.
11-On left menu, click Organizations I am Interesting in
12-Select any organization, and click Remove organization from my favorite list icon.

3. Post-Tasks Questionnaire

<table>
<thead>
<tr>
<th>Question 1</th>
<th>DID YOU FINISH ALL TASKS SUCCESSFULLY?</th>
</tr>
</thead>
<tbody>
<tr>
<td>Answers</td>
<td></td>
</tr>
<tr>
<td>Yes</td>
<td>90.91%</td>
</tr>
<tr>
<td>No</td>
<td>9.09%</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Question 2</th>
<th>IF YOU ANSWERED THE PREVIOUS QUESTION WITH (No), TELL US WHICH TASK(S) YOU FOUND DIFFICULT?</th>
</tr>
</thead>
<tbody>
<tr>
<td>Answers</td>
<td>Add event to my favorite list</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Question 3</th>
<th>HOW YOU EVALUATE THE FAMILIARITY OF USER INTERFACE?</th>
</tr>
</thead>
<tbody>
<tr>
<td>Answers</td>
<td></td>
</tr>
<tr>
<td>1 (Lowest)</td>
<td>0%</td>
</tr>
<tr>
<td>2</td>
<td>0%</td>
</tr>
<tr>
<td>3</td>
<td>9.09%</td>
</tr>
<tr>
<td>4</td>
<td>63.64%</td>
</tr>
<tr>
<td>5 (Highest)</td>
<td>27.27%</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Question 4</th>
<th>TO WHICH LEVEL YOU FIND THE APPLICATION EASY TO USE?</th>
</tr>
</thead>
<tbody>
<tr>
<td>Answers</td>
<td></td>
</tr>
<tr>
<td>1 (Lowest)</td>
<td>0%</td>
</tr>
<tr>
<td>2</td>
<td>0%</td>
</tr>
<tr>
<td>3</td>
<td>9.09%</td>
</tr>
<tr>
<td>4</td>
<td>63.64%</td>
</tr>
<tr>
<td>5 (Highest)</td>
<td>27.27%</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Question 5</th>
<th>HOW YOU EVALUATE “FIND EVENTS” SERVICE?</th>
</tr>
</thead>
<tbody>
<tr>
<td>Answers</td>
<td></td>
</tr>
<tr>
<td>1 (Lowest)</td>
<td>0%</td>
</tr>
<tr>
<td>2</td>
<td>0%</td>
</tr>
<tr>
<td>3</td>
<td>9.09%</td>
</tr>
<tr>
<td>4</td>
<td>63.64%</td>
</tr>
</tbody>
</table>
4. Results

<table>
<thead>
<tr>
<th>Question 6</th>
<th>TO WHICH LEVEL YOU FIND THE APPLICATION GIVE YOU CONTROL TO MANAGE THE SERVICES?</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Answers</strong></td>
<td><strong>Responses</strong></td>
</tr>
<tr>
<td>1 (Lowest)</td>
<td>0%</td>
</tr>
<tr>
<td>2</td>
<td>0%</td>
</tr>
<tr>
<td>3</td>
<td>36.36%</td>
</tr>
<tr>
<td>4</td>
<td>36.36%</td>
</tr>
<tr>
<td>5 (Highest)</td>
<td>27.27%</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Question 7</th>
<th>DO YOU THINK THE APPLICATION IS INNOVATIVE &amp; ATTRACTIVE?</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Answers</strong></td>
<td><strong>Responses</strong></td>
</tr>
<tr>
<td>Yes</td>
<td>100%</td>
</tr>
<tr>
<td>No</td>
<td>0%</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Question 8</th>
<th>DO YOU RECOMMEND THIS APPLICATION TO YOUR FRIENDS?</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Answers</strong></td>
<td><strong>Responses</strong></td>
</tr>
<tr>
<td>Yes</td>
<td>100%</td>
</tr>
<tr>
<td>No</td>
<td>0%</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Question 9</th>
<th>DO YOU HAVE ANY COMMENTS OR SUGGESTIONS?</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Answers</strong></td>
<td>Add small descriptive text next to the buttons will make it more clear</td>
</tr>
</tbody>
</table>

IN WHICH AGE RANGE ARE YOU?

- 19-24: 18.18%
- 25-34: 45.45%
- 35-44: 36.36%
- 45-64: 0%
- 65+: 0%

- 19-24: 0%
- 25-34: 0%
- 35-44: 0%
- 45-64: 0%
- 65+: 0%
What is your gender identity?

- Male: 72.73%
- Female: 27.27%

What is your highest degree?

- High School: 54.55%
- Diploma: 18.18%
- Bachelor: 18.18%
- Master: 0%
- Doctorate: 9.09%

Did you ever participate in any kind of events or exhibitions?

- Yes: 90.91%
- No: 9.09%

Do you think creating new application dedicated for managing a particular exhibition is worthy?

- Yes: 72.73%
- No: 9.09%
- Not Sure: 18.18%

Do you use any social/professional network applications?

- Yes: 90.91%
- No: 9.09%

Do you plan to visit Expo 2020 - Dubai?

- Yes: 72.73%
- No: 27.27%
- Not Sure: 0%
Appendix D

DID YOU FINISH ALL TASKS SUCCESSFULLY?

- Yes: 90.91%
- No: 9.09%

HOW YOU EVALUATE THE FAMILIARITY OF USER INTERFACE?

- 1 (Lowest): 0%
- 2: 0%
- 3: 0%
- 4: 9.09%
- 5 (Highest): 63.64%

TO WHICH LEVEL YOU FIND THE APPLICATION EASY TO USE?

- 1 (Lowest): 0%
- 2: 0%
- 3: 9.09%
- 4: 63.64%
- 5 (Highest): 27.27%

HOW YOU EVALUATE “FIND EVENTS” SERVICE?

- 1 (Lowest): 0%
- 2: 0%
- 3: 0%
- 4: 9.09%
- 5 (Highest): 63.64%

TO WHICH LEVEL YOU FIND THE APPLICATION GIVE YOU CONTROL TO MANAGE THE SERVICES?

- 1 (Lowest): 0%
- 2: 27.27%
- 3: 36.36%
- 4: 36.36%
- 5 (Highest): 0%
DO YOU THINK THE APPLICATION IS INNOVATIVE & ATTRACTIVE?

- Yes: 100%
- No: 0%

DO YOU RECOMMEND THIS APPLICATION TO YOUR FRIENDS?

- Yes: 100%
- No: 0%