University of Palestine
Faculty of Information Technology


# Mobile Application For Personal Activities Management ( Life Scheduler )

**Prepared By :**
Ahmad Abu Sultan
Ahmed Al-Atbash
Ahmed Al-Alami

**Supervised By :**
Eng. Eman Al Ajrami

A Thesis Submitted in Partial Fulfillment of the Requirements for the Degree of Bachelor in Information Technology

## Acknowledgment

We would like to express our heartfelt gratitude to everyone who supported us through our work on this project, both directly and indirectly. First of all, thanks to Allah Almighty for giving us the courage and power to accomplish this project.

We would like to thank our supervisor, Eng. Eman Alajramy, for her optimistic and open-minded research spirit and methods and for her constant guidance, inspiration, generous help, support and patience throughout the development of this project. We have encountered many difficulties during the work on the project, Eng. Eman Alajramy has had the wisdom to point us in the right direction, yet allow us to find our own answers.

We would like also to thank our friends at the University of Palestine for making the best of the worst times during our study.

Finally, and most importantly, none of this would have been possible without the love and patience of our families, our fathers, mothers, brothers and sisters. Without their unconditional love, faith and unimaginable patience, this project would never have been possible.

## Abstract

Nowadays mobile applications are the most important thing in people life because it helped assigning several aspects of our life to it like entertainment, sociality, health, and etc. life Scheduler is a mobile app developed to help people manage and organize their daily tasks and activities .

This research aims to provide a mobile application that helps people managing and organizing their tasks. Because of the fast growing and development of the life around us the tasks that we want to do are increasing, so we might forget some of them or we don’t have enough time to manage them which can affect on our life from different sides, so this app came to solve this problem .

Extreme programming software development methodology was followed to develop this app, that the functional and non-functional requirements were defined form looking on other similar apps, that we have collected the most important features to the user and adding the new features to contribute with new useful things. Then the functional and non-functional requirements are formed and analyzed using different UML techniques such as use-case diagrams, sequence diagrams and activity diagrams .

The design of the system is then presented and analyzed using class diagram, database schema and ER diagram, that was used in implementing the app. Then the app was tested to see if it meets the user specifications, and the results proved that the app works well .

Then a user evaluation was conducted using questionnaire to measure the degree of user acceptance about the app, and the results proved that the app is useful and can be launched in the market but need some more improvements .

## Table Of Contents

## List Of Tables

## List Of Figures

## List Of Abbreviations

| | |
|---|---|
| IT | Information Technology |
| AI | Artificial Intelligence |
| OS | Operating System |
| app | Application |
| GUI | Graphical User Interface |
| PDA | Personal Digital Assistants |
| MIS | Management Information Systems |
| FMS | Flight Management System |
| MTS | Mobile Telephone Service |
| GPS | Global Positioning System |
| RDBMS | Relational Database Management System |
| XP | Extreme Programming |
| UML | User Modeling Language |
| ART | Android RunTime |
| DVM | Dalvik Virtual Machine |
| API | Application programming Interface |
| IDE | Integrated Development Environment |
| ERD | Entity Relationship Diagram |
| BRS | Business Requirement Specification |
| SRS | System Requirement Specifications |
| SIT | System integration testing |
| PU | Perceived Usefulness |
| PEOU | Perceived Ease of Use |

## Glossary

**Activity diagram:** it is a flow chart, which represents the flow form one activity to another one. The activity can be described as an operation of the system. Therefore, the control flow is drawn from one operation to another. This flow can be sequential, branched or concurrent. Activity diagrams deals with all type of flow control by using different elements like fork, join etc.

**Class diagram:** it represents the static view of an application. It is not only used for visualizing, describing and documenting different aspects of a system but also for constructing executable code of the software application. It also, describes the attributes and operations of a class and the constraints imposed on the system.

**Database Schema:** it is the skeleton structure, which represents the logical view of the entire database. It defines how the data is organized and how the relations among them are associated. It formulates all the constraints that are to be applied on the data.

**Entity Relationship Diagram:** It defines the conceptual view of a database. It works around real-world entities and the associations among them. At view level, the ER diagram is considered a good option for designing databases.

**Flowchart:** A flowchart is a visual representation of the sequence of steps and decisions needed to perform a process. Each step in the sequence is noted within a diagram shape. Steps are linked by connecting lines and directional arrows. This allows anyone to view the flowchart and logically follow the process from beginning to end.

**Functional Requirement:** The functional requirements are user "visible" features that are typically initiated by stakeholders of the system, such as generate report, login, and signup.

**Non-Functional Requirements:** they are requirements that describe how the system will do what it is supposed to do, for example, security, reliability and maintainability.

**Sequence diagram:** It models the collaboration of objects based on a time sequence. It shows how the objects interact with others in a particular scenario of a use case.

**Use-case diagram:** it is a graphic depiction of the interactions among the elements of a system. Also, it is considered a methodology used in system analysis to identify, clarify, and organize system requirements. Use case diagrams are employed in Unified Modeling Language (UML), a standard notation for the modeling of real-world objects and systems.

# CHAPTER 1 :

# INTRODUCTION

## 1.1 Preface

In this chapter we will demonstrate a basic information about the research and, why we have chosen this area, and the objectives that we put for the problem we attempting to solve, and how our project will contribute a new thing to the science .

## 1.2 Background

Nowadays our life has became full of duties that we have to do , whether it's on career side or on personal side , and with a lot of duties people might forget what they should or have to do which can cause a shortage in their performance in the job or with their families , so they might lose many things in their life .

The usage of Information Technology (IT) in different life disciplines has caused a faster growing and development to the world than any time ago , which in turn increased the duties on people .

The reason for using IT is to increase the productivity and efficiency of our life , so one of the disciplines that IT can be used in is our life activities, which must be managed to help us contributing new things in the world, so we need to know about (IT) to find out how to make it help us doing this .

IT may be defined as the technology that is used to acquire, store, organize, process, and disseminate processed data which can be used in specified applications [1] .

Information is processed data that improves our knowledge, enabling us to make decisions and initiate actions [2], So now we can know that IT is important to facilitate managing our life activities.

One of the IT areas is the Artificial Intelligence (AI) , that is used to let the machine helps making better decisions and controlling things at the same time that the human wants to do something else like the activities of our day , that we need to manage them to find free time to focus on other things like our businesses to increase our productivity.

AI as John McCarthy defined it " the goal of AI is to develop machines that behave as though they were intelligent " [3], the meaning of the definition is to make computers or software computers behave (think) in the same way that the human brain does, to assign the difficult jobs to computers rather than assigning it to humans which will take long time.

There are several things to use AI in it such as smart phones. since the electricity was discovered, new things were innovated to serve people like the telephone, which was innovated in 1876 by Alexander Graham Bell to exchange human voice to make communications.

Generation after generation, it was developed to become digitally till we reached the mobile phone in 1970s that allows users to establish their calls wirelessly (cell phones), and then they added to it an operating system to add new features (personal digital assistants (or PDAs)) like sending messages, choosing ringtones when receiving a call, using alarm, etc [4] .

And it was developed to have another options like camera, flash light and it's connected with the computer so we can insert or take things from its files and it's connected with the internet to add wide variety of tasks to do [4].

Till we reached that we can add applications, so the mobile phone starts acting like small computers, so from this point the concept of smart phones arise [4].

If we want to define the smart phones we can say that it is a mobile phone with an advanced mobile operating system which combines features of a personal computer operating system with other features useful for mobile or handheld use, so smart phone became one of the most important personal things that the person must own, from this point we need to use these features of smart phones to help us managing our life activities [4].

The applications used in the smart phone are called mobile applications. a mobile application is a software application designed to run on mobile devices such as smart phones and tablet computers [5], usually the apps used in these devices are separated into two types pre-installed apps like calendar, SMS messages, web browser and so on, the other type is that the user can get it from stores such as google play or app store, etc.

To develop these apps of course you need a programming language and an operating system (OS) on the device that must be compatible with these apps to operate them, there are many operating systems used in smart phones such as Android and IOS.

Android operating system was developed by Android, Inc. which was acquired by google in 2005 [6], now android OS is becoming more popular in most smart phones because of its advantages like it is an open source , some of the development tools are free, and it provides appropriate hardware platform for developers so they can spend less efforts to realize their ideas [7].

Android apps are developed using java programming language because it's an object oriented and can access extensive class libraries that help developers to develop powerful apps, and the Graphical User Interface (GUI) programming in java is event driven like screen touch, android uses the functionality of mobile phones with multitouch screen that allows controlling the device with gestures involving one touch or multiple touch according to the need [5], so using android to develop the apps is a good choice.

Now after a brief investigating about the technology and its features we are looking for developing a mobile application that helps people managing their life activities, the application shall have four main functions.

The first function is the prioritization that will arrange the activities according to values inserted by the user like the date of the activity, the day of the activity, the duration time, starting time, end time and the strength of the activity (low, medium, high) .

The second function is the reminder that will alert the user about the begging time of the activity so the user starts doing this activity.

The third function is the suggestion that will give the user a list of pre-inserted activities or he can reuse previous lists or choose a place from the map to visit if the user has leisure time and he doesn't know what to do he can use this function .

The forth function is join a friend, in this function there will be a map that shows the other users who use this application so they can share their activities to let their friends join them in their activities .

All these functions will work after the user insert a set of his activities and some required values.

## 1.3 Motivation for the research

We as a team chose this topic because we are interested with the advantage of the mobile applications , which are increasing very fast because it is not limited to a specific users, but anyone can have a mobile phone and download the apps he wants, and users can use them anywhere .

So we want to take this advantage to help people manage their tasks or duties in times they don't have time to do this , which if they did it by their own effort, would cause a shortage in their performance .

## 1.4 Problem statement

The difficulties that people face when they have many tasks or duties to do whether it is on business side or personal life side , that they might forget some of them because they have large amount of duties or tasks or because they are attempting to do them later after a period of time, therefore they will lose part of their life (in business or with their families) , and they might don't know how to arrange them or put a plan to execute them [8, 9].

Most of apps in the stores like S-planner, Microsoft ToDo [10], ToDo list task reminder [11], ToDo list [12], and more have solved part of the problem but they all alert more than one task at the same time which can create conflict to the person about which task is more important than the other to do first, so if he chose one of them he will forget the others .

Also most of these apps do not organize the tasks for the user, even the apps which has an algorithm to do so have the problem of alerting on more than one task at the same time, so the user will handle effort and mind thinking to decide which task is more important than the other to do first, which will make him waste time doing this process .

## 1.5 Research questions

In relation to the problem statement, this research aims to investigate the following :

- How to use the advantage of mobile apps to facilitate managing or arranging the tasks and duties (activities) of the user automatically without user's effort ?
- How to remove the conflict of alerting on several tasks at the same time .

## 1.6 Research objectives

- To develop an android mobile app that helps people managing their life activities, so they can concentrate on their businesses or other sides in their life.

- To develop an android mobile app that has an algorithm which organize the tasks of the user automatically .

- To develop an android mobile app that alerts on one task only to let the user focus in it .

## 1.7 Significance of the research

This research will merge between two sciences, the science of management and the science of AI, to facilitate human life through organize their tasks to let them find opportunities doing other things, this is important for people who spend their times in innovation.

Our app is not new, there are many task management apps in stores which have all or some functions of ours like reminder on tasks, creating list, providing suggested tasks, and joining a group event. But we added something new, which is the arrangement algorithm, that arranges the tasks to do them sequentially one after one .

## 1.8 Scope and limitations of the research

This app is targeting all interested people in organizing their life around the world, and it will be an Android app, because it is the most popular OS in the world.

It will be supported in two languages (English and Arabic), and the user can try it as a full feature for one month, then after that interval he has to buy it.

The app will not have an admin user, and there will not be an event that collects a group of users in it, and there will not be a synchronization in accounts .

## 1.9 Conclusion

In this chapter we have discussed a background about each discipline that our project uses, and the reasons why we choose to work in this area, and the problems that we are attempting to solve, and the objectives that the project will achieve, and the new things that the project will contribute to the science, and finally what the project includes and excludes .

# CHAPTER 2 : LITERATURE REVIEW

## 2.1 Introduction

This chapter demonstrates the technologies we have used to build our app and illustrates them clearly, and other similar apps to ours identifying the differences and similarities in the functionality .

## 2.2 technologies used

### 2.2.1 Management Systems

In every day of our life we think about the things that we want to do and we may make list to organize them , but our tasks are not the only thing that must be managed , every life aspect can be managed like business operations, computer operations, factories operations, even aircrafts .

There are different definitions for management, but the general one " Management is the process of achieving organizational goals by engaging in the four major functions of planning, organizing, leading and controlling by co-ordination of human, material, technological and financial resources " [13] .

So management is an important thing in our life, and here are some reasons [13] :

- Management helps in economic growth by enabling a country to experience a substantial level of economic development .
- Management decides the direction and correct way of doing the work .
- According to the need and environment of society, management decides suitable course of action .

When we merge between IT and management we get a management information systems (MIS) which they are " systems to convert data from internal and external sources into information, and to communicate that information in an appropriate form, to help managers at all levels in all functions making timely and effective decisions for planning, directing and controlling the activities which they are responsible " [14] .

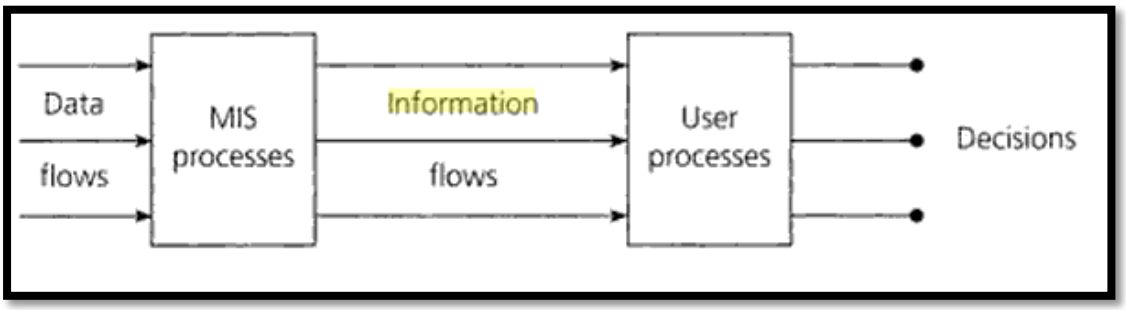



**Figure 2.1 Decision focus of MIS [14]**

Indeed these systems are computer based, that serve many disciplines, for example, the Flight Management System (FMS) which is used in aircrafts to help pilots identify basic information like current airport, destination airport, takeoff runway,

takeoff speed, weights and etc, after inserting these information the system tells the pilot about the steps that he must follow and alert him if something went wrong .

So according to the previous example we can notice that MIS used to make better decisions figure (2.1) illustrates how these systems work .

As in figure (2.1) and the previous example, we can notice that our app follows the same processes but there is no decision to make it just helps remembering the tasks and organizing them without user effort .

### 2.2.2 Mobile Technology

Communication is the most important element in the human life, because every life aspect depends on it such as trade, traveling, wars, or communicating with our relatives. The term mobile can be used to define any device that communicates remotely or wirelessly, these devices used to perform tasks remotely like communication, navigation, computation and other things, also this term usually connected with telecommunication usage specially with the telephone because the first mobile device began with phoning task which was called mobile phone.

Any wireless or cellular system depends on concept of electromagnetic waves transmission that send radio signals over the air with specific frequency to hold the data [15] .

As we mentioned before The first form of the mobile technology is the mobile phone that began in 1940s for military usage [16] .

To track the phases of mobile development we need to look at the wireless system connected to . After the war world 2 many cities lay in distraction so the infrastructure of the landline networks were destroyed, many researches and developments were done on mobile radio to use it instead of the landline networks [16] .

In 1945 they established a small zone of radio system in New York, a year after that in 1946 the first American commercial mobile-telephone service began, that the AT&T company and south-western Bell company began operating Mobile Telephone Service MTS, which used a centrally located antenna that transmit the signal to a moving mobile devices across the area which these devices are car based radio telephones to continue transmitting the signal to the receivers [16] .

In 1948 they created the first fully automatic radiotelephone service system, till the transistor was discovered they provide an automatic dialing in 1964 so people did not have to press a button to talk but conversation went back and forth just like the wire telephone, and in 1960s people could conduct their calls while they are moving using computerized control system to manage the process [16] .

Around 1969 the first all transistor mobile telephones appeared so the tube era for radio telephones ended, in 1971 Intel corporation introduced the first commercial microprocessor which was used in several electronic devices including telephone switches and cell phones allowing the mobile phones to be digitally, in 1990s the concept of personal cellular system appeared and the commercial GSM network was used which provided to every mobile phone to contain encryption to prevent eavesdropping, authentication to prevent fraud, short messaging service or SMS, and

an SIM card to easily add accounts to a handset, this network became the most popular cellular radio service around the world in 2004 [16] .

### 2.2.3 Mobile Application

Application is a computer program built to perform many types of coordinated functions , tasks , or activities that provide benefits to the user .

Mobile applications also called apps are add-on software for mobile (hand held) devices, such as smartphones and personal digital assistants (PDA) , these applications differs in the type of functionality they provide like entertainment apps (games) , social apps (facebook, whatsapp) , foreign language phrasebooks apps (dictionaries) , maps to show nearby locations or people (google map) , list of local events (alarm , calendar) , multimedia players apps (for audio , video ) , decision support apps , and all other types of intelligent apps to serve people [17] [18] [19] .

Each type of smartphones has an operating system to run it , and each operating system has a specific programming language built it which this language will develop its apps , table (2-1) illustrates the types of mobile platforms and the programming languages for each one and the store where we can download them , so there are some apps are free to download and others are paid [17] .

**Table 2-1 Characteristics of selected mobile platforms [17]**

| vendor | Operating system | Programming language | Application store |
|---|---|---|---|
| Apple | iPhone OS | Objective-C | App Store (July 2008) |
| LiMo Foundation | LiMo Platform (Linux) | Java, native (C/C++) | R2 (autumn 2009) |
| Microsoft | Windows Mobile | Visual C#/C++ | Windows Marketplace for Mobile (autumn 2009) |
| Open handset alliance | Android (Linux) | Java | Android Market (October 2008) |
| Palm | Palm OS | C/C++ | Palm App catalog (June 2009) |
| | webOS (Linux) | JavaScript, HTML 5 | |
| Qualcomm | BREW | C/C++ | Plaza Retail (May 2008) |
| RIM | BlackBerry OS | Java | BlackBerry App World (April 2009) |

| | | | |
|---|---|---|---|
| Symbian Foundation | Symbian | C++ | Nokia Ovi Store (May 2009) |

Following the table above mobile applications are created by third-party environment called IDE (Integrated Development environments) that provide source code editor, build automation tools and a debugger for the programming language used , Mobile apps are first tested within the development environment using emulators and later subjected to field testing. Emulators provide an inexpensive way to test applications on mobile phones to which developers may not have physical access [17] [20] .

### 2.2.4 Android

Android is the world's most popular mobile operating system, which was developed by Android (Inc) and acquired by Google in 2005 . It is based on Linux kernel [7] .

Android devices include smartphones, tablets, e-readers, robots, jet engines, NASA satellites, game consoles, refrigerators, televisions, cameras, health-care devises, smartwatches, (the OS of cars) automobile in vehicle "information" systems (for controlling radio, GPS, phonecalls, thermostat, etc ) and more, of course all these devices used Android as an OS because its mobility feature [7] .

Android had 82.8% of the global smartphone market share, compared to 13.9% for Apple and 2.6% for Microsoft . According to *PC World* , approximately 230 million tablets shipped in 2014 of which 67.3% were Android tablets, compare to 27.6% for iOS and 5.1% for Microsoft Windows [7] .

Here are some Android features [7] :

- **Openness and Open Source :** One benefit of developing Android apps is the openness of the platform. The operating system is open source and free. This allows viewing Android's source code and see how its features are implemented .
- **Multitouch Screen :** Android smartphones wrap the functionality of a mobile phone, Internet client, MP3 player, gaming console, digital camera and more into a handheld device with full-color multitouch screens. With the touch of fingers, the user can navigate easily between using the phone, running apps, playing music, web browsing and more. The screen can display a keyboard for typing e-mails and text messages and entering data in apps .
- **Gestures :** The multitouch screens allow the user to control the device with gestures involving one touch or multiple simultaneous touches (Table 2-2) .

**Table 2-2 Some common Android gestures [7] .**

| Gesture Name | Physical action | Used to |
|---|---|---|
| Touch | Tap the screen once. | Open an app, "press" a button or a menu item. |
| Double touch | Tap the screen twice. | Zoom in on pictures, Google Maps and web pages. Tap the screen twice again to zoom back out. |
| Long press | Touch the screen and hold your finger in position. | Select items in a view—for example, checking an item in a list. |
| Swipe | Touch the screen, then move your finger in the swipe direction and release. | Flip item-by-item through a series, such as photos. A swipe automatically stops at the next item. |
| Drag | Touch and drag your finger across the screen. | Move objects or icons, or scroll through a web page or list. |
| Pinch zoom | Pinch two fingers together, or spread them apart. | Zoom in and out on the screen (e.g., resizing text and pictures). |

- **Built-in Apps :** Android devices come with several default apps, which may vary depending on the device, the manufacturer or the mobile service carrier. And here are some common apps pre-installed with any device (Phone, Contacts, Messenger, Browser, Calculator, Calendar, Clock and Photos) .
- **Web Services :** are software components used in apps to combine the functionalities of other apps (or other software component) on another computer over the Internet, For example, 100 Destinations ( http://www.100destinations.co.uk ) combines the photos and tweets from Twitter with the mapping capabilities of Google Maps to allow you to explore countries around the world through the photos of others, (Table 2-3) lists some popular web services .

**Table 2-3 Some popular web services [7] .**

| Web services source | How it's used |
|---|---|
| Google Maps | Mapping services |
| Twitter | Microblogging |
| YouTube | Video search |
| Facebook | Social networking |
| Instagram | Photo sharing |
| Foursquare | Mobile check-in |
| LinkedIn | Social networking for business |
| eBay | Internet auctions |
| PayPal | Payments |
| Skype | Internet telephony |
| Microsoft Bing | Search |
| WeatherBug | Weather |

As other OSs Android has versions, and each new version of Android is named after a dessert, and each version has new features according to the capabilities of the hardware components of the device that uses Android, (Table 2-4) shows the versions of the Android [7] .

**Table 2-4 Android version numbers and the corresponding names [7] .**

| Android version | name |
|---|---|
| Android 1.5 | Cupcake |
| Android 1.6 | Donut |
| Android 2.0-2.1 | Éclair |
| Android 2.2 | Froyo |
| Android 2.3 | Gingerbread |
| Android 3.0-3.2 | Honeycomb |
| Android 4.0 | Ice Cream Sandwich |
| Android 4.1-4.3 | Jelly Bean |
| Android 4.4 | Kit Kat |

| | |
|---|---|
| Android 5.0-5.1 | Lollipop |
| Android 6.0 | Marshmallow |
| Android 7.0 | Nougat |

Users can get the apps from online stores, the most popular stores are Google Play and Play Store app, which are installed with the Android [7] .

#### 2.2.4.1 Java

Android apps are developed with Java, one of the world's most widely used programming languages. Java was a logical choice for the Android platform, because it's powerful, free, open source and used by millions of developers [7] .

Experienced Java programmers can quickly dive into Android development, using Google's Android APIs (Application Programming Interfaces) and others available from third parties [7] .

Java is object oriented and has access to extensive class libraries that help quickly develop powerful apps. GUI programming in Java is event driven, which is good with Android [7] .

#### 2.2.4.2 Software Development Kit (SDK)

The Android SDK is a programming package that provides the tools and APIs (Application Programming Interface) needed to build Android apps [7] .

Also SDK provides the Android Emulator, which allows the developer to run Android apps in a simulated environment within Windows, Mac OS X or Linux, without using an actual Android device [7] .

The emulator displays a realistic Android user-interface window. It's particularly useful when there is no need to access an Android devices for testing. The developer should test his apps on a variety of Android devices before uploading them to a store [7] .

#### 2.2.4.3 Mobile Maps

In the past people used to put marks on the trees or on the stones so they can remember the way they have taken when they are travelling, and generation after generation they became drawing their ways on surfaces like paper to help them identify, understand and navigate their way around .

Map is a visual representation of an entire area or a part of an area, and used to illustrate specific and detailed features of a particular area such as political boundaries, physical features, roads, topography, population, climates, natural resources and economic activities [21] .

Since human travelled to the space and established satellites it became easy to create digital maps which in turn more accurate than the hand drawn maps, this allowed a new technology to emerge, it is the GPS (Global Positioning System) .

GPS functionality can be applied on mobile devices and used to calculate the user's position, and it can be combined with map data to provide ways for users to find locations and get directions to them, so maps without GPS are nothing except a static images [22] .

Maps apps used the features of the mobile devices (discussed in Table 2-2) which made using it easier than desktop or website map application, because simply the user effort in smartphone is less than any device specially when using touch gesture instead of mouse dragging [23] .

One of the most common maps apps is Google map, which was developed by Google, it has functionalities like [24] :

- Locating the user's (device) position .
- real-time navigation .
- real-time transit information for catching bus, train, or ride-share .
- re-routing based on live traffic .
- Find pit stops along your route like gas stations and coffee spots .
- Discover and explore places .
- Find top-rated restaurants and local businesses .
- Decide on the best places to go with reviews, ratings, and pictures of foods and interiors .
- Plan for visit and see menus, make reservations, and find when places are typically busiest .
- Help others discover the best places by sharing reviews or photos .

#### 2.2.4.4 NoSQL

NoSQL stands for (not only SQl), it is not relational database, which does not use tables to store data, but it stores entities as tree that each entity has its own data, that means it is not structured [25] .

NoSQL used for applications that have large volume of data like social media, because it provides quick access to data [25] .

Generally it does not use the principles of the relational database management system (RDBMS) for data manipulation [25] .

Table 2-5 shows list of NoSQL databases that can be used for applications (websites/mobile apps) [25] .

**Table 2-5 list of NoSQL databases [25].**

| Document | Key-value | XML | Column | Graph |
|---|---|---|---|---|
| MongoDB | Redis | BaseX | BigTable | Neo4J |
| CouchDB | Menbase | eXist | Hadoop/Hbase | FlockDB |
| RavenDB | Voldemort | | Cassandra | InfiniteGraph |
| Terrastore | MemcasheDB | | SimpleDB | |

## 2.3 Related systems

To demonstrate the robustness and the new feature of our app we have done a comparison with other related apps that have related functions .

### 2.3.1 App1 : Chaos Control-GTD Task List

This app has been developed specially to serve businessmen , innovators , designers , writers , developers , and any person has several tasks and goals which he does everyday . By this app the user can manage his tasks and goals , defining his priorities , and avoiding chaos that could be consequenced by several daily tasks , the app provides some functions that helps to manage the tasks of the user's work , projects planning , marketing management , determining goals , organizing tasks , and set alerts to appointments , etc .

The app is available on free version and paid version for all devices and smartphones for all platforms .

The main functionalities of this app :

- adding goals : the user can add his goals to study them .
- make a work plan .
- adding daily activities : the user can add his daily activities and make lists of them but he can organize them manually as he wants .
- alert for the activities .

Following these functions we can describe the features of this app which the user can manage his time , planning for his work and his activities , manage his bills and shopping lists , and write his thoughts and notes . Also this app uses cloud computing to synchronize the stored data with all devices so the user can access his account from any device [26] .

### 2.3.2 App2 : ASUS Do It Later

This app lets the user focus on his most important tasks without missing a beat. When the user is too busy to deal with a call, reply to an email or follow an interesting link, he just taps to send the task to the ASUS Do It Later list .

ASUS Do It Later integrates tightly with many other popular third-party apps, such as Google Maps, Chrome and YouTube .

ASUS Do It Later also synchronizes seamlessly with Google and Microsoft Exchange accounts to keep all user's lists in check .

When the user is ready to deal with his tabled tasks, ASUS Do It Later helps him to pick up the task from where he is by just tapping the task card's action button to add the task to the app [27] .

The main functionalities of this app :

- Seamless integration with many third-party apps, that just one tap on the menu option sends tasks to ASUS Do It Later .
- Dynamic action button adjusts to reflect the task type, allowing the user to deal easily and instantly with emails, messages, phone calls and much more .
- Smooth snapshot-style animation indicates clearly when that the task has been sent to ASUS Do It Later .
- Easy task-management page lets the user sees, sorts and swipes his tasks .
- All-new widget shows unfinished tasks even more clearly and allows you to create new tasks .

### 2.3.3 App 3 : To DO List

To Do List is an App that allows the user to take down his tasks quickly and easily, and manage his time effectively [12] .

The main functionalities of this app :

- add multiple tasks to the user to-do list .
- Sliding left or right to view different to-do lists .
- Organize and effectively manage the to-do items based on the user's preferred criteria such as completed date, due date and priority .
- Set alarm reminders for the to-do items .

### 2.3.4 App 4 : TODO LIST Task Reminder

This app makes the user focuses on really important things At home, at work and in free time. And allow the user to add, schedule, and re-schedule tasks from phone, tablet, desktop, email, and more, even offline. It provides an automatic, 24/7 synchronization across all user's devices [11] .

The app sets up and receives notifications, email or SMS reminders based on user's location related to the created tasks or a specific due date and time. It easily allows uploading files, media and photos to user's tasks from user's computer, Dropbox, etc .

The main functionalities of this app [11] :

- Smart voice input while creating tasks and Set reminders .
- Add and view tasks on Google calendar and synchronization .
- Re-arrange tasks easily by dragging and dropping.
- Set customized repeating reminders: daily/weekly/monthly, and so on .
- Add comments and attachment like image along with tasks .

- Sort tasks manually, smart sort, by due date, by priority, by title, or by last modified .
- Miss call alert and call later functionality .
- Voice reminders, that tasks will be spoken with task name .

### 2.3.5 App 5 : Microsoft To-Do

This app is a simple and intelligent to-do list that makes it easy to plan for the day, whether it is for work, school or home, this app will help the user to increase his productivity and decrease his stress levels [10] .

It combines intelligent technology and beautiful design to empower the user to create a simple daily workflow. The user can organize his day with To-Do's smart Suggestions and complete the most important tasks or chores he needs to get done, every day [10] .

The main functionalities of this app :

- synchronization between phone and computer, so the user can access his tasks from school, the office, or the grocery store or even while he is traveling .
- add reminders and due dates for tasks And if the user have tasks that he needs to do on a daily, weekly or yearly basis, he can set up recurring due dates to remind him each and every time .
- adding detailed notes to every task .
- create list of tasks for each goal .
- reuse previous lists .

### 2.3.6 App 6 : S Planner

This app is one of the built in apps that are downloaded with each Android OS, which allows the user to know the date of each day in calendar form .

The has more features such as adding tasks or events on the selected day .

The main functionalities of this app :

- Show a calendar .
- Add tasks or events to a day or set of days .
- Set reminder to tasks using time .
- Set priority to the task or event .
- Attaches description or image to the task or event .
- Repeat reminder to a specific period .
- Enter a location to visit using map .

**Table 2-6 comparison between other similar apps**

| criteria | apps | | | | | | |
|---|---|---|---|---|---|---|---|
| | Chaos Control-GTD Task List | To Do List | TODO LIST Task Reminder | Microsoft To-Do | S Planner | ASUS Do It Later | Life scheduler |
| Add tasks | √ | √ | √ | √ | √ | √ | √ |
| Create lists | √ | √ | √ | √ | √ | √ | √ |
| Reminder alert | √ | √ | √ | √ | √ | × | √ |
| Auto prioritization for taskss | × | × | √ | × | × | × | √ |
| Provide suggested tasks | × | × | × | √ | × | × | √ |
| Joining on map | × | × | × | × | × | × | √ |
| Synchronization with other devices | √ | × | √ | √ | × | √ | × |
| Add tasks from other apps | × | × | × | × | × | √ | × |
| Voice input and reminder | × | × | √ | × | × | × | × |

As shown in table (2-6) there are many apps in the stores about task management with different features and functionalities, all apps share in functionalities like adding tasks, creating lists, and set reminder alert for each task, and most of these apps share the synchronization functionality that allows the user to access his tasks from different devices, and each app has a distinct feature that makes it different than others .

Our app has some functionalities that are existed in most of these apps, but we have added a new features such as the arrangement algorithm (discussed in Chapter 4 ) that arranges the tasks one by one, this means that the app will not allow two or more tasks to be at the same level so it will alert about only one task, according to this feature the user is obligated to do this task before anything else, but these apps alert two or more tasks at the same time which will create conflict to the user about which to do first, the highlighted row in the table (Auto prioritization) shows that there is no app has this functionality except one but it is also not the same with ours because this app allow alerting two tasks at the same time .

The other feature which we have added is to let the user join his friends tasks, according to the table we did not found any app that has this functionality .

## 2.4 Summary

We have discussed in this chapter what technologies we have used to develop our app , and other similar apps which have functions related to ours .

# CHAPTER 3 : RESEARCH METHODOLOGY

## 3.1 Introduction

This chapter demonstrates the methodology used to develop this project, and the reason for choosing this methodology, and describing how go through each phase of this methodology .

## 3.2 Methodology used

To develop our project we have used Extreme Programming (XP) methodology .

### 3.2.1 Extreme Programming

Extreme Programming (XP) is a system of practices that a community of software developers is evolving to address the problems of quickly delivering quality software and then evolving it to meet changing business needs [28] .

XP is considered as a lightweight, efficient, low-risk, flexible, predictable, scientific and fun way to develop software [29] .

It was developed by Kent Beck in 1996, and it works by bringing the whole team together in the presence of simple practices, with enough feedback to enable the team to see where they are and to tune the practices to their unique situation [29] .

Beck states that XP was developed to address the specific needs of software development conducted by small teams in the face of vague and changing requirements [29] .

XP is distinguished from other methodologies by [29] :

- Its early, concrete, and continuing feedback from short cycles .
- Its incremental planning approach, which quickly comes up with an overall plan that is expected to evolve through the life of the project .
- Its ability to flexibly schedule the implementation of functionality, responding to changing business needs .
- Its reliance on automated tests written by programmers and customers to monitor the progress of development, to allow the system to evolve, and to catch defects early .
- Its reliance on oral communication, tests, and source code to communicate system structure and intent .
- Its reliance on an evolutionary design process that lasts as long as the system lasts .
- Its reliance on the close collaboration of programmers with ordinary skills .
- Its reliance on practices that work with both the short-term instincts of programmers and the long-term interests of the project .

#### 3.2.1.1 Extreme Programming Values

XP is not really a set of rules, but rather a way to work in harmony with your personal and corporate values. It is a lightweight discipline of software development based on values of simplicity, communication, feedback, and courage. These are

values, which, if implemented correctly, will improve any software development project [30] .
These values can be summarized as follows [29] :

- **Simplicity :** XP is betting that it is better to do a simple thing today and pay a little more tomorrow to change it if it needs it, than to do a more complicated thing today that may never be used anyway. Further, it encourages starting with the simplest solution then, extra functionality can be added later. This maximizes the value created for the investment. Programmers take small simple steps to their goal and mitigate failures as they happen .
- **Communication :** XP aims to keep the right communications flowing by employing many practices that cannot be done without communicating. They are practices that make short-term sense, like unit testing, pair programming, and task estimation. The effect of testing, pairing, and estimating is that programmers and customers and managers have to communicate. Everyone is part of the team and they communicate face to face daily. The team work together on everything from requirements to code .
- **Feedback :** Feedback works at different time scales. At the scale of minutes and days, the programmers write unit tests for all the logic in the system that could possibly break. The programmers have minute-by-minute concrete feedback about the state of their system. When customers write new "stories", the programmers immediately estimate them, so the customers have concrete feedback about the quality of their stories. The person who tracks progress watches the completion of the tasks to give the whole team feedback about whether they are likely to finish everything they set out to do in a span of time. Feedback also works at the scale of weeks and months, where customers and testers write functional tests for all the stories implemented by the system. They have concrete feedback about the current state of their system. The customers review the schedule every two or three weeks to see if the team's overall velocity matches the plan, and to adjust the plan .
- **Courage :** the team tell the truth about progress and estimates. They do not document excuses for failure because they plan to succeed. Communication supports courage because it opens the possibility for more high-risk, high-reward experiments. Simplicity also supports courage because you can afford to be much more courageous with a simple system. Further, courage supports simplicity because as soon as you see the possibility of simplifying the system you try it. Concrete feedback supports courage because you feel much safer trying radical surgery on the code if you can push a button and see the tests turn green at the end .

#### 3.2.1.2 Extreme Programming Practices

The previous four values of XP are supported by twelve XP practices. These practices tend to keep the team on track while they build up the system. In XP, every contributor to the project is an integral part of the whole team formed around a business representative – the “Customer”.
These twelve practices can be summarized as follows [29] :

- **The Planning Game :** The XP planning process allows the XP "customer" to define the business value of the desired features and uses cost estimates

provided by the programmers, to choose what needs to be done and what needs to be deferred. The effect of XP's planning process is that it is easy to steer the project to success. There are two planning steps in XP the release and iteration planning. The Release planning is a practice where the customer presents the desired features to the programmers in the team who in return estimate their difficulty. While, iteration planning is a practice whereby the team is given direction every couple of weeks building a software in 2 weak "iterations" and delivering running useful software at the end of each iteration .

- **Small Releases :** XP teams put a simple system into production early, and update it frequently on a very short cycle. The team releases running, tested software, deliver ring business value chosen by the Customer, every iteration. The most important aspect is that the software is visible, and given to the customer, at the end of every iteration. This keeps everything open and tangible .
- **Customer tests :** XP teams focus on validation of the software at all times. Programmers develop software by writing tests first and then software that fulfils the requirements reflected in the tests. Customers provide acceptance tests that enable them to be certain that the features they need are provided. The best way for success is that once the test runs, the team keeps it running correctly thereafter .
- **Metaphor :** XP teams develop a common vision of how the program works, which is called metaphor. In other words, they use a common "system of names" and a common system description that guides development and communication .
- **Simple Design :** A program built with XP should be the simplest program that meets the current requirements. The focus is on providing business value. There are design steps in release planning and iteration planning, plus teams engage in quick design sessions and design revisions through refactoring, through the course of the entire project. In an incremental, iterative process like XP, good design is then essential. That is why there is so much focus on design throughout the course of the entire development .
- **Refactoring :** also called design improvement in which XP focuses on delivering business value in every iteration. To accomplish this over the course of the whole project, the software must be well designed. This is done by keeping the software clean: without duplication, with high communication, simple, yet complete. Refactoring is strongly supported by comprehensive testing to be sure that as the design evolves. Thus the customer tests are a critical enabling factor. The XP practices support each other: they are stronger together than separately .
- **Pair Programming :** XP programmers write all production code in pairs, two programmers working together at one machine. This practice ensures that all production code is reviewed by at least one other programmer and results in better design, better testing and better code. In fact, many experiments have shown that pair programming produces better software at similar or lower cost than programmers working alone do .

- **Collective Code Ownership :** All the code belongs to all the programmers. This lets the team go at full speed, because when something needs changing, it can be changed immediately, which increases code quality and reduces defects .
- **Continuous Integration :** XP teams integrate and build the software system multiple times per day. This keeps all the programmers on the same page, and enables very rapid progress. Perhaps surprisingly, integrating more frequently tends to eliminate integration problems that plague teams who integrate less often .
- **Sustainable Pace :** sometimes known as 40-hour week. XP programmers work hard and at a pace that can be sustained indefinitely. This means they do not work overtime, unless it is effective, keeping themselves fresh, healthy, as to reduce as much as possible mistakes .
- **On-site Customer :** An XP project is steered by a dedicated individual who is empowered to determine requirements, set priorities and answer questions as the programmers have them. The effect of being there is that communication improves, with less hard-copy documentation .
- **Coding Standard :** For a team to work effectively in pairs and to share ownership of all the code, all programmers need to write the code in the same way, with rules that make sure the code communicates clearly .

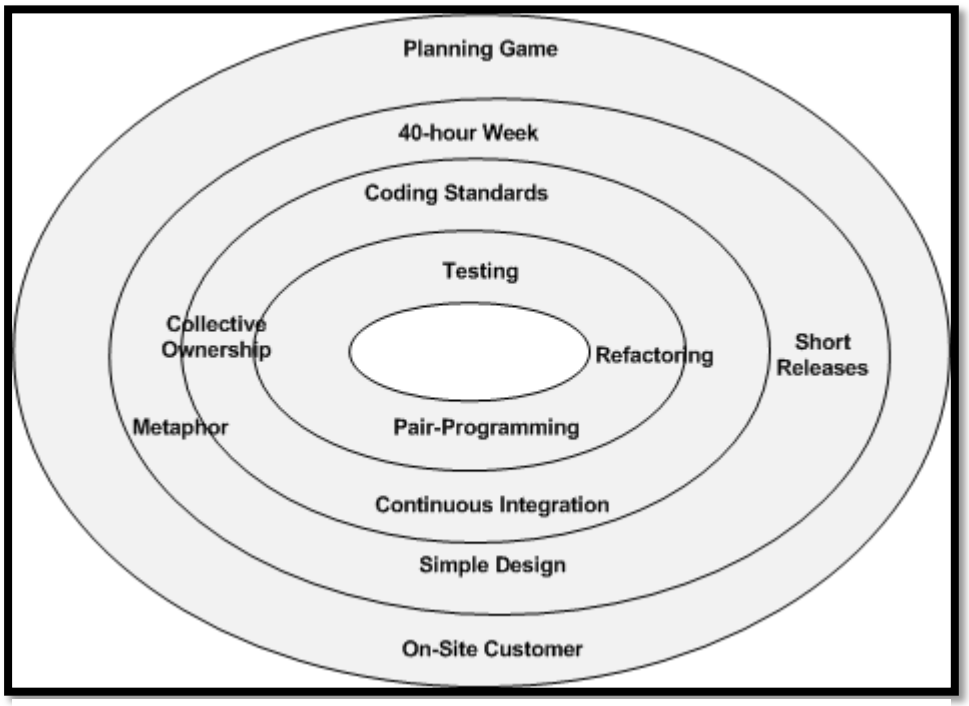



**Figure 3.1 XP practices [29]**

### 3.2.1.3 Advantages of Extreme Programming

XP has several advantages, some of these :

- Customer focus increase the chance that the software product will actually meet the needs of the users .
- The focus on small, incremental release decreases the risk of your project by showing that your approach works and by putting functionality in the hands of your users, enabling them to provide timely feedback regarding your work
- Continuous testing and integration helps to increase the quality of your work .
- XP is attractive to programmers who normally are unwilling to adopt a software process, enabling your organization to manage its software efforts better .

### 3.2.1.4 Disadvantages of Extreme Programming

On the other hand, the disadvantages of XP are :

- XP is code centric rather than design centric development. The lack of XP design concept may not be serious for small programs .
- XP does not measure or plan Quality aspect of development .
- XP emphasizes on refactoring during software development process .
- XP has not been proven to work with systems that have .
- Scalability issues (new applications must integrate into existing systems) .

### 3.2.1.5 Extreme Programming Phases

Extreme Programming can be implemented by a disciplined methodology that focuses prim airily on customer satisfaction. As all other software engineering methods, the process in XP is similar to the process used elsewhere. In fact, we can divide the above discussed practices into four clear progressions that are the planning, the design, the coding and the testing, as shown in Figure (3.2) .

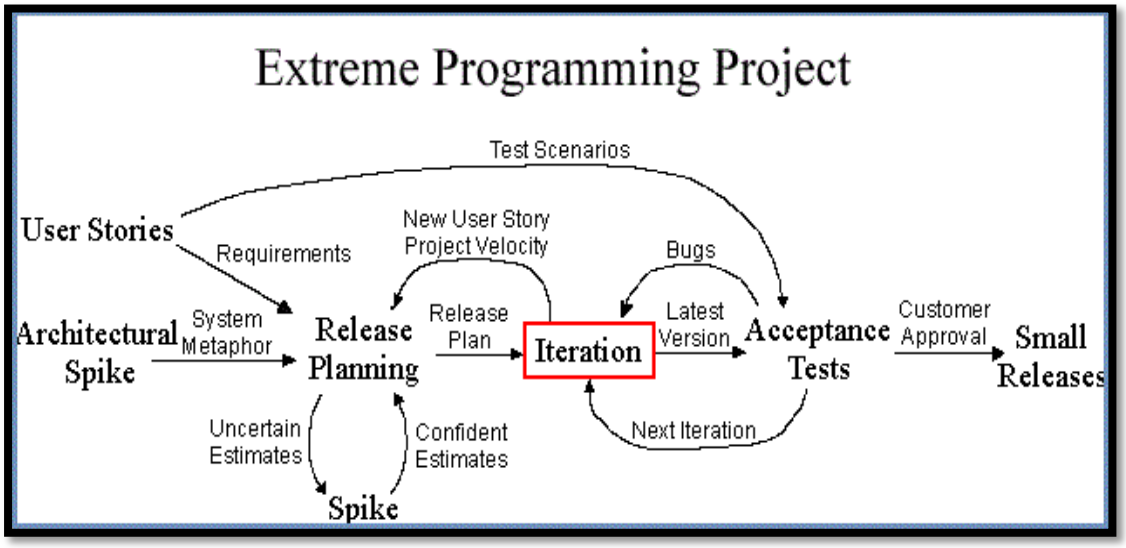



**Figure 3.2 XP life cycle**

In the **planning phase**, the development team interviewed the customer to determine user stories about how the system will work. Then, user stories were combined or split to come up with a story that can be described on a Class-Responsibility-Collaboration (CRC) card and completed by a pair of programmers during one release cycle. The customer prioritized the cards according to the business value of the user stories, which puts stories into the release schedule based on the development resources available. This lets customers have an accurate commitment schedule. After each release, the customer had a system that works according to the stories completed so far; they do not have to wait for the whole thing to be done to start using the initial functionality. The development group identified which stories are risky to complete on time and does worst things first based on a spike solution .

On the other hand, in the **Implementation phase** , Team work in pairs in two main phases the first phase was the **design phase** we start development app by create the 3d shape first step we chose the image target to make 3d shape same then We started modeling the 3d animal after modeling we go to texture the 3d shape by add colors and details then we animated the 3d object and save the animation .

and the second phase was **code phase** team working together on the code, to insure that the entire development team achieves Code Stewardship. Code Stewardship is the opposite of Code Ownership, emphasizing that code is the team's property, and not the sole province of any one person or pair. Programmers coded Unit Tests first. Programmers followed the coding conventions (standards) whenever they wrote a new code, and while they refactor mercilessly. Then written code was integrated, typically each day and was optimized as it is written so that there is no need for revision afterwards. Coding was done according to the corresponding user stories written in the order that the customer informs. Each story was written stand alone and at the end of the day integrated into the rest of the project .

**Testing phase** included Unit Tests for all code, creating tests when bug is found, and running Acceptance Tests. With extreme testing the software was developed more quickly, with more confidence, and with higher quality. Extreme Testing was conducted by two tests: unit tests and functional tests. Unit Tests let developers evolved the system rapidly and with confidence. Unit Tests for every feature were written before coding to ensure that new features work. Then, all the Unit Tests in the entire system were run before any code is released. When every Unit Test in the entire system is running 100% that showed us not just that the new feature works, but that the changes haven't broken anything anywhere. On the other hand, functional tests were used to check each increment to see if the value was there. These tests gave customers and developers confidence that the whole product is progressing in the right direction .

## 3.3 Why Extreme Programming?

XP is rapidly becoming recognized as an approach particularly well suited to small teams facing vague or rapidly changing requirements. That is, the majority of projects in today's fast-paced software development world. XP was used due to the fact that:

- Requirements are changing rapidly .
- High risk is involved as this is a new challenge projects .
- Small groups of programmers (between 2-10) .
- It is able to create automated tests .
- Direct customer involvement is possible .

According to the previous points XP is suitable for our project because we are small team that consists of three members, and because we have to deliver the project in a specific date so there are risks (one of these risks that one of the members has lift the group), and because we faced requirements changing every time specially in the arranging algorithm .

## 3.4 Conclusion

This chapter discussed the XP model of the project and methodology phases, And advantage and limitation of XP model, and we discussed why we have used this model in this project then we have talked about XP Values And we have explained the twelve Practices of XP .

# CHAPTER 4 : REQUIREMENTS ANALYSIS

## 4.1 Introduction

This chapter demonstrates a full description of the app and its users, and the functional and non-functional requirements, and the requirements analysis tools like use-case diagram, sequence diagram, and activity diagram .

## 4.2 System Description

Life scheduling is a mobile application that helps people to organize their activities and tasks to facilitate their life .

The app consists of four basic functions including :

- Prioritization .
- Reminder .
- Suggestion .
- Join a friend .

**The prioritization function :** after inserting the tasks and a set of values to each task including name of task, type of task (specified time or duration time), start and end time, duration, degree of priority (high, medium, or low), the app will create a list includes all inserted tasks, and the tasks in the list will be arranged according to arrangement algorithm discussed later in this chapter .

**The reminder function :** the app will send a notification or alert the user about the task that he have to do .

**The suggestion function :** allows the user to choose tasks if he has leisure time and wants to do a task but he doesn't know what is it, it has different ways like pre-inserted tasks, use previous lists, choose a place from the map to visit .

**The join a friend function :** provides a map that shows the friends users who use the same app to allow the user to joining a friend and doing the same task .

Now let's go through the app and explain how each function works .

The first step for the user in the app is adding tasks, for each task the user will insert the name of it and then if he knows the start and end time for it he will choose the specific time field and fill these values, if he does not know a specific time for the task he will set a duration time for it and determine the degree of priority of the task using non-specific time field, then adding this task to the today list .

There are some settings values the user must define because they are required to the arrangement algorithm, the user must determine which task he would prefer to do first the shortest or the longest if the app faced two or more tasks with same priority but each has different duration time (the default will be the shortest), and the user must determine the start and end time of the day according to his needs (the default will be from 7:00 AM to 10:00 PM) .

Now with every time the user adds a task an arrangement algorithm will work in these steps (see appendix C, D, and E for more explanation) :

- First all the times like start/end time of tasks and start/end time of the day must be converted to 24 hours system to make it easier dealing with them in calculations .

- In the database there are four tables the first one for the specific tasks, the second for the non-specific tasks, the third one for the today list, and the forth one for all lists .
- The table of the specific tasks must be arranged according to the start time in descending order, if there are two or more tasks with the same start time an exception message appears saying “this task cannot be added” and remove it .
- The period of each task must be blocked so no other tasks can be added in this period, this will happen with these conditions :
  - the start/end time of the new task must not come in the middle period of any task .
  - the period of the new task must not come outside the start/end time of the day .
- After that the specific tasks table is arranged and all the tasks in it are transferred to the today list .
- The table of the non-specific tasks must be arranged according to the priority (from high, middle, to low) and according to the value of (what to do first) setting .
- Now the space time between specific tasks must be detected allowing the other type of tasks to be added in it if the space time is sufficient for it .
- Then the duration of each non-specific task is compared with the durations which were detected to them in the appropriate one, if there is no appropriate duration for a task an exception message appears saying “there is no enough time for the task” and remove it .

Now after all these processes the user is able to show his today list and edit it by delaying tasks or removing tasks .

If the user has leisure time he can choose some suggested tasks from pre-inserted list of tasks, or he can reuse a previous list which he created before, or he can pick up a location from the map to visit .

The app provides a map that shows the friends who use this app too, if the user wants to join his friend’s task he will send him a notification that tells his friend that he wants to join him, if his friend accepted the request a notification is send back to the user tells him that your friend accepted your request, if his friend rejected the request a notification is send back to the user tells him sorry this task is private .

## 4.3 User Description

For our app there will be one type of users who have interest about the functionalities of this app and they will be able to use all features of it .

## 4.4 Requirements Gathering

In any technical project the requirements gathering is one of the important parts because it provides full understanding of what the project will do, so to gather requirements we have looked on other similar apps and decided what is the most important and interesting to put in ours, also to make new contribution we have done brain storming to add new features .

## 4.5 What is a Requirement ?

Requirements are the necessary attributes in the system, they are the statement that identifies a capability, characteristic or quality factor of the system in order to have value and utility to the users.

Once the requirements are defined, then the developers can initiate the other technical work including system design, development, testing, implementation, and operation .

## 4.6 System Requirements

For any system, there are functional and non-functional requirements to be considered while determining the requirements of the system, The functional requirements are user visible features that are typically initiated by stakeholders of the system, such as generate report, login, and signup. On the other hand, nonfunctional requirements are requirements that describe how the system will do what it is supposed to do, for example, security, reliability and maintainability.

### 4.6.1 Functional Requirements

The functions that the user can do to use the app are :

- The user shall be able to add new activity .
- The user shall be able to chose the app language .
- The user shall be able to determine the start and end time of the day .
- The user shall be able to show current list .
- The user shall be able to choose pre-inserted tasks as suggested tasks .
- The user shall be able to add task to the pre-inserted tasks list to be one of suggested tasks .
- The user shall be able to choose previous list to reuse it .
- The user shall be able to choose location as task to visit .
- The user shall be able to add a new location to the map .
- The user shall be able to choose a friend to join him his task from the map .
- The user shall be able to edit his today list .
- The app shall be able to send exception message if there are two or more tasks with the same priority and different duration .
- The user shall be able to determine which task to do first the shortest or the longest according to the previous point .
- The app shall be able to send exception message if there are two or more tasks with the same start time .
- The app shall be able to send exception message if the number of tasks and their times exceeded the total day duration .
- The app shall be able to send notification to the user’s friend to join him his task .
- The app shall be able to send notification to the user whether his friend accepted his request or rejected it .

### 4.6.2 Non-Functional Requirements

The non-functional requirements can be divided into several categories, the four basic categories are the operational, performance, security, and cultural requirements.

The operational requirements are the physical and technical environment in which the app will operate .

The performance requirements are the speed, capacity, and reliability of the app .

The security requirements are the determining who has authorization to access the app and under what circumstances .

The cultural requirements are the cultural factors that affect on the app .

#### 4.6.2.1 Operational Requirements

- The app should work on any handheld device .
- The app should work on any version of android OS only .

#### 4.6.2.2 Performance Requirements

- The application must be accurate in arranging tasks .
- The app must be accurate and fast when alerting for the task .

#### 4.6.2.3 Cultural Requirements

- The app should serve all interested users, so it must be delivered in English language because it is a universal language, and it must be delivered in Arabic language to serve the local area .

### 4.6.3 Hardware Requirements

Hardware requirements which were used during the development processing :

- Mobile devise with Android Operating system .

## 4.7 Development Requirements

To develop this app different tools were used :

- Draw.io : a website that provides all shapes for drawing all UML diagrams .
- Adobe Photoshop CS6
- Android Studio 1.5

## 4.8 System Analysis

This part contains the analysis of the functional and non-functional requirements using use-case diagrams, and use-cases details. In addition, the interactive behavior of the activities is analyzed using sequence diagrams and activity diagrams.

### 4.8.1 Use-case Diagram

Use case diagram is a representation of a user's interaction with the system and depicting the specifications of a use case, it illustrates the different types of users of a system and the various ways that they interact with the system .

Figure (4.1) demonstrates the use case diagram of our app, and as shown in the diagram the app has one main actor including android user which has different processes including all the functional requirements mentioned before .

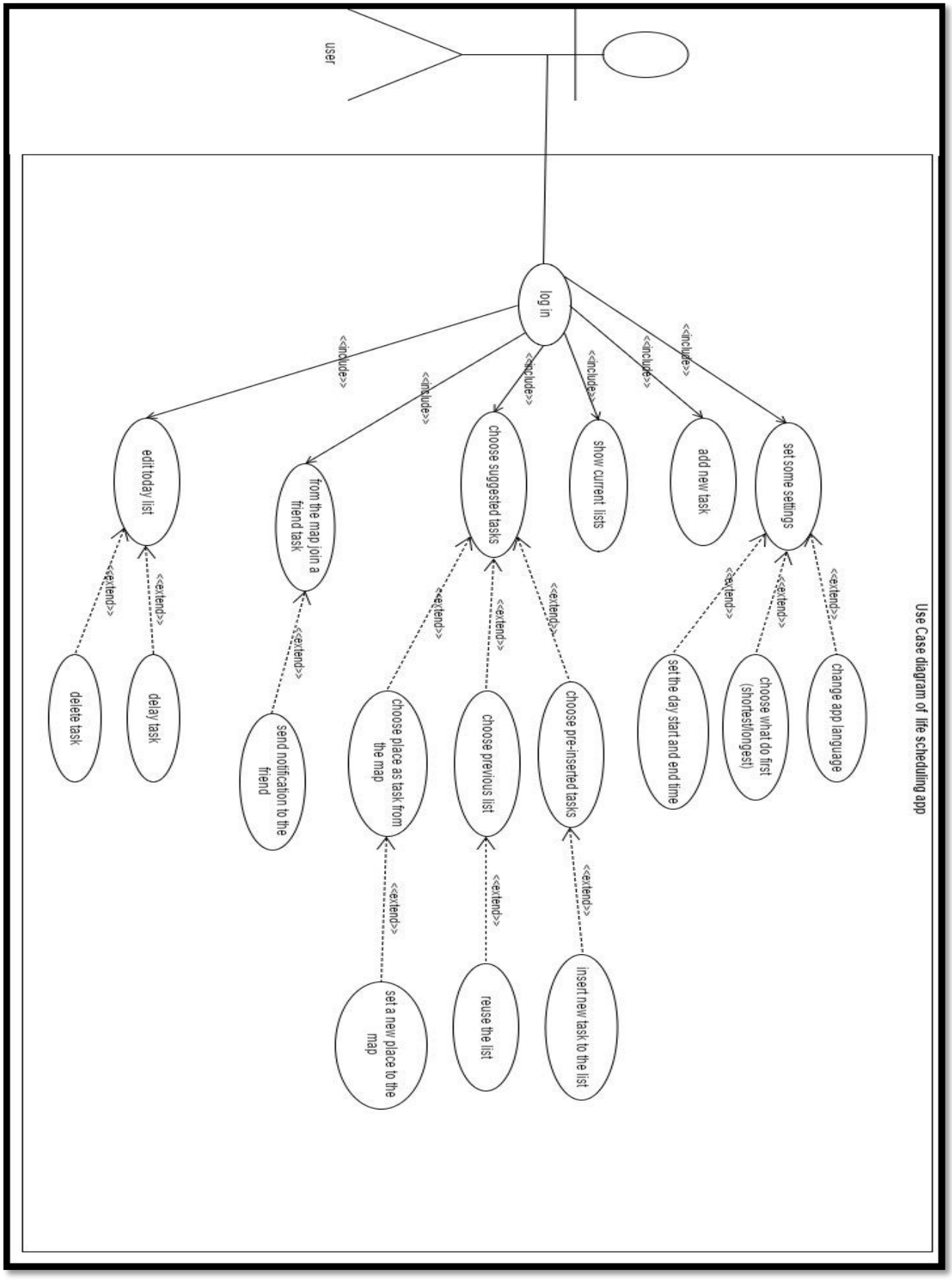



**Figure 4.1 Use-case Diagram for the app**

### 4.8.2 Scheduling Use Cases priorities

**Table 4-1 Scheduling Use Cases priorities**

| Use case name | priority | description | Preconditions | post conditions |
|---|---|---|---|---|
| Set some settings | medium | The user specifies some values needed | - | Values must be saved |
| Change app language | low | The user can specify the language of the app | - | - |
| Choose what to do first | medium | The user specifies which task to do first the longest or the shortest | - | The value must be saved |
| Set the day duration | medium | The user specifies the start and end time of the day | - | The value must be saved |
| Add new task | high | The user inserts new task with all values required | - | The must create a list to put the tasks in it |
| Show current list | low | The user is able to see his list | The current list must be saved | - |
| Choose suggested tasks | medium | The user can select some tasks from different sources | - | - |
| Choose from pre-inserted tasks | medium | The user can select tasks from provided from the app | - | The selected tasks must go to the current list |
| Choose previous lists | medium | The user can reuse a list he did create | All previous lists must be saved | - |
| Choose places from the map | medium | The user can pick up a location as a task | - | The selected location must go to the current list as task |

| | | | | |
|---|---|---|---|---|
| From the map join a friend | medium | The user can select a friend from the map and join him his task | - | - |
| Send notification to the friend | medium | If the user wants to join his friend task he sends a notification to him | Select a friend | A notification must arrive to the other user |
| Edit today list | high | The user can edit his list | There must be a current list | - |
| Delay task | high | The user can freeze a task for another time | There must be a current list | Rearrange the list |
| Delete task | high | The user can remove a task | There must be a current list | Rearrange the list |

### 4.8.3 Use cases scenario

**Table 4-2 Set settings use case scenario**

| **Use Case Name :** Set some settings | **ID :** UC-1 | **Priority :** Medium |
|---|---|---|
| **Actor :** App User | | |
| **Description :** the user specifies some values that are needed in the arranging algorithm like the day start and end time and which task the user prefers to do first the longest or the shortest if there are two or more tasks with same priority but with different duration, also the user can choose the language of the app . | | |
| **Preconditions :**<br>1- the user must have knowledge why these settings are needed . | | |
| **Normal Course :**<br>1- the user inserts the start and end time of the day.<br>2- the user selects the longest or the shortest task first .<br>3- the user choose the language of the app . | | |
| **Alternative Course :**<br>1- the user depends on the default values . | | |
| **Post-conditions :**<br>1- the values must be stored in the app . | | |

See other scenarios in Appendix H .

### 4.8.4 Sequence Diagram

The sequence diagram was used as a form of interaction diagram, which shows the interactions between the objects over time, During the analysis of the requirements, use-cases were extended to the next level by providing a more formal level of refinement, Accordingly, use-cases were refined into one or more sequence diagrams.

Figure (4.2) demonstrates the sequence diagram of adding new task, the user opens the app then the main screen which has buttons about all the functionality of the app appears then he selects the add task button to add his tasks, when he sets the task and all required values the app saves it and creates a list containing this task and all today's tasks in a new screen and when the user finishes adding tasks he selects OK button in the list screen and the app back to the main screen allowing the user to do something else .

Other sequence diagrams such as show, suggested tasks, join a friend, and settings are in appendix A .

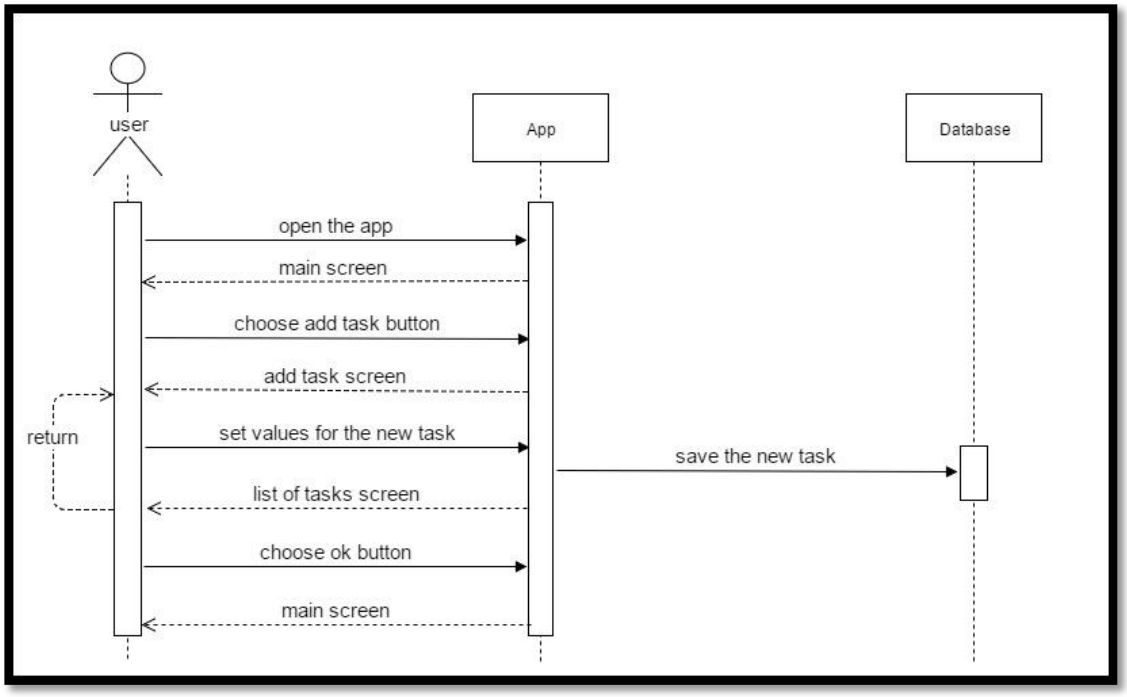



**Figure 4.2 Add task sequence diagram**

### 4.8.5 Activity Diagram

Activity diagram is a flow chart that represents the flow form one activity to another, the activity can be described as an operation of the system, therefore, the control flow is drawn from one operation to another, this flow can be sequential, branched or concurrent, it was used to show message flow from one activity to another .

Figure (4.3) demonstrates the activity diagram of how a user can add tasks, the user first opens the app and select add task button then insert the name of the task and

determine whether if it is with specific time (start/end time) or non-specific time (duration and priority) and then select add button to save the task and put it in a list, when the app puts the task in the list the arrangement algorithm works to put it in its place .

Other activity diagrams such as show, suggested tasks, join a friend, and settings are in appendix A .

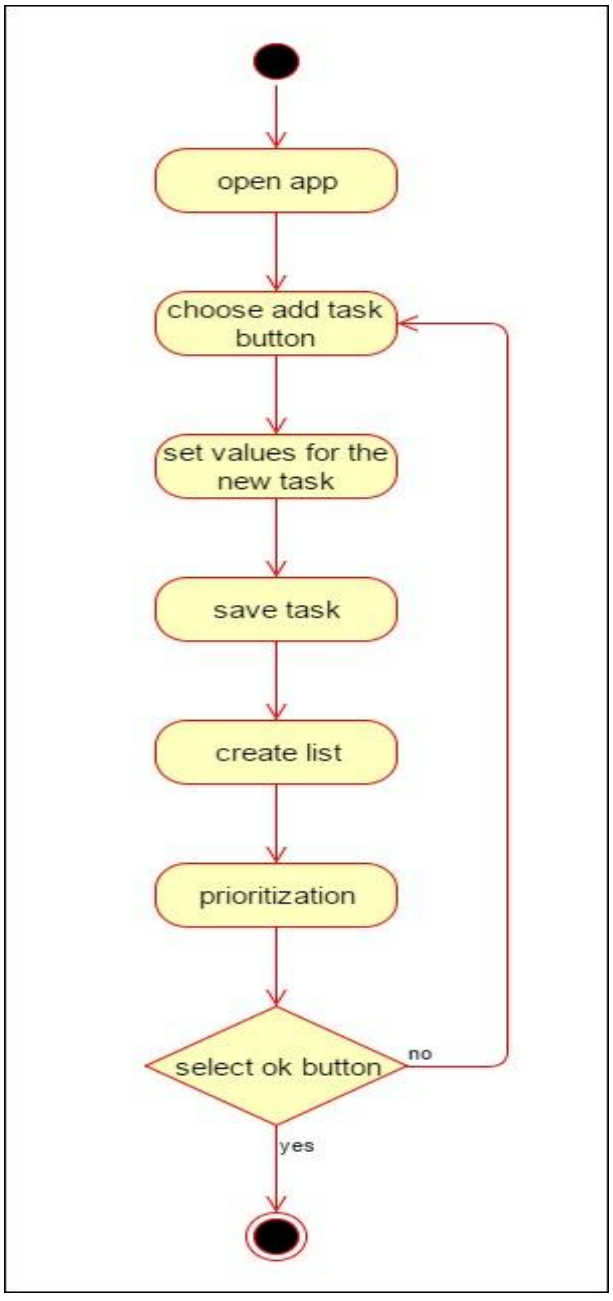



**Figure 4.3 Add task Activity Diagram**

## 4.9 Conclusion

In this chapter we have discussed a full description for the app and we have defined the functional and non functional requirement according to the requirements gathering way which we have used and we have drawn all UML diagrams required for developing this app,

# CHAPTER 5 :

# DESIGN AND IMPLEMENTATION

## 5.1 Introduction

This chapter demonstrates the design and implementation phases of the app, it describes the design strategies and Graphical User Interfaces, also it shows the class diagram, ER diagram and database schema of the app and system architecture .

## 5.2 Design strategies

Is a process to conceptualize the software requirements into software implementation, Software design takes the user requirements as challenges and tries to find optimum solution [31] .

While the software is being conceptualized, a plan is chalked out to find the best possible design for implementing the intended solution [31] .

Function oriented were used as a design strategy . In function-oriented design, the system is comprised of many smaller sub-systems known as functions. These functions are capable of performing significant task in the system. The system is considered as top view of all functions [31] .

This design mechanism divides the whole system into smaller functions, which provides means of abstraction by concealing the information and their operation.. These functional modules can share information among themselves by means of information passing and using information available globally [31] .

Another characteristic of functions is that when a program calls a function, the function changes the state of the program, which sometimes is not acceptable by other modules. Function oriented design works well where the system state does not matter and program/functions work on input rather than on a state [31] .

## 5.3 System Architecture

The app runs on Android operating system, which is a pile of software components, The main components of Android operating system architecture are Linux kernel, native libraries, Android Runtime, Application Framework and Applications as shown in figure (5.1) [32] .

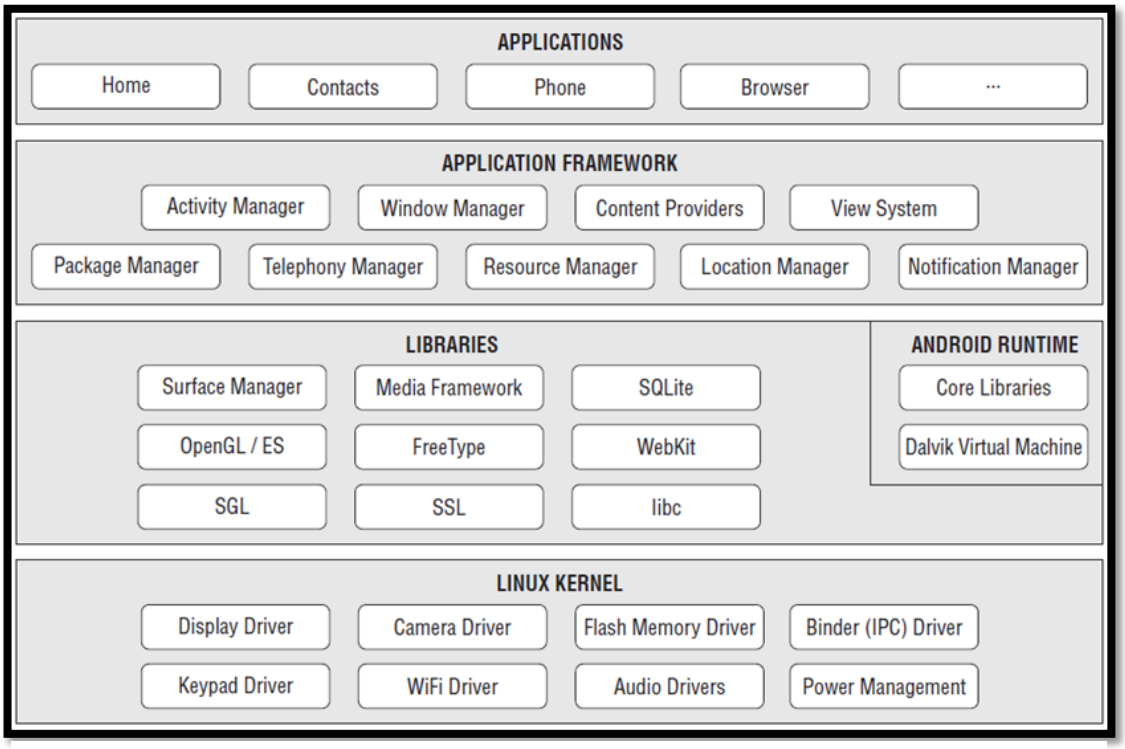



**Figure 5.1 Android Architecture**

**Linux kernel :** is at the bottom layer of the android architecture. The entire Android operating system is built on this layer. Like the main operating system, it provides the following functionalities process management, memory management, device management (e.g. camera, keypad and display). Android operating system interacts with the hardware of the device with this layer. It is also responsible for managing virtual memory, networking, drivers, and power management [33] .

**Native Libraries layer :** Android's native libraries comes after the Linux Kernel layer, it enables the device to handle different types of data that is specific to hardware. All these libraries are written in C or C++ language, these libraries are called through java interface, and here are some important native libraries are [33] :

- Surface manager : it is used for composing windows on the screen .
- SQLite : it is the database used in android for data storage. It is relational database and available to all applications .
- WebKit : it is the browser engine used to display HTML content .
- Media framework : Media framework provides playbacks and recording of various audio, video and picture formats .
- FreeType : responsible for Bitmap and Font Rendering .
- OpenGL/ES : used to render 2D or 3D graphics content to the screen .
- Libc : it contains System related C libraries .

**Android Runtime :** Android Runtime (ART) includes a set of core libraries that includes most of the functionality available in the core libraries of the Java programming language. It also includes the Dalvik Virtual Machine (DVM) and is located on the same level as the library layer. DVM is a type of Java Virtual Machine

used for running applications on Android device. The DVM enables every Android application to run in its own process, with its own instance of the DVM. It also allows multiple instance of virtual machine to be created simultaneously providing security, isolation, memory management and threading support. Unlike Java VM, which is process-based, Dalvik VM is register-base. Dalvik Virtual Machine run .dex files, which are created from the .class file by dx tool. DVM is optimized for low processing power and low memory environments [33] .

**Application Framework Layer :** it provides many higher-level services or major Application Programming Interfaces (APIs) to applications in the form of Java classes. Application developers are allowed to use these services in their applications. Developer's applications directly interact with these blocks, Some of the important blocks of application framework includes [33] :

- Activity manager : It manages the life cycle of applications .
- Content providers : It manages the data sharing between applications, manages how to access data from other applications .
- Telephony manager : It manages all voice call related functionalities .
- Location manager : It is used for Location management, using GPS or cell tower .
- Resource manager : Manage the various types of resources used in Application .

**Applications Layer :** is at the top of the Android architecture. Some applications come pre-installed with every device, such as SMS client App, Dialer, Web browser and Contact manager. A developer can write his own application and can replace it with the existing application [33] .

## 5.4 Development Environment

Development environment refers to the hardware and software tools that the developer uses to build any system, as the technology improves and user expectations grow .

To build our app we have used Android studio, which is the official integrated development environment (IDE) for the Android platform provided by Google .

Android studio allows the developer to run and debug apps on virtual device called emulator or on real device directly, we have used a real device to run and test our app during implementation phase .

## 5.5 Class Diagram

To illustrate the relationships and source code dependencies among classes, class diagram was developed, the class defines the methods and variables in an object, which is a specific entity in a program or the unit of code representing that entity. Figure (5.2) illustrates the class diagram of the app .

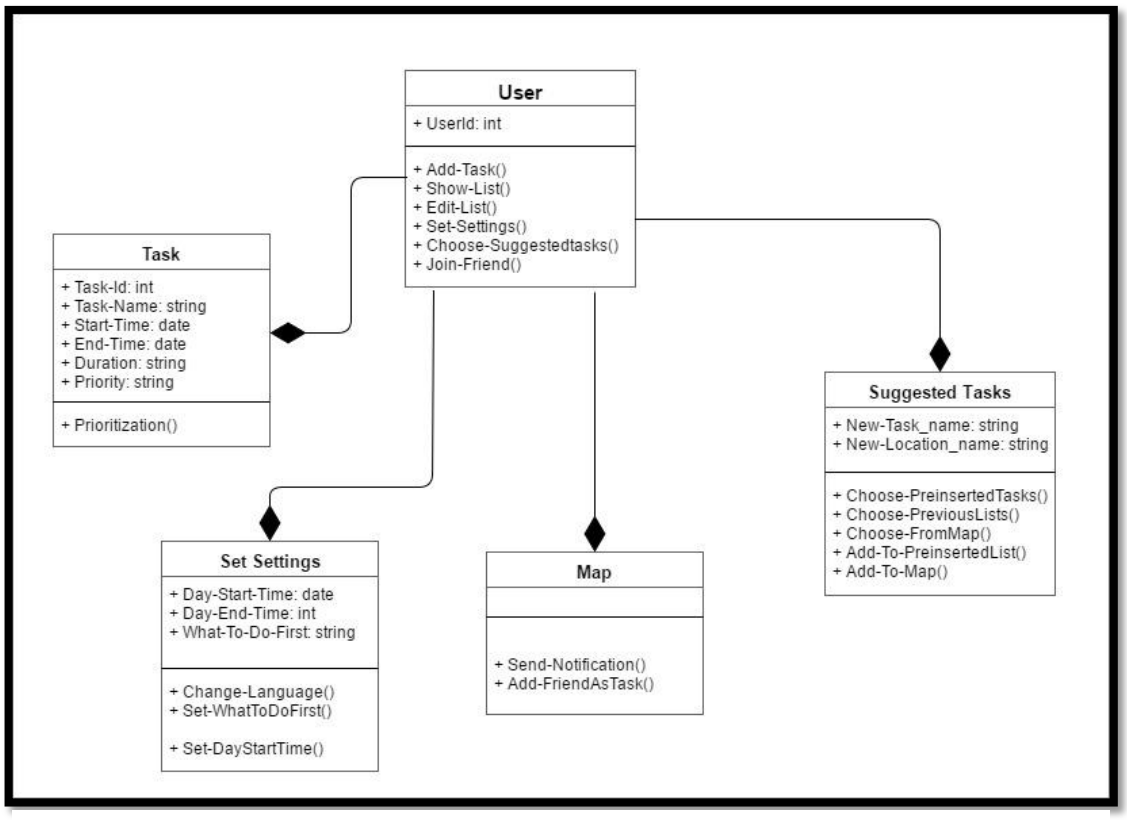



**Figure 5.2 App's class diagram**

In this diagram, there are five classes, which are arranged in way that represents the common interactions and connections, the class diagram looks like a flowchart in which classes are represented as boxes, each box has three rectangles inside it, the top rectangle contains the name of the class, the middle rectangle contains the attributes of the class and the lower rectangle contains the methods of the class, the lines between the classes define the relationships between them. These relationships include aggregation, composition and association .

For example the class of task entity contains attributes including task id, task name, start time, end time, duration, and priority with different data type like integer, string, and date, the method related to this class is the prioritization .

## 5.6 Entity Relationship Diagram

An entity relationship diagram is a graphical representation of an information system that shows the relationship between people, objects, places, concepts or events within that system [34] .

An ERD is a data modeling technique that can help define business processes and can be used as the foundation for a relational database [34] .

Figure (5.3) represents the ER diagram of the app, as shown in the figure, there are four database tables, also it represents the relationship between these tables, for example today list can have more than one specific task and non-specific task

In addition, each database table has several attributes related to it, for example as shown in the figure the specific table  includes task ID, task name, start time and end time .

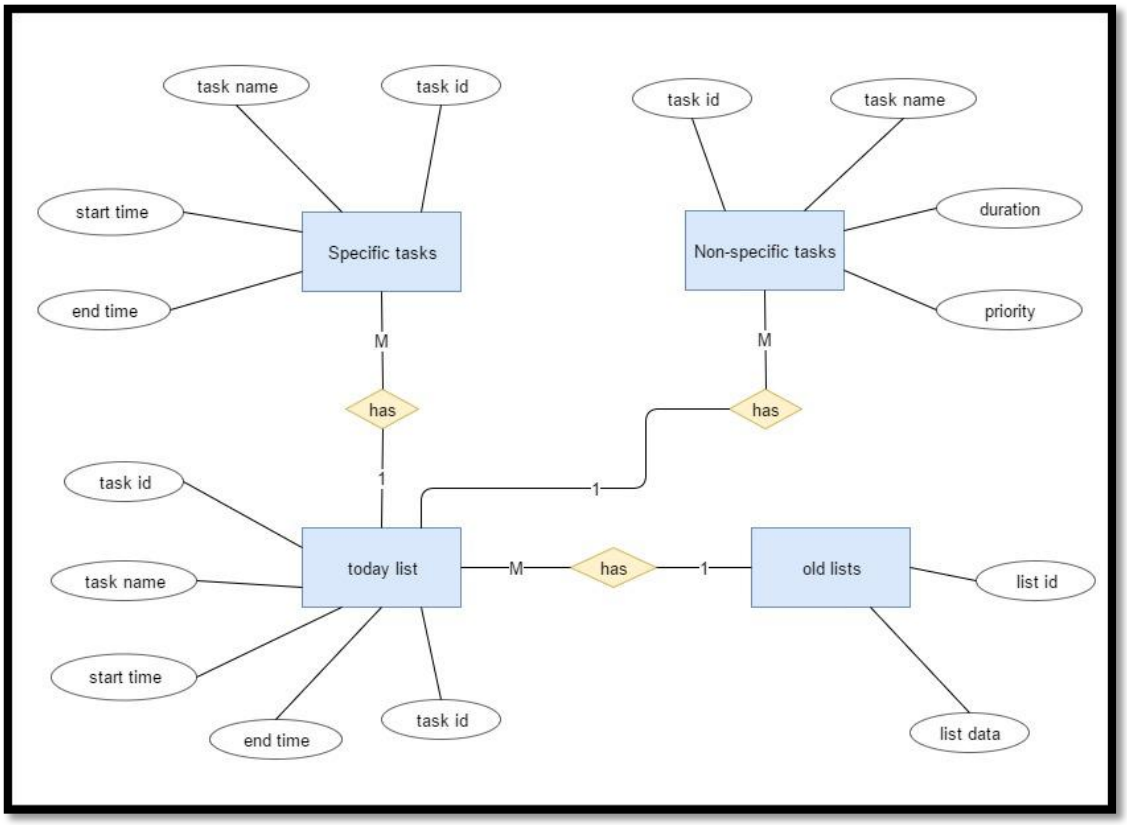



**Figure 5.3 ERD of the app**

## 5.7 Database Schema

A database schema is the skeleton structure that represents the logical view of the entire database, it defines how the data is organized and how the relations between them are associated, also it formulates all the constraints that are to be applied on the data .

The app has four database tables as shown in figure (5.4), each table contains an ID and some different attributes related to different functions in the app, for example the non-specific table includes attributes such as task name, duration, priority, and task id (which is the foreign key from table today-list ) .

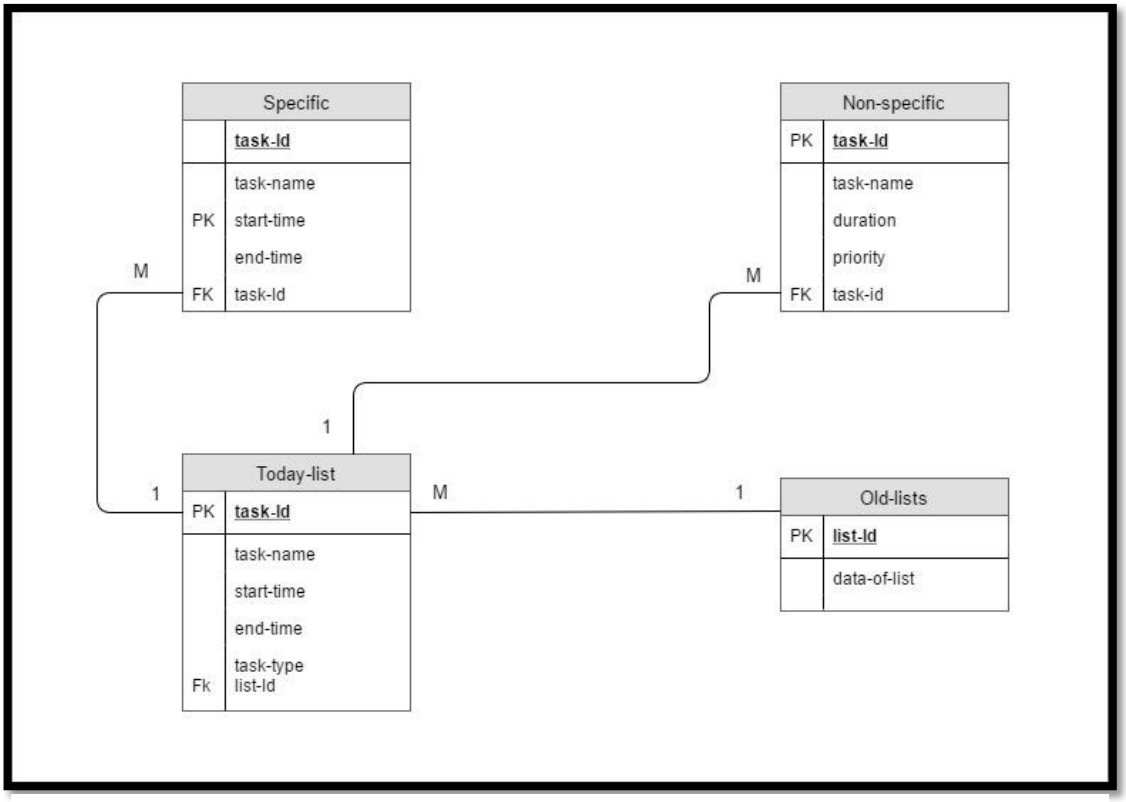



**Figure 5.4 Database schema for the app**

## 5.8 Graphical User Interfaces

All the interfaces of the app are represented in this section .

The beginning screen :

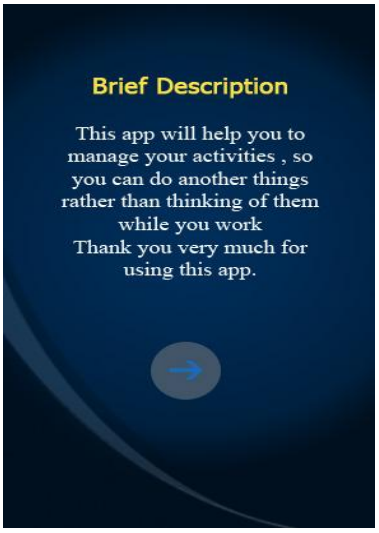



**Figure 5.5 beginning screen**

Main screen which has the buttons for each function of this app :


**Figure 5.6 Main screen**

Settings option :

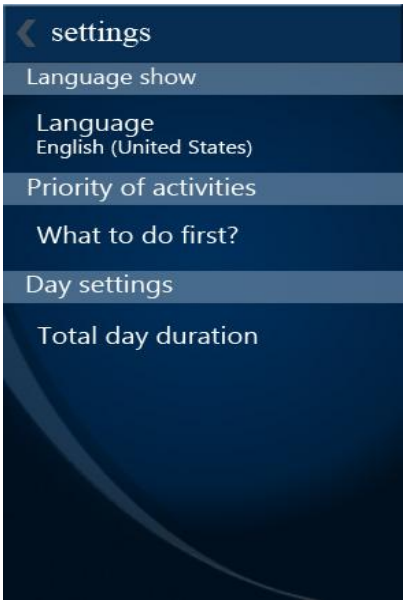



**Figure 5.7 Settings option screen 1**

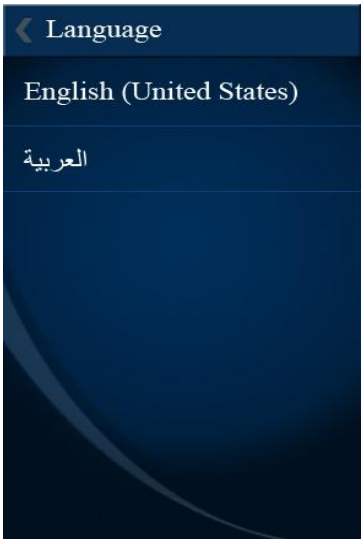



**Figure 5.10 Settings option screen 1.1**


**Figure 5.9 Settings option screen 1.2**


**Figure 5.8 Settings option screen 1.3**

Add task screen and list screen which is created after adding a task :

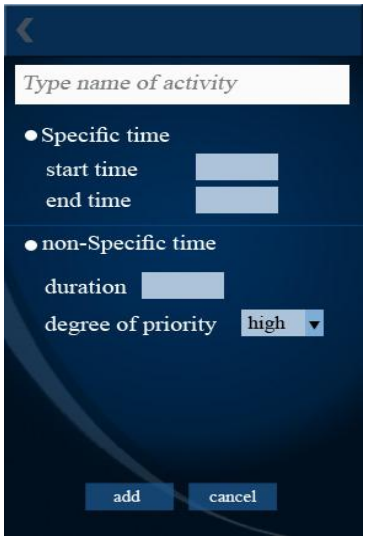



**Figure 5.12 Add task screen**

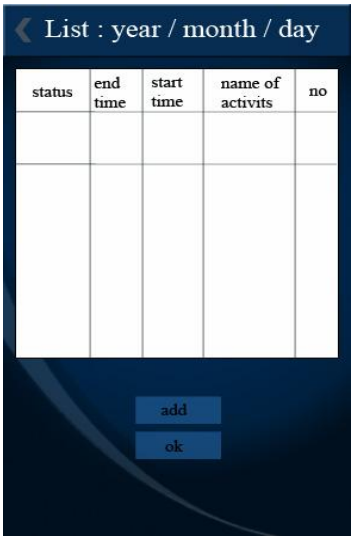



**Figure 5.11 List screen**

Show today list screen with share the tasks on map option :

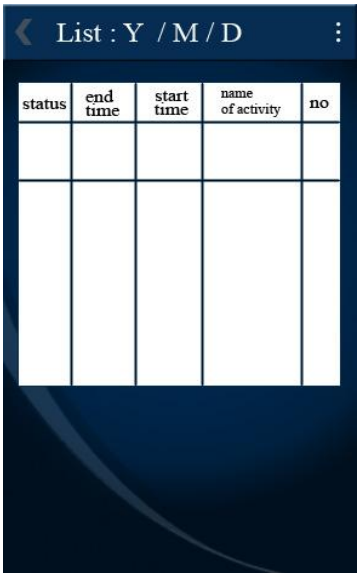



**Figure 5.13 Show today list screen**

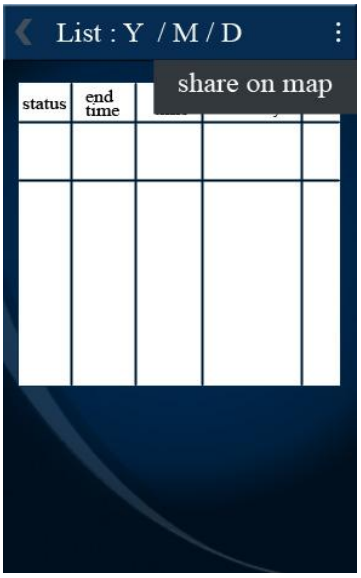



**Figure 5.14 Share on map option screen**

Suggested tasks screen :


**Figure 5.15 Suggested tasks screen**

Choose tasks screen with adding new suggested task option :


**Figure 5.18 Choose tasks screen**


**Figure 5.17 Insert new task option screen 1**


**Figure 5.16 Insert new task option screen 1.1**

Choose previous list screen :


**Figure 5.20 Choose previous list screen 1**

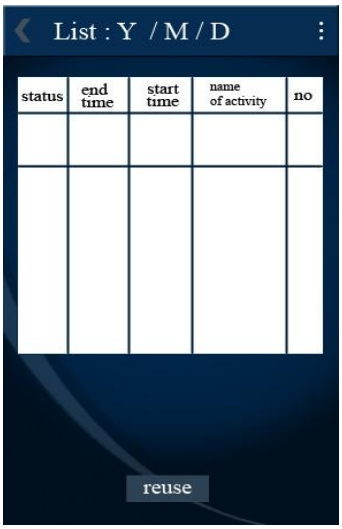



**Figure 5.19 Choose previous list screen 1.1**

Choose from map screen :


**Figure 5.23 choose from map screen 1**

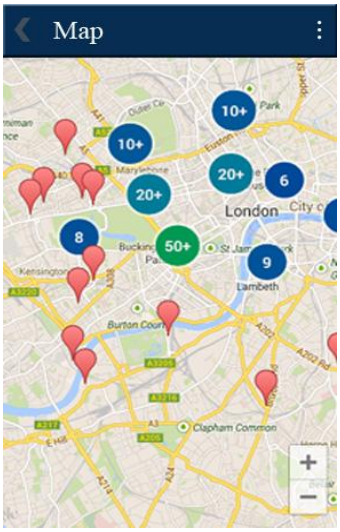



**Figure 5.22 choose from map screen 1.1**


**Figure 5.21 choose from map screen 1.2**

Join a friend screen :

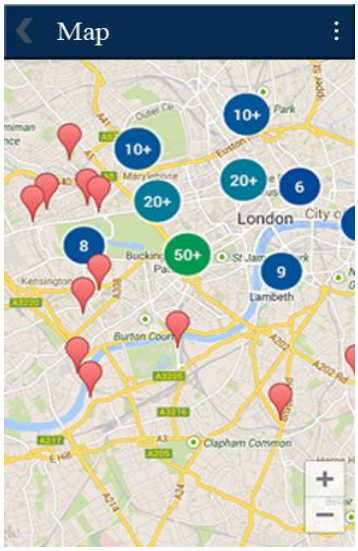



**Figure 5.25 Join a friend screen 1**

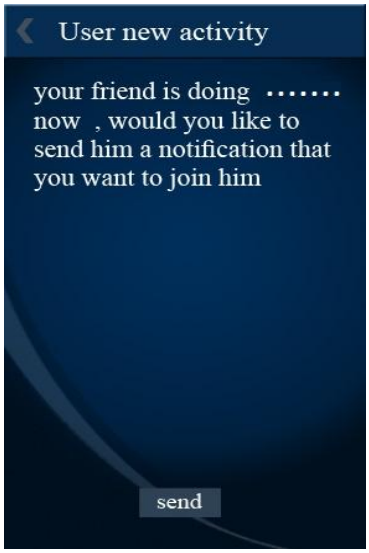



**Figure 5.24 Join a friend screen 1.1**

Edit screen :


**Figure 5.26 Edit screen**

## 5.9 Conclusion

This chapter discussed the designed strategy we have used to develop our app, and the system architecture of the operating system which our app will operate on, and the hardware and software tools we have used to develop the app. A class diagram was drawn to see the logic relationships between entities, and the ERD to help building the tables in the database, and the database schema to illustrate the relationships between tables in the database, and we have developed the interfaces of the app which the user will use .

# CHAPTER 6 : TESTING AND EVALUATION

## 6.1 Introduction

This chapter demonstrates the testing and evaluation phases, in the testing phase functionality testing, usability testing, database testing, interface testing, and security testing are performed. In addition, the evaluation was performed through user evaluation using a questionnaire .

## 6.2 Purpose of software testing

Testing has different goals and objectives, the major objectives of testing are as follows [35] :

- Finding defects which may get created by the programmer while developing the software .
- Providing information about the level of quality .
- To prevent defects .
- To make sure that the end result meets the business and user requirements .
- To ensure that the software satisfies the BRS (Business Requirement Specification) and SRS (System Requirement Specifications) .
- To gain the confidence of the customers by providing them a quality product .

Software testing helps in finalizing the software application or product against business and user requirements. It is very important to have good test coverage in order to test the software application completely and make sure that it is performing well and as the specifications [35] .

While determining the test coverage the test cases should be designed well with maximum possibilities of finding the errors or bugs, the test cases should be very effective, this objective can be measured by the number of defects reported per test cases, higher the number of the defects reported the more effective are the test cases [35] .

Software testing makes sure that the testing is being done properly and hence the system is ready for use, Good coverage means that the testing has been done to cover the various areas like functionality of the application, compatibility of the application with the OS, hardware and different types of browsers, performance testing to test the performance of the application and load testing to make sure that the system is reliable and should not crash or there should not be any blocking issues [35] .

It also determines that the application can be deployed easily to the machine and without any resistance. Hence the application is easy to install, learn and use [35] .

## 6.3 Stages of tests

Testing levels are used to identify missing areas and prevent overlap and repetition between the development life cycle phases, in software development life cycle models there are defined phases like requirement gathering and analysis, design, coding or implementation, testing and deployment, each phase goes through the testing, so there are various levels of testing [36] :

- **Unit testing :** It is basically done by the developers to make sure that their code is working fine and meet the user specifications, they test their piece of

code which they have written like classes, functions, interfaces and procedures .

- **Component testing :** It is also called as module testing. The basic difference between the unit testing and component testing is in unit testing the developers test their piece of code but in component testing the whole component is tested .
- **Integration testing :** Integration testing is done when two modules are integrated, in order to test the behavior and functionality of both the modules after integration .
- **Component integration testing :** when both modules or components are integrated then the testing done is called as Component integration testing, this testing is basically done to ensure that the code should not break after integrating the two modules .
- **System integration testing :** System integration testing (SIT) is a testing where testers basically test that in the same environment all the related systems should maintain data integrity and can operate in coordination with other systems .
- **System testing :** In system testing the testers basically test the compatibility of the application with the system .
- **Acceptance testing :** Acceptance testing are basically done to ensure that the requirements of the specification are met .
- **Alpha testing :** Alpha testing is done at the developers site, it is done at the end of the development process .
- **Beta testing :** Beta testing is done at the customers site, it is done just before the launch of the product .
- **White box testing :** This is also known as clear box testing, glass box testing, translucent box testing or structural testing, It uses the internal perspective of the system and then designs test cases based on this internal structure. Basically the code itself and all the conditions, statements and paths along with it are tested. Programming skills are required for noticing all the paths through the software. White box testing does many things such as analyzing the data flow, control flow, information flow and coding practices .
- ***Black box testing :*** Which is the methodology that we have used to test our app. The black-box approach is a testing method in which test data are derived from the specified functional requirements without regard to the final program structure. It is also termed data-driven, input/output driven, or requirements-based testing. In testing, various inputs are exercised and the outputs are compared against specification to validate the correctness. All test cases are derived from the specification. No implementation details of the code are considered .

## 6.4 Why software testing

Software Testing is necessary because we all make mistakes, and some of those mistakes are unimportant, but some of them are expensive or dangerous, we need to check everything and anything we produce because things can always go wrong [37].

There are several reasons which clearly tells us why Software Testing is important and what are the major things that we should consider while testing of any product or application, software testing is very important because of the following reasons [37] :

- Software testing is really required to point out the defects and errors that were made during the development phases .
- It is essential since it makes sure of the Customer's reliability and their satisfaction in the application .
- It is very important to ensure the Quality of the product .
- Testing is necessary in order to provide the facilities to the customers like the delivery of high quality product or software application which requires lower maintenance cost and hence results into more accurate, consistent and reliable results .
- Testing is required for an effective performance of software application or product .
- It is important to ensure that the application should not result into any failures because it can be very expensive in the future or in the later stages of the development .

## 6.5 Checklist for Life scheduler app testing

This checklist used to check the following aspects for the project :

- Functional testing .
- Usability testing .
- Database testing .
- Interface testing .
- Security testing .

### 6.5.1 Functional testing

The remark (A) denotes to Accepted, and the remark (NE) denotes to Need To Improve .

**Table 6-1 Functional testing**

| # | Description | OK/NOK | Remark |
|---|---|---|---|
| 1.1 | Test the app should not display the error message for optional fields. | OK | A |
| 1.2 | Test the numeric fields should not accept the alphabets and proper error message should display. | OK | A |
| 1.3 | Test for negative numbers if allowed for numeric fields. | OK | A |
| 1.4 | Test the max length of every field to ensure the data is not truncated. | OK | A |
| 1.5 | Test the pop up message ("This field is limited to 500 characters") should display if the data reaches the maximum size of the field. | OK | A |

| | | | |
|---|---|---|---|
| 1.6 | Test that a confirmation message should display for update and delete operations. | OK | A |
| 1.7 | Test all input fields for special characters. | OK | A |
| 1.8 | Test the sorting functionality. | OK | A |
| 1.9 | Test the functionality of the buttons available. | OK | A |
| 1.10 | Test all the data inside combo/list box is arranged in an appropriate order. | OK | A |
| 1.11 | Test the navigation between screens | OK | A |
| 1.12 | Test all the mandatory fields should be validated. | OK | A |
| 1.13 | Does entered data is saved to the database and value gets saved fully to the database. | OK | A |
| 1.14 | Does numbers of days and months validated and do not cause errors/miscalculations? | OK | A |
| 1.15 | Does Lowest and highest values are handled correctly? | OK | A |
| 1.16 | Does numeric fields with a blank in position are processed? | OK | A |
| 1.17 | Does that both + and - values are correctly processed? | OK | A |

### 6.5.2 Usability testing

**Table 6-2 Usability testing**

| # | Description | OK/NOK | Remark |
|---|---|---|---|
| 2.1 | All screens' content should be correct without any spelling or grammatical errors . | OK | A |
| 2.2 | All fonts should be same as per the requirements. | OK | A |
| 2.3 | All the text should be properly aligned. | OK | A |
| 2.4 | All the error messages should be correct without any spelling or grammatical errors . | OK | A |
| 2.5 | All the fields should be properly aligned. | OK | A |
| 2.6 | Enough space should be provided between field . | OK | A |
| 2.7 | All the buttons should be in a standard format and size. | OK | A |

| | | | |
|---|---|---|---|
| 2.8 | Confirmation message should be displayed for any kind of delay and delete operation. | OK | A |
| 2.9 | Check the end user can run the app without frustration. | OK | A |
| 2.10 | If there is an error message on add, the information filled by the user should be there. | OK | A |
| 2.11 | Title of each screen should be displayed in each screen . | OK | A |
| 2.12 | Check if the dropdown data is not truncated due to the field size. | OK | A |
| 2.13 | Verify the fonts are usable on any device . | OK | A |
| 2.14 | Are the field backgrounds with correct color? | Ok | A |
| 2.15 | Are the screen and field colors adjusted correctly for non-editable mode? | OK | A |
| 2.16 | Are all the buttons in standard format and size? | OK | A |
| 2.17 | Is the general screen background with correct color? | OK | A |

### 6.5.3 Database testing

To perform the Database testing the DB unit testing tool was used, which is a JUnit framework allowing to put a database in a know state and to perform assertion against its content [38] .

From this framework we used DBTestCase subclass .

**Table 6-3 Database testing**

| # | Description | OK/NOK | Remark |
|---|---|---|---|
| 3.1 | Verify the database name: The database name should match with the specifications. | OK | A |
| 3.2 | Verify the Tables, columns, column types and defaults: All things should match with the specifications. | OK | A |
| 3.3 | Verify whether the column allows a null or not. | OK | A |
| 3.4 | Verify the Primary and foreign key of each table. | OK | A |
| 3.5 | Verify the parameter names, types and number of parameters. | OK | A |
| 3.6 | Test the parameters if they are required or not. | OK | A |

| | | | |
|---|---|---|---|
| 3.7 | Verify the data gets properly saved into the database after the each add. | OK | A |
| 3.8 | Verify the data if the DML (Update, delete and insert) operations are performed. | OK | A |
| 3.9 | Check the length of every field . | OK | A |
| 3.10 | Verify the database size. Also test the response time of each query executed. | Ok | A |
| 3.11 | Verify the data displayed on the screen and make sure it is same in the database . | OK | A |
| 3.12 | Verify the data validity by inserting the invalid data in the database. | OK | A |

### 6.5.4 Interface testing

**Table 6-4 Interface testing**

| # | Description | OK/NOK | Remark |
|---|---|---|---|
| 4.1 | To keep controls as unobtrusive as possible for instance by fading them out if they are not used for a while . | OK | A |
| 4.2 | Make it possible for users to go back to a previous screen for instance by adding a back or cancel button . | OK | A |
| 4.3 | The main function of the app should be apparent immediately. It should speak for itself . | OK | A |
| 4.4 | Use at most one action on the screen that is highlighted as the most likely for the user. (Example: in iOS a blue button represents the default or most likely action). | OK | A |
| 4.5 | Minimize user actions by using a picker or a table view where users can select a certain choice over a data entry field where users have to type a choice . | OK | A |
| 4.6 | In the app, the user should not be able to store files locally, outside the app sandbox . | OK | A |
| 4.7 | In the app, the user should not be exposed to the permissions of a specific file . | OK | A |
| 4.8 | If there is a long list of data to scroll through, provide a search option above the list . | OK | A |

| | | | |
|---|---|---|---|
| 4.9 | If performance is slow, indicate a progress status icon (“Loading…”), preferably with specific message . | OK | A |
| 4.10 | In case of ‘live’ filtering of data while the user enters his search query, verify the performance . | OK | A |
| 4.11 | The appearance of buttons that perform standard actions are not altered in the app (for instance: refresh, organize, trash, Reply, etc.) | OK | A |
| 4.12 | Do not use standard buttons for other functions that they are normally used for . | OK | A |
| 4.13 | The app should respond to all changes in device orientation, as per the design . | OK | A |
| 4.14 | Tap able elements should be about 7x7 mm in size, using the pixel density of the target device you can calculate the amount of pixels . | OK | A |
| 4.15 | Do not redefine gestures in your app that have a standard meaning (example: swiping from top to bottom enables the notification centre) . | OK | A |
| 4.16 | Requirement to login is delayed in the app as long as possible | OK | A |
| 4.17 | If the app is stopped at an unexpected time, user data should be saved locally and available at start-up . | OK | A |
| 4.18 | Keyboard adjusts to expected input (for instance numbers/letters when expected) . | OK | A |
| 4.19 | Are inactive buttons clearly distinguished from active buttons? | OK | A |

### 6.5.5 Security testing

**Table 6-5 Security testing**

| # | Description | OK/NOK | Remark |
|---|---|---|---|
| 5.1 | Verify the error messages should not display any important information. | OK | A |
| 5.2 | Verify the “View Source code” option is disabled and should not be visible to the user. | OK | A |
| 5.3 | Verify if, any functionality is not working, the app should not display any application, server, or database information. Instead, it should display | Ok | A |

| | | | |
|---|---|---|---|
| | the custom error message. | | |
| 5.4 | Verify the SQL injection attacks. | OK | A |
| 5.5 | Verify the app does not show user data online before his permission . | OK | A |

## 6.6 Test cases

To apply the black box testing Android studio provides functions for testing that generates reports about the results, and here are the summarization for the reports .

**Table 6-6 Add task test case**

| **TC1-Add task screen-check successful of adding a task** |
|---|
| **Description :** the app user should successfully add his task . |
| **Preconditions :** the user must be aware about the values that will contribute when adding the task that are the start and time of the day and the longest or shortest task first . |
| **Assumptions :** a supported device is being used . |
| **Test steps :**<br>1- open the app .<br>2- choose Add Task button .<br>3- in the task name field insert the name of the task .<br>4- choose whether the task will be with specific time or with non-specific time buttons .<br>5- if it was with specific time insert the start and end time in the fields .<br>6- if it was with non-specific time insert the duration of the task in the field and select the priority of it from the drop-down list .<br>7- click Add button . |
| **Expected result :** the screen of the today list should be displayed containing the list with the added task showing its number, name, start time, and end time . |

See other test cases in Appendix I .

## 6.7 Evaluation

Evaluation is an essential part in any project, because evaluation involves assessing the strengths and weaknesses of programs, policies, personnel, products, organizations, and etc to improve their efficiency and effectiveness .

There are several techniques to evaluate any system, but the most common way is by conducting a user questionnaire, which we have used to evaluate our app .

### 6.7.1 Usability evaluation

The usability is considered as an important attribute of software quality and is referred to as the efficiency, effectiveness and satisfaction with which users can perform tasks with a tool, the term is used to describe the quality of a user's experience when interacting with a system whether a website, a software application, mobile technology, or any other human operated device.

A usable system is the one that enables users to perform their job effectively and efficiently [39] .

Evaluating usability is considered an essential part of the system development process and there are different methods to support the human factors professionally .

Traditionally, the concept of usability has been defined in different ways, basically usability is equated to terms such as 'ease of use' or 'user-friendliness', without formal definition of the properties of the construct .

### 6.7.2 End user evaluation

To apply the usability evaluation a user questionnaire was conducted , a sample of 50 random selected participants response as shown in table (6-6) and were asked to use the app .

The questionnaire was divided into two parts. The first part includes general information about the participants including, their gender, age, and specialization. The second part was used for the evaluation of the app and it includes four groups, each group evaluates the participant’s opinion about specific consideration about the app, the questionnaire data was then entered, filtered, coded and analyzed using SPSS .

The template of the questionnaire is in Appendix F and G .

As shown in table (6-6) 52% of respondents were male and 48% female as shown in figure (6.1). Majority of respondents (52%) were aged less than 20 years old as shown in figure (6.2). science studies were 38%. literature studies were only 18% as shown in figure (6.3). The end-users were also chosen according to their experience in using mobile apps, This was considered as important criteria in the selection of the users because the users of the app must be familiar with this technology, half of the participants uses mobile Apps since more than 10 years. The second half of them are distributed among those whose experience between 5-10 years and less than 5 years as shown in figure (6.4) .

**Table 6-7 Demographic data**

| Measure | Item | N | (%) | Cumulative(%) |
|---|---|---|---|---|
| Gender | Male | 26 | 52 | 52 |
| | Female | 24 | 48 | 100 |
| specialization | science | 19 | 38 | 38 |
| | Business | 10 | 20 | 58 |

| | | | | |
|---|---|---|---|---|
| | Literature | 9 | 18 | 76 |
| | Others | 12 | 24 | 100 |
| Age | Less than 20 | 26 | 52 | 52 |
| | 20-25 | 22 | 44 | 96 |
| | 26-30 | 2 | 4 | 100 |
| Experience | Less than 5 years | 6 | 12 | 12 |
| | 5-10 years | 19 | 38 | 50 |
| | More than 10year | 25 | 50 | 100 |

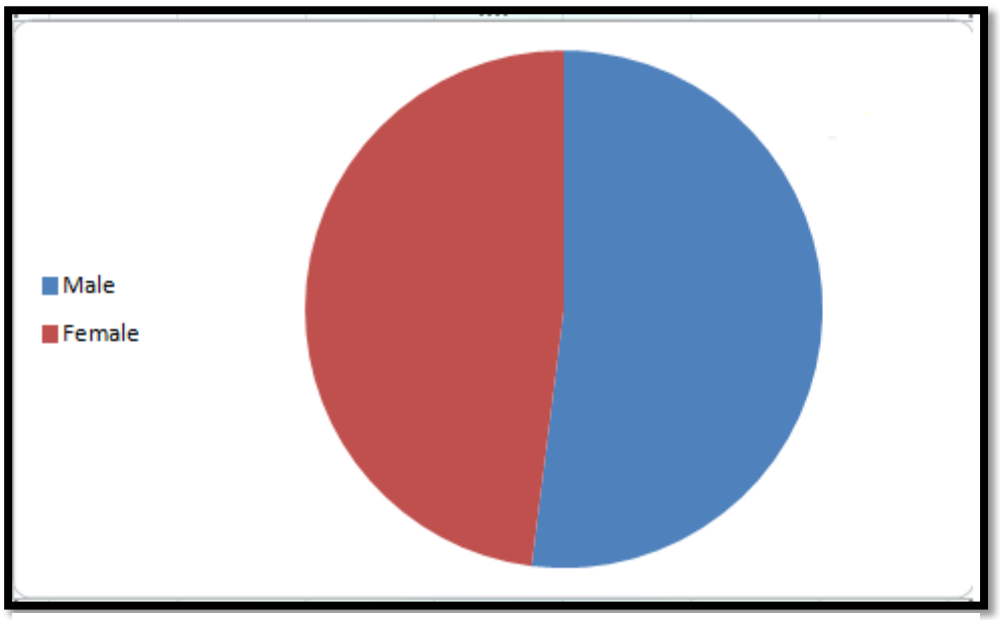



**Figure 6.1 Distribution of end-users according to gender**

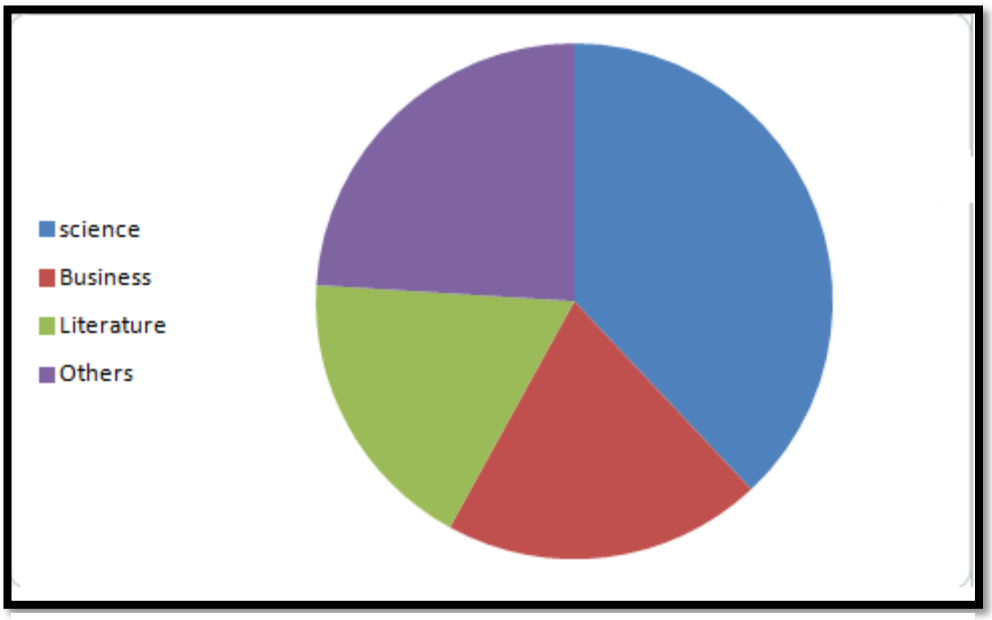



**Figure 6.3 Distribution of end-users according to specialization**

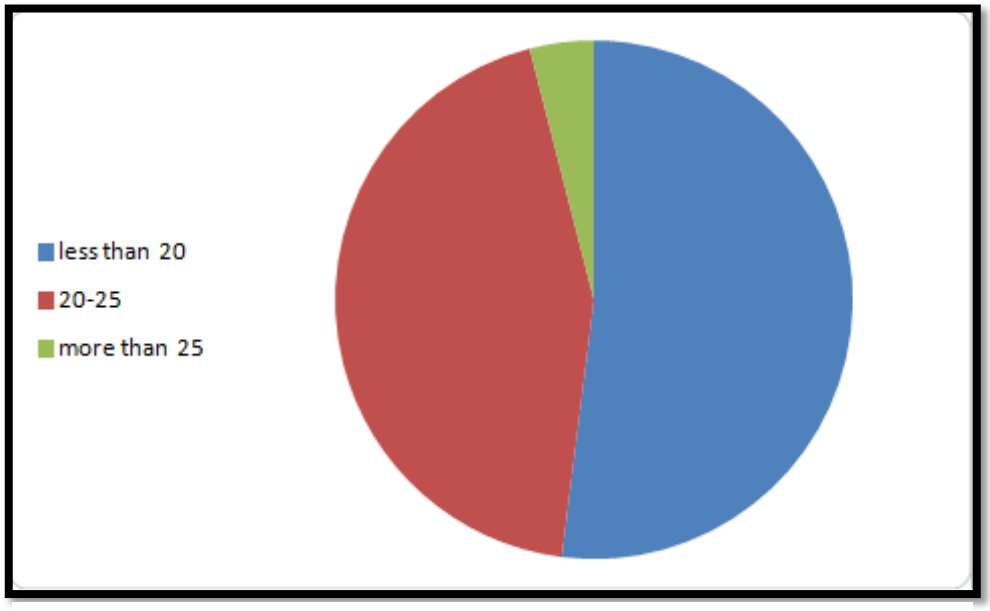



**Figure 6.2 Distribution of end-users according to age group**

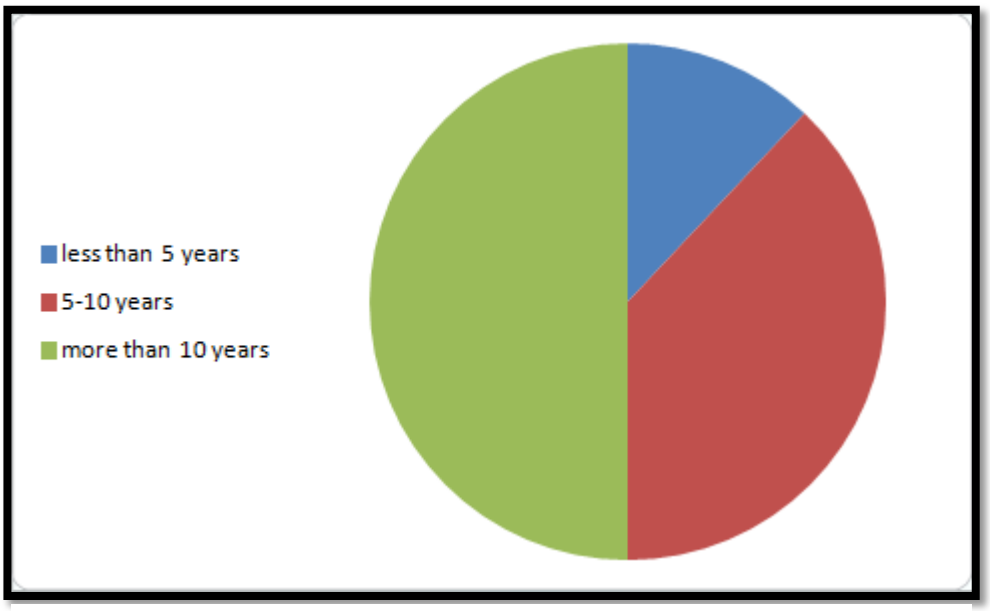



**Figure 6.4 Distribution of end-users according to their experience in mobile Apps**

To determine the reliability of the scale a Cronbach's alpha analysis was used, which is the most common measure of internal consistency ("reliability") and most commonly used when you have multiple Likert questions in a survey/questionnaire that form a scale .

Table (6-7) presents the Cronbach alpha (α) value for each measure. All measures have Cronbach alpha of greater than (0.7) except the learnability it was (0.253), these measures satisfy the internal reliability criterion .

**Table 6-8 Cronbach's alpha**

| Variable | Number of items | Alpha (α) |
|---|---|---|
| Perceived usefulness | 6 | .859 |
| perceived ease of use | 6 | .885 |
| learnability | 3 | .253 |
| future use | 5 | .837 |

The ranges of five point Likert-scales were categorized into equal sized categories of low, moderate and high. Therefore, score of less than 2.33 [4/3 +lowest value (1)] are considered low; scores of 3.67 [highest value (5)-4/3] are considered high and those in between are considered moderate. Five of measures with high means are bolded which indicate that most of the participants highly agreed on perceived usefulness, perceived ease of use, learn ability and future use . Overall, the results indicate that the participants agreed that the app has appropriate usability .

Figure (6.5) shows the first group in the questionnaire which focused on evaluating the Perceived Usefulness (PU) of the app, about 6% of users strongly agreed that the app enabled them to accomplish their tasks more quickly and 28% agreed, 28% were neutral, 16% disagreed, and 22% strongly disagreed. About 18% of users strongly agreed that the app helped them to improve their performance in their life and 28% agreed, 22% were neutral, 20% disagreed, and 12% strongly disagreed. About 12% of users strongly agreed that the app increased their productivity in their life and 32% agreed, 28% were neutral, 18% disagreed, and 10% strongly disagreed. About 14% of users strongly agreed that the app enhanced their effectiveness in their work and 30% agreed, 32% were neutral, and 12% disagreed and strongly disagreed. About 12% of users strongly agreed that the app helped them to engage in the social life and 34% agreed, 32% were neutral, 12% disagreed, and 10% strongly disagreed. About 18% of users strongly agreed that the app was useful for them and 36% agreed, 16% were neutral, 24% disagreed, and 6% strongly disagreed. From this analysis it is obvious that most of users found the app useful for them but it still need some improvements .

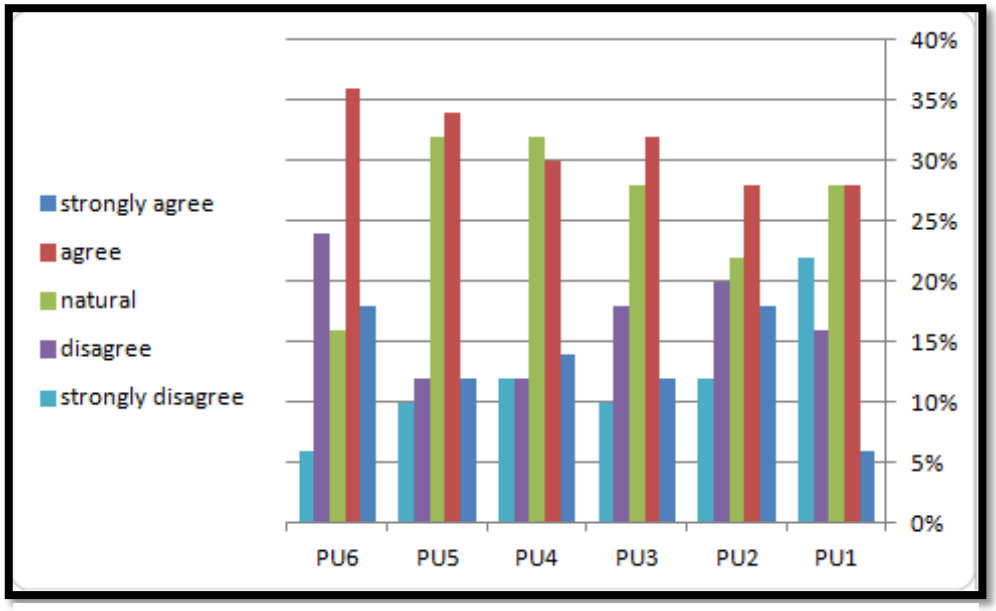



**Figure 6.5 Perceived Usefulness questions' frequency distribution**

Figure (6.6) shows the second group in the questionnaire which focused on evaluating the Perceived Ease of Use (PEOU) of the app. About 20% of users strongly agreed and agreed that the app was easy to learn and 26% were neutral, 24% disagreed, and 10% strongly disagreed. About 16.3% of users strongly agreed that the app was easy to use and 30.6% agreed, 26.5% were neutral, 20.4% disagreed, and 6.1% strongly disagreed. About 10.2% of users strongly agreed that the interaction with the app was clear and understandable and 36.7% agreed and neutral, 4.1% disagreed, and 12.2% strongly disagreed. About 16% of users strongly agreed that the app was flexible to interact with, 38% agreed, 18% neutral, and 14% disagreed and strongly disagreed. About 30% of users strongly agreed and agreed that they can become skillful in using the app, 24% were neutral, 14% disagreed, and 2% strongly disagreed. About 18% of users strongly agreed that the app is easy to use, 46%

agreed, 12% neutral, 18% disagreed, 6% strongly disagreed. From this analysis it is obvious that the app is easy to use and learn .

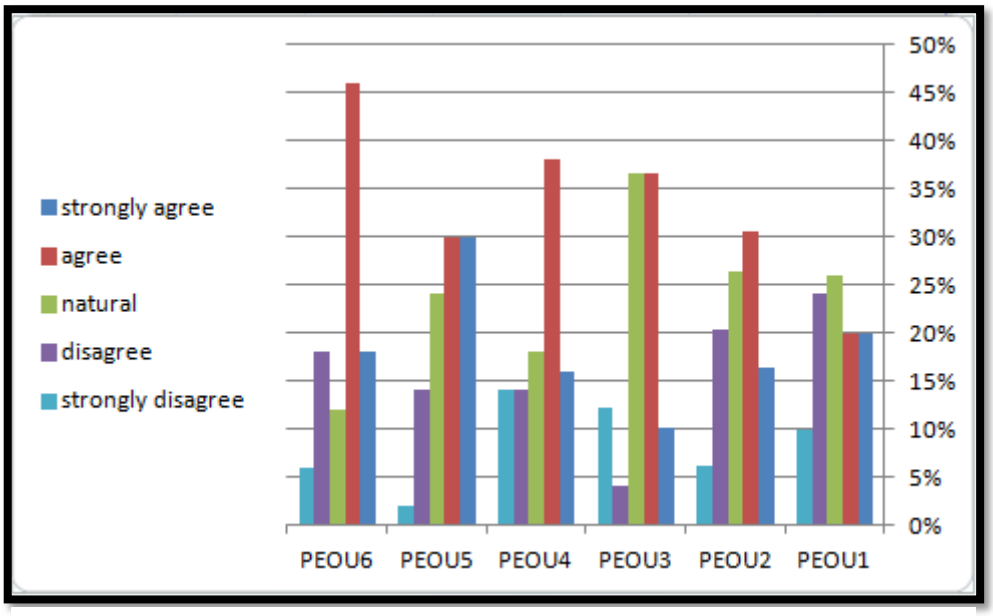



**Figure 6.6 Perceived Ease of Use questions' frequency distribution**

Figure (6.7) shows the third group in the questionnaire which focused on evaluating the Learnability (L) of the app. About 20% of users strongly agreed that it was easy to learn to use the app, 24% agreed, 32% neutral, 18% disagreed, 6% strongly disagreed. About 4% of users strongly agreed that there was too much information to read before they can use the app, 20% agreed, 34% neutral, 20% disagreed, and 22% strongly disagreed. About 24% of users strongly agreed that the information provided by the app was easy to understand, 32% agreed, 18% neutral, 22% disagreed, 4% strongly disagreed. From this analysis we can say that the app is easy to use .

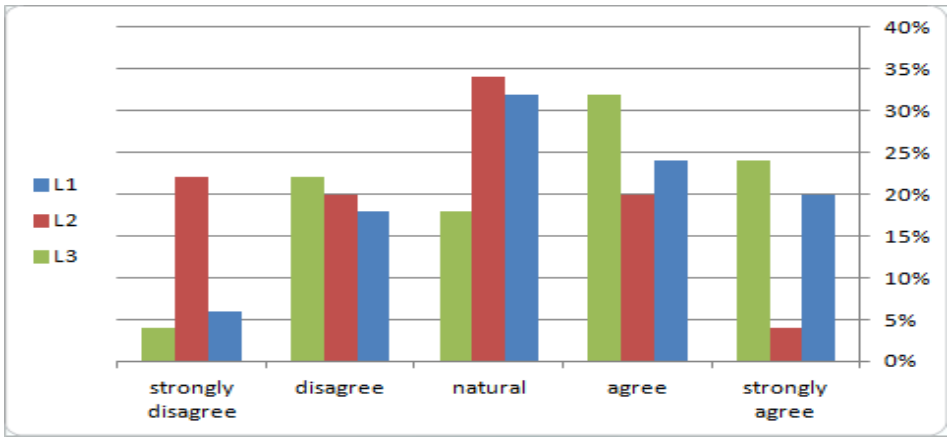



**Figure 6.7 Learnability questions' frequency distribution**

Figure (6.8) shows the forth group in the questionnaire which focused on evaluating the Outcome / Future Use (FU) of the app. About 10% strongly agreed that they were able to complete their tasks quickly using this app, 24% agreed, 36% neutral, 20% disagreed, and 10% strongly disagreed. About 14% of users strongly agreed that they could effectively complete their tasks using this app, 28% agreed, 34% neutral, 18% disagreed, and 6% strongly disagreed. About 18% of users strongly agreed that they were able to complete their tasks efficiently, 36% agreed, 30% neutral, and 16% disagreed. About 2% of users strongly agreed that they could become productive quickly using the app, 48% agreed, 24% neutral, 14% disagreed, and 12% strongly disagreed. About 24% of users strongly agreed that they would use the app regularly, 22% agreed, 26% neutral, 8% disagreed, and 20% strongly disagreed. From this analysis it is obvious that the app will be used by many users .

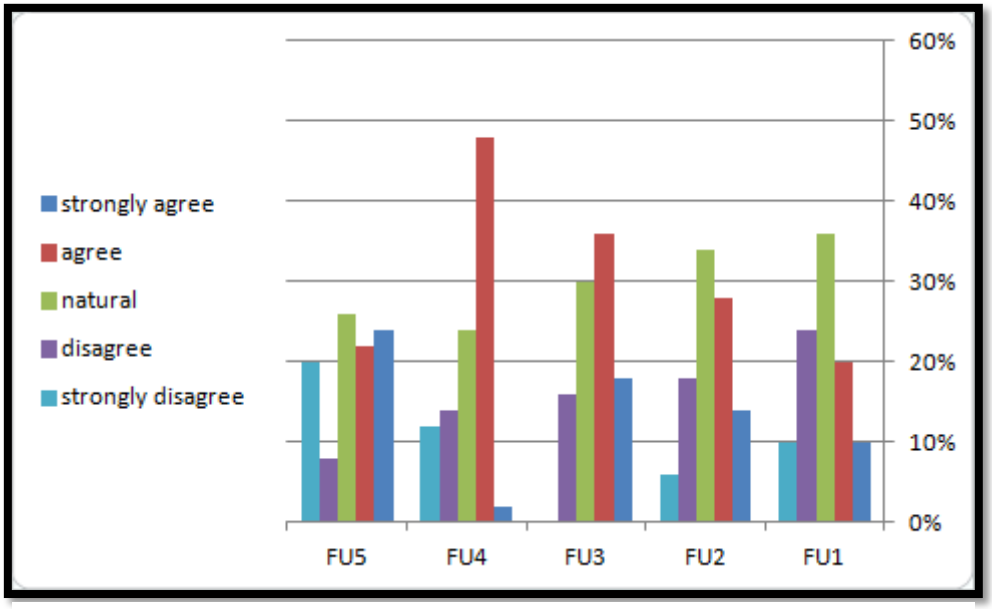



**Figure 6.8 Outcome / Future Use questions' frequency distribution**

## 6.8 Conclusion

In this chapter we have discussed the testing and evaluation phases of this project, the results of testing phase indicated that the app is good to use and meets the user's specifications, and the results from evaluation phase indicated that most users accepted to use this app and they found it easy to use, but it is also need more improvements .

# CHAPTER 7 : CONCLUSION AND FUTURE WORK

## 7.1 Introduction

This chapter is the last part in the project which demonstrates the conclusion of the study and some future ideas that make the app more reliable and can cover all the needs of the user .

## 7.2 Conclusion

Because of the fast development of the world around us we get more and more things to do, until we start to forget some of them or we cannot focus what we want to do, but there are several advantages in our life that we can use them to help us manage our life like smart phones which operate a complete small systems like our app .

The mobile technology helped people in many disciplines because the wireless feature that enabled them doing things outside their home or work, this technology developed new generation of devices that holds operating systems which helped assigning several of tasks of our life to it, so this app has been innovated as one of those features, which helps people in managing and organizing their tasks and it was named Life Scheduler .

Extreme Programming has been used as a software development methodology to develop this app, because we are a small team which consists of three members, and because this type of projects can have rapid requirements changing .

This methodology started with defining the functional and non-functional requirements through looking on other similar apps and deciding what is the most important and interesting for the user, then they were formed and analyzed using different UML techniques such as use-case diagrams, sequence diagrams and activity diagrams, so the developers can use them to develop the app .

The design of the system was then presented and analyzed using class diagram, database schema and ER diagram, also the graphical user interfaces were designed as a prototype for the app to see if the app meets the user specifications, after that the app was implemented .

To see the performance of the app it was tested and evaluated by a sample of end users, according to these two phase we found that this app is useful for users and can be implemented in the market .

## 7.3 Future work

Some future ideas can be added to make the app more reliable and more interesting, they can be summarized in these points :

- Connecting Facebook events with the app to be as task .
- Create a group of friends to do the same task (create events) .
- Using the camera the student can capture an exam record of exams table to be as task, that the app will recognize the name of the exam and the start and end time and the date of it .
- Do not disturb the user during the execution of the task, in case the user wants not to disturb him as making the phone calls silent .
- Connecting the app with the smart home, for example when it is time to sleep (as task has start time) the lights turn off at certain times or any electronic machine .
- Developing a web site for admin and end users, and setting an evaluation for the active users, for each user there will be points to compare it with other user to create challenge atmosphere .

## references


[1] R. v, "data and information" introduction to information technology, vol. 2 edition, Delhi, 2013.

[2] R. v. "data and information" introduction to information technology, vol. 2 edition, Delhi, 2013.

[3] e. Wolfgang, "Introduction" introduction to artificial intelligence, vol. 2 edition, germany, 2011.

[4] c. Liane, "What makes a smart phone smart ?," *lifewire,* 2016.

[5] "technopedia," [Online]. Available: https://www.techopedia.com/definition/2953/mobile-application-mobile-app.

[6] m. li, g. lei and w. jin, "research and development of mobile application for andeoid platform," *international journal of multimedia and ubiquitous engineering,* 4 nov 2014.

[7] d. paul, d. harvey and w. alexander, "Introduction" android 6 for programmers an app-driven approach, vol. 3 edition, Michigan, 2015.

[8] "runrun," [Online]. Available: https://blog.runrun.it/en/task-management-3-old-problems-now-solved/.

[9] "gamelearn," [Online]. Available: https://game-learn.com/problems-task-management-tips-for-managing-to-do-list/.

[10] "googleplay," [Online]. Available: https://play.google.com/store/apps/details?id=com.microsoft.todos&hl=en.

[11] "gooleplay," [Online]. Available: https://play.google.com/store/apps/details?id=com.tasks.todo&hl=en.

[12] "google play," [Online]. Available: file:///C:/Users/pc/Desktop/To%20Do%20List%20-%20Android%20Apps%20on%20Google%20Play.html.

[13] B. v.s, Management Science"Introduction to Management", 2008.

[14] l. Terry, Management Information Systems "Management Information Systems- an overview", 2005.

[15] g. A. gow and r. K. smith, Mobile and Wireless Communications, New York , 2006.

[16] T. Farley, "Mobile telephone history," *Telektronikk,* pp. 22-32, 2005.

[17] "The boom in software designed for mobile phones," *International Telecommunication Union,* july 2009.

[18] H. Dwived, C. Clark and D. Thiel, "top mobile issues and development strategies",application isolation . Mobile Application Security, 2010.

[19] J. McWherter and S. Gowe, "Types of mobile apps" Professional Mobile Application Development.

[20] "TechTarget," [Online]. Available: http://searchsoftwarequality.techtarget.com/definition/integrated-development-environment.

[21] "MapsofIndia," [Online]. Available: http://www.mapsofindia.com/what-is-map.html.

[22] "Interaction Design Foundation," [Online]. Available: https://www.interaction-design.org/literature/article/getting-lost-and-found-maps-and-the-mobile-user-experience.

[23] "Nielsen Norman Group," [Online]. Available: https://www.nngroup.com/articles/mobile-maps-locations/.

[24] "googleplay," google map, [Online]. Available: https://play.google.com/store/apps/details?id=com.google.android.apps.maps&hl=en.

[25] "google books," [Online]. Available: https://books.google.ps/books?id=oPiT-V2eYTsC&printsec=frontcover&dq=nosql&hl=en&sa=X&redir_esc=y#v=onepage&q=nosql&f=false.

[26] "فيوتشر عرب," 11 2014. [Online]. Available: http://www.arabefuture.com/2014/11/chaos-control-gtd-task-list-1-4-5-APK-iOS-Win8-xap.html.

[27] "googleplay/," ASUS Do It Later , [Online]. Available: https://play.google.com/store/apps/details?id=com.asus.task&hl=en.

[28] B. k and F. M, Planning Extreme Programming, vol. first edition, Addison-Wesley, 2000.

[29] B. k, Extreme Programming Explained, vol. second edition, Addison-Wesley, 2004.

[30] j. R, "ronjeffries," 2010. [Online]. Available: http://ronjeffries.com/xprog/book/whatisxp/.

[31] "tutorialspoint," [Online]. Available: https://www.tutorialspoint.com/software_engineering/software_design_strategies.htm.

[32] s. A. s. H. C. J. Y. K. K. O. C. A. and A. S. , Enhancing Security of Linux-based Android Devices, 2008.

[33] C. K. D. J. and S. A. , "An Evolution of Android Operating System and Its Version", International Journal of Engineering and Applied Sciences, 2015.

[34] "tutorialspoint," [Online]. Available: https://www.tutorialspoint.com/dbms/er_diagram_representation.htm.

[35] "ISTQB Exam Certification," [Online]. Available: http://istqbexamcertification.com/what-is-the-software-testing-objectives-and-purpose/.

[36] "ISTQB Exam Certification," [Online]. Available: http://istqbexamcertification.com/what-are-software-testing-levels/.

[37] "ISTQB Exam Certification," [Online]. Available: http://istqbexamcertification.com/why-is-testing-necessary/.

[38] "DB unit," [Online]. Available: http://dbunit.sourceforge.net/index.html.

[39] N. A. Research Methodology In Information Technology, vol. first edition, 2012.

[40] S. I. Software Engineering, 8-edition ed., vol. , UK: Addison-Wesley Publishers Limited 1982, 1984, 2007.

[41] s. k, o. kazyuki and u. k, "content-based video retrieval using video ontology," *IEEE,* pp. 283-389, 12 dec 2007.

[42] s. ja-hwung, h. yu-ting and t. vincent, "Efficient content-based video retrieval by mining temporal patterns," *ACM,* pp. 36-42, - - 2008.

[43] p. B.V, d. A.V and m. B.B, "content-based video retrieval using entropy, edge detection, black and white color features," *IEEE,* pp. 272-276, 18 Apr 2010.

[44] p. s, a. G.S and s. M, "an effective content-based video retrieval utilizing texture, color and optimal key frame features," *IEEE,* pp. 1-5, 5 november 2011.

[45] s. K, u. K and g. M, "examining the applicability of virtual reality technique for video retrieval," *IEEE,* pp. 1-6, 29 june 2012.

[46] C. D. M. P. R. and H. S. , Introduction to Information Retrieval, UK:

Cambridge University Press, 2008, 2008, p. 482 .

[47] a. d. "forums.androidcentral.com," 2014. [Online]. Available: http://forums.androidcentral.com/general-help-how/399060-guide-getting-started-android-updated-2015-lollipop.html. [Accessed 20 8 2016].

[48] . M. Solutions, "www.cheetah3d.com," 2013. [Online]. Available: http://www.cheetah3d.com/?gclid=CImwjpCT28oCFRK3GwodTTIL1w. [Accessed 20 8 2016].

[49] "vuforia developer portal," [Online]. Available: https://developer.vuforia.com.

[50] techopedia, "www.techopedia.com," 2009. [Online]. Available: https://www.techopedia.com/definition/27490/sequence-diagram-uml.

[51] abdxbxbxbxb, 2017.

# Appendices

## Appendix A : Sequence Diagrams

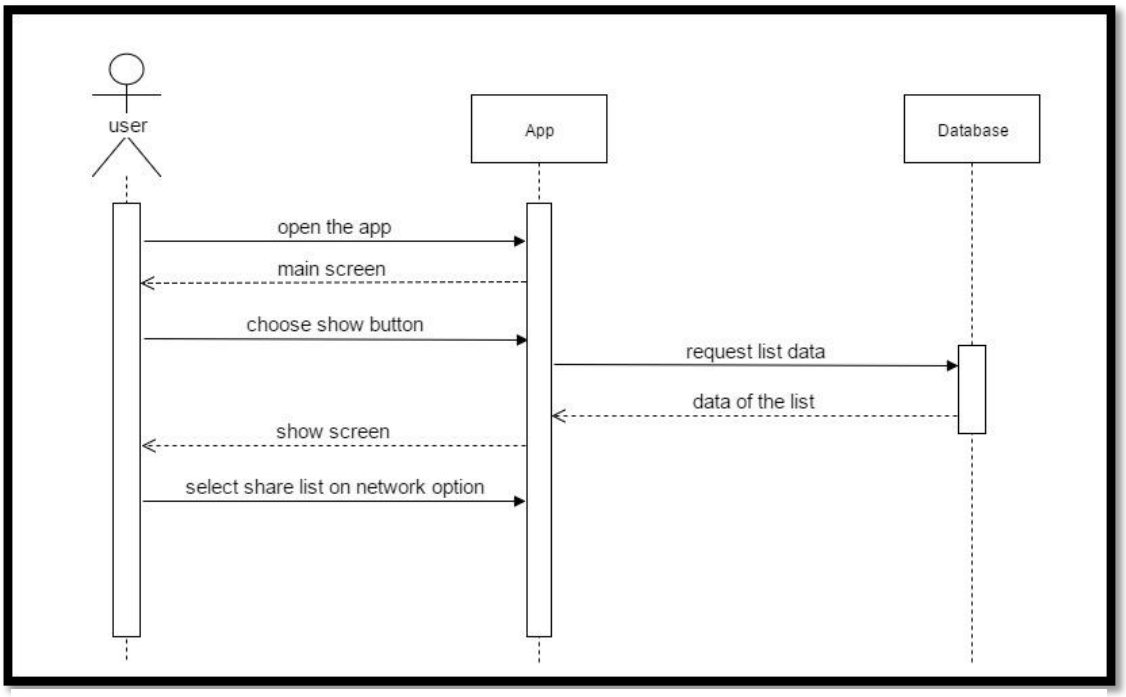



**Figure A.0.1 Show list sequence diagram**

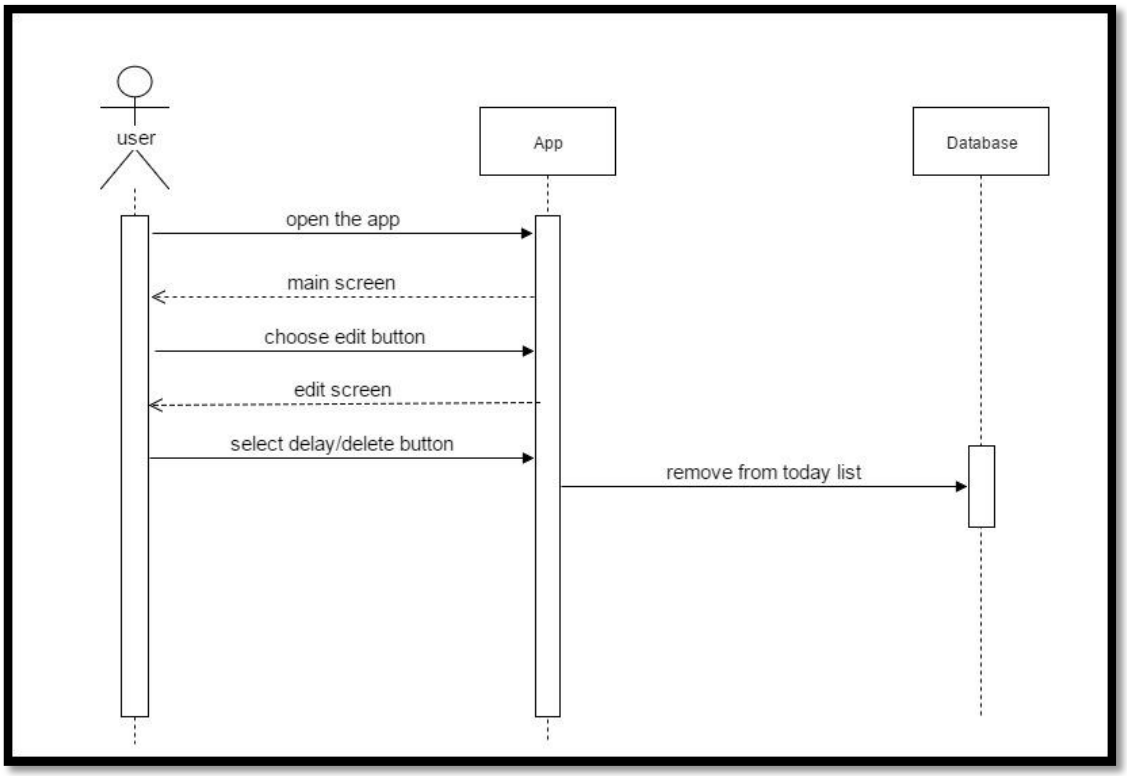



**Figure A.0.2 Edit sequence diagram**

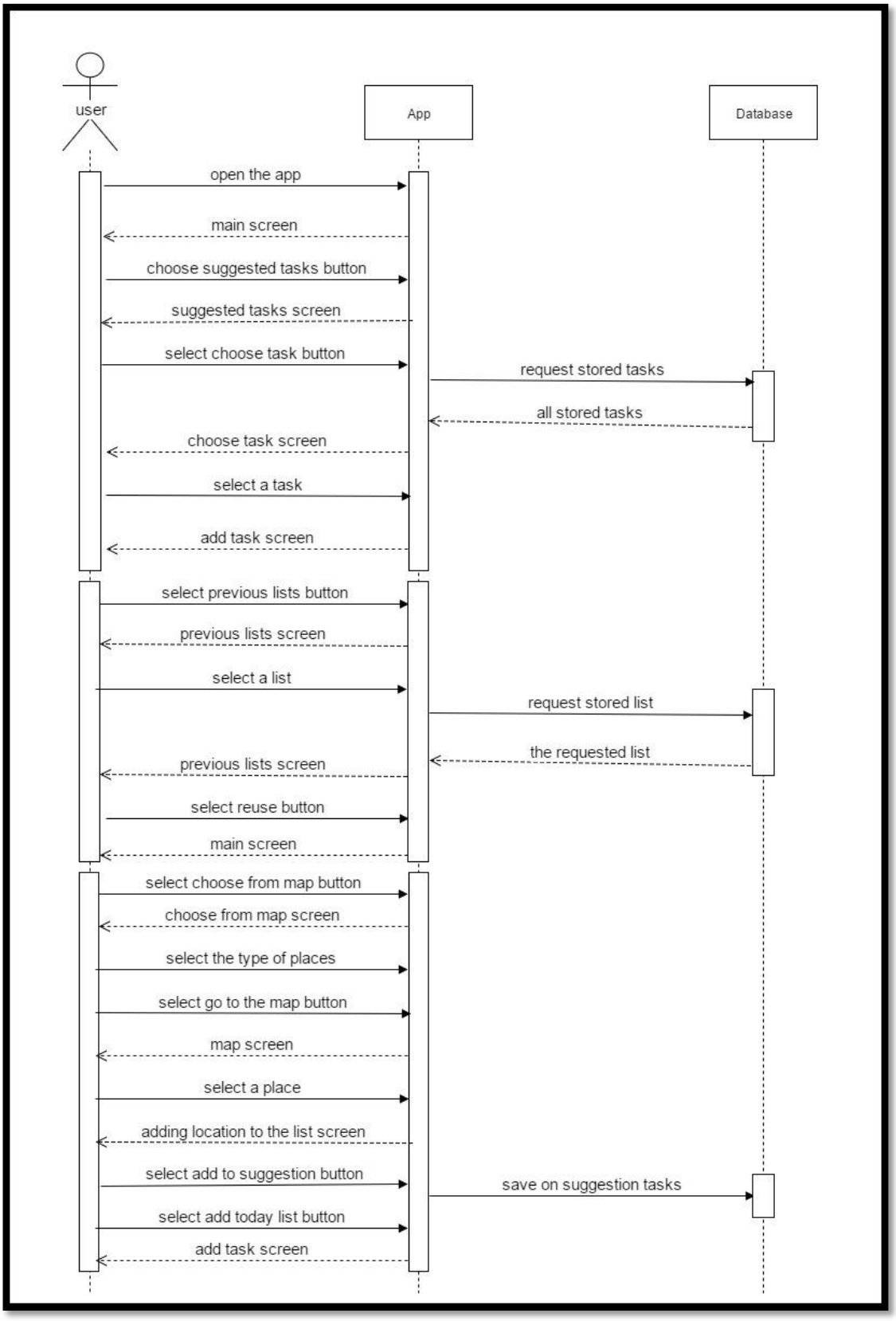



**Figure A.0.3 Suggested tasks sequence diagram**

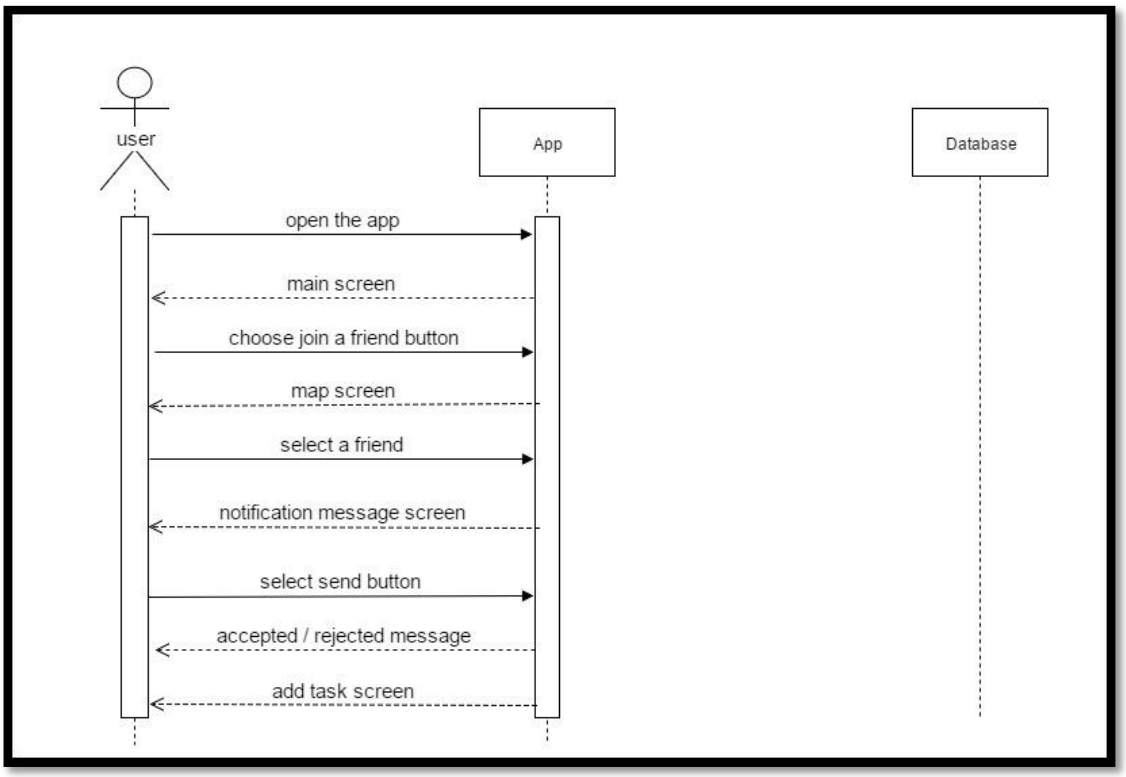



**Figure A.0.4 Join a friend sequence diagram**

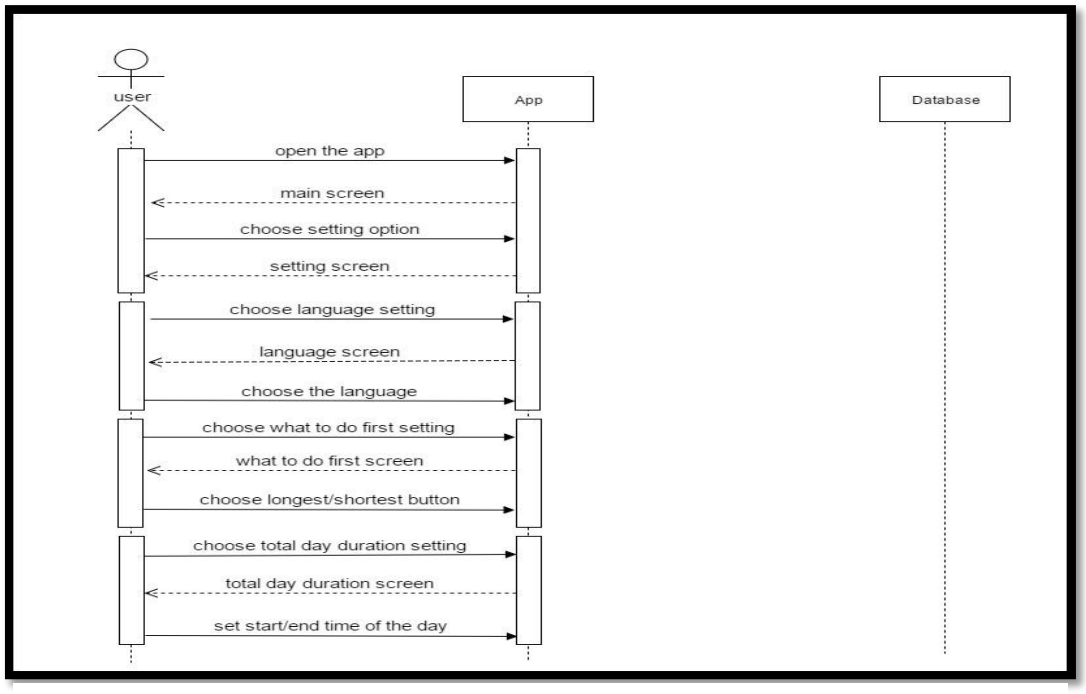



**Figure A.0.5 Settings sequence diagram**

## Appendix B : Activity diagrams

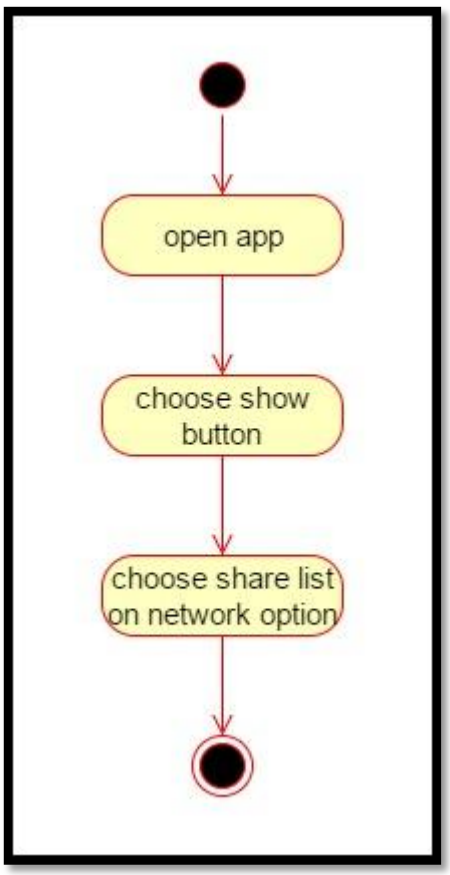



**Figure B.0.1 Show activity diagram**

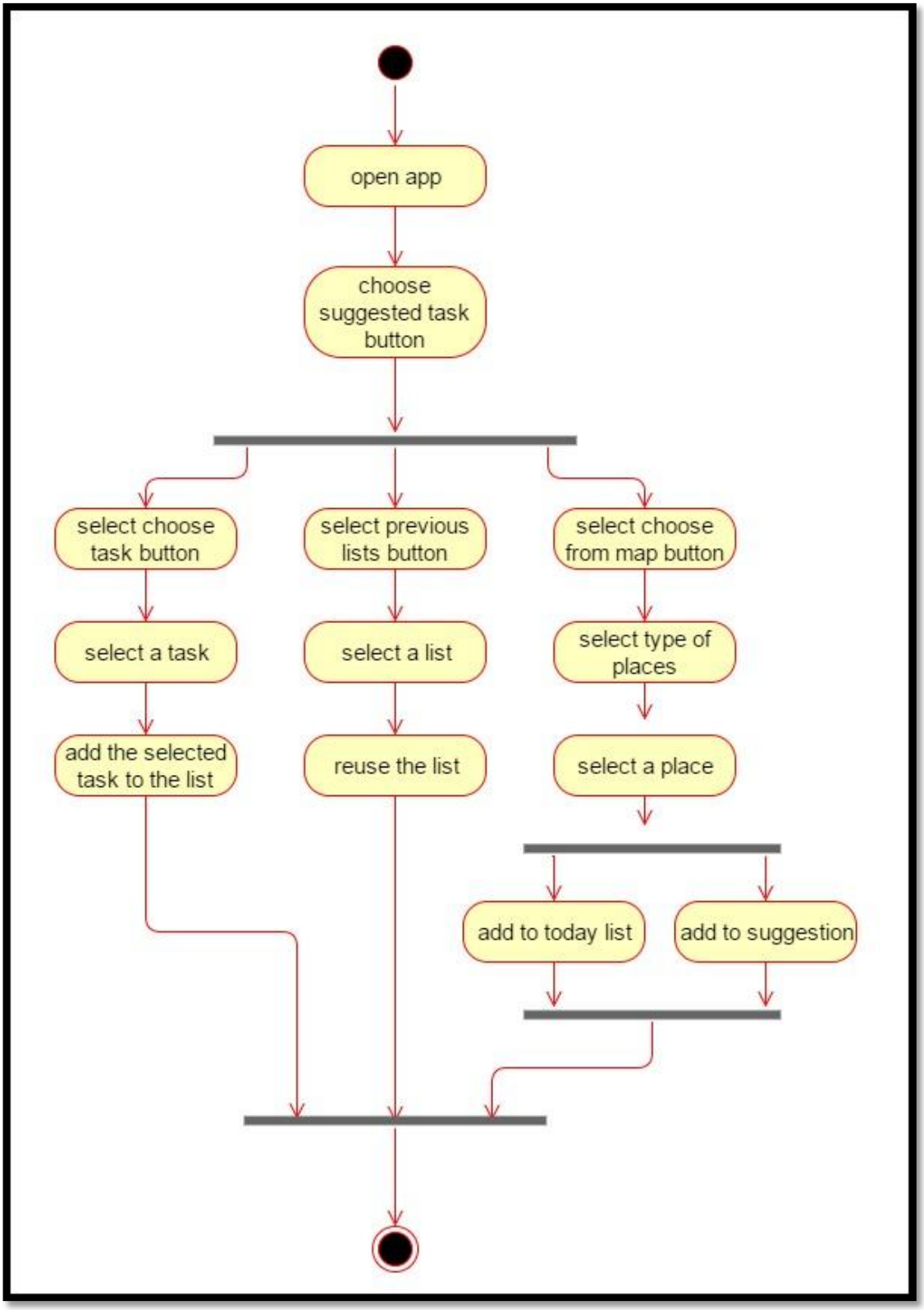



**Figure B.0.2 Suggested tasks activity diagram**

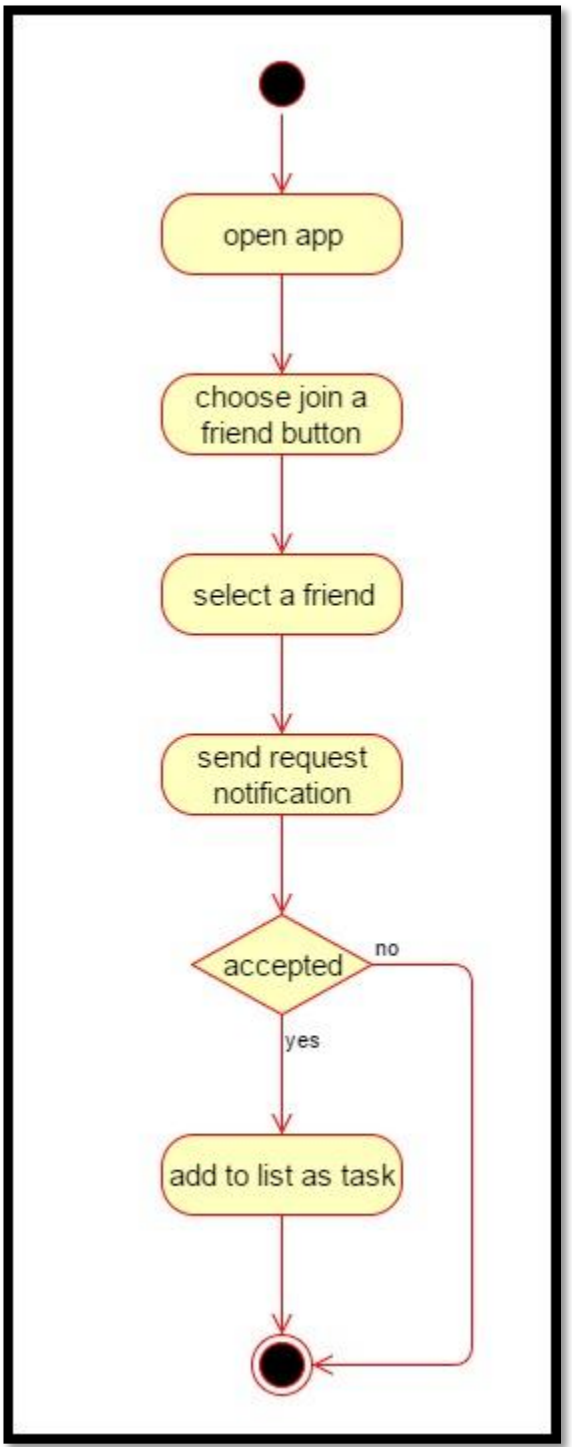



**Figure B.0.3 join a friend activity diagram**

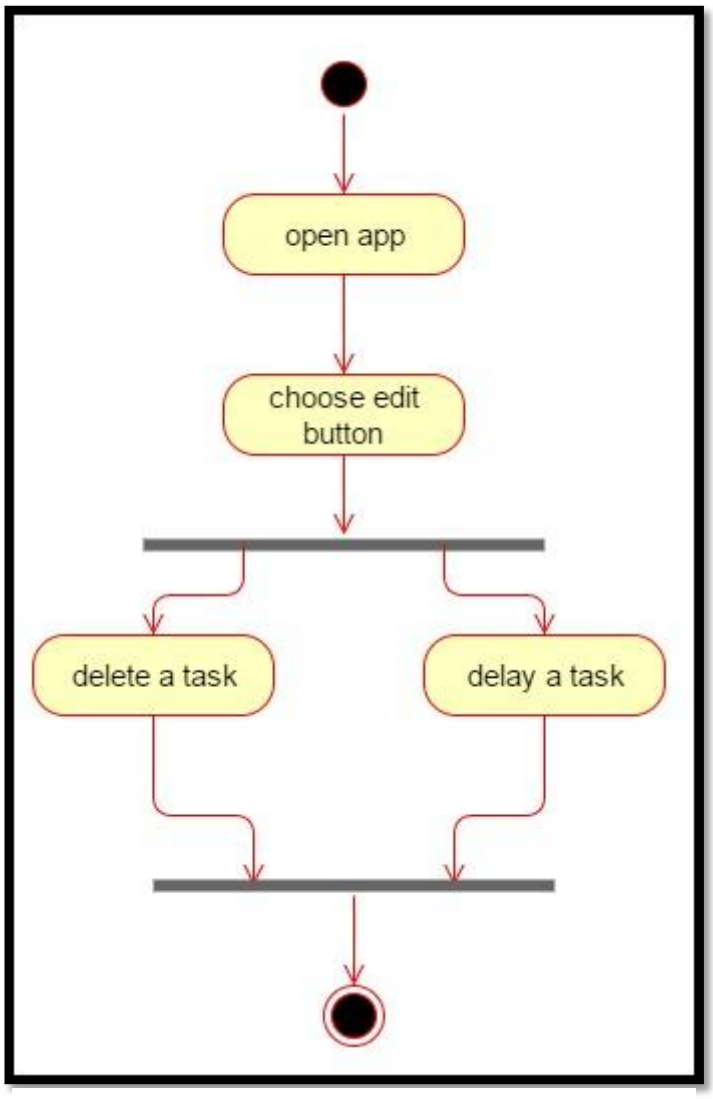



**Figure B.0.4 Edit activity diagram**

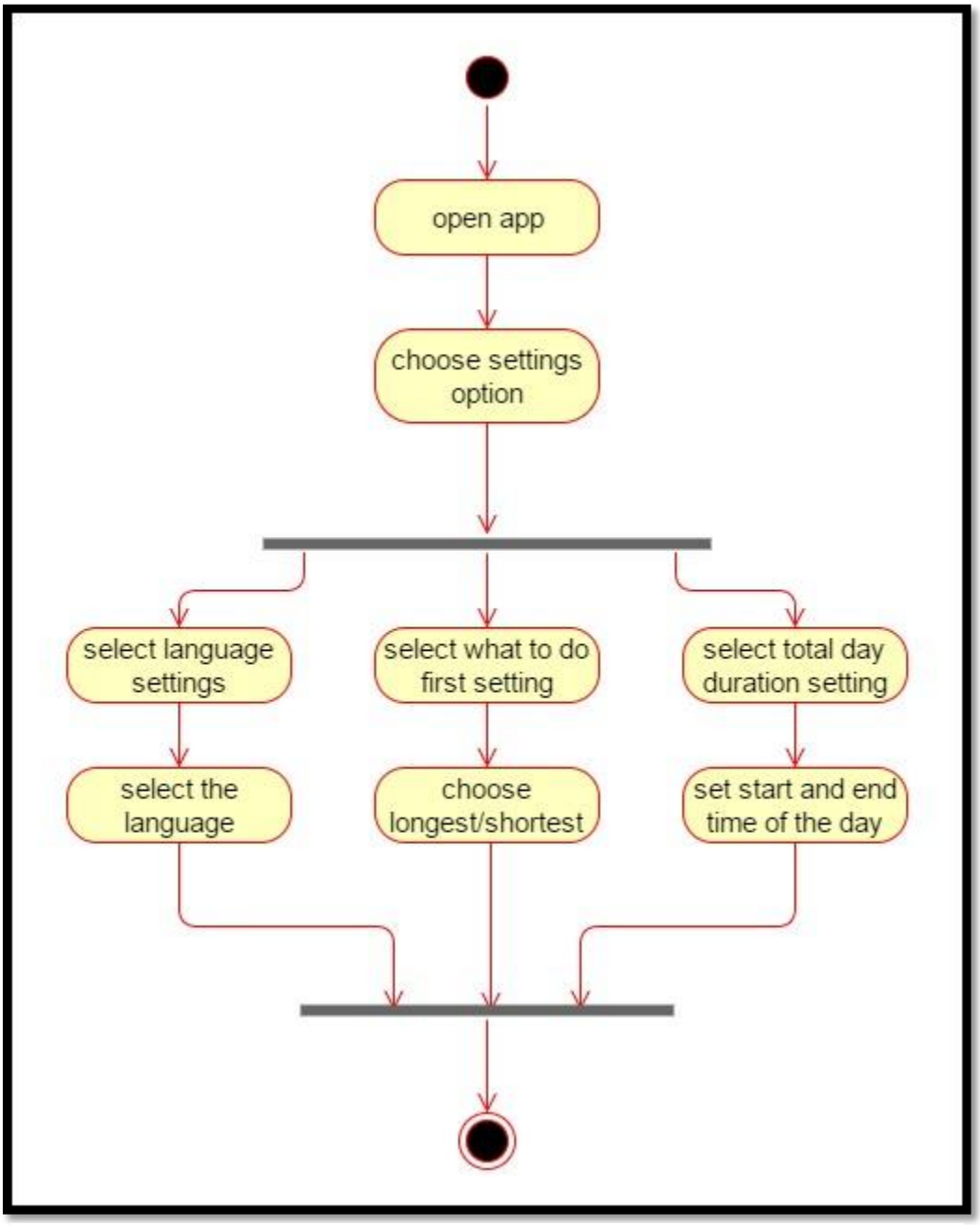



**Figure B.0.5 settings Activity diagram**

## Appendix C : Arrangement algorithm analysis

1. In the database there will be four tables with attributes as follow :
   -Specific (Id, Name, Start time, End time) .
   -Non-specific (Id, Name, duration, priority) .
   -Today-list (Id, Task name, Start time, End time, type) .
   -Old-lists (Id, Date of list, list) .
2. All the times must be converted to the 24 hours system .
3. The specific table must be rearranged in ascending order based on start time, and if there are two or more tasks with the same start time one of them will be removed and an exception message will appear saying cannot add this task .
4. For each task there must be a block on its duration that no other task can be added in this period, and this can be done using these conditions (notice: these conditions will be performed on each row in the specific table) :
   - the start time of the new task must not come in the middle of any task duration
   If (start time of the new task < end time of the previous task && start time of the new task > start time of the previous task)
   - the end time of the new task must not come in the middle of any task duration
   If (end time of the new task < end time of the previous task && end time of the new task > start time of the previous task)
   - the duration of the new task must not come outside the start time of the day
   If (start time of the new task < start time of the day)
   - the duration of the new task must not come outside the end time of the day
   If (start time of the new task > end time of the day || end time of the new task > end time of the day)
5. Now after the arranging the specific table, all the tasks in it must be transferred to the today list table .
6. After all of that the non-specific table must be rearranged based on the priority (from high to low) and the what to do first .
7. There will be a class that has only three values (start, end, duration), which will be used to check if there is a space time the specific tasks to allow another tasks to be added, that an array of objects from this class will be created, that each object will represent a duration of a space time has a start, end, and duration .
8. Then the algorithm will go through the specific table, and for the first row it will subtract the start time of the day from the start time of the task

   Start time of the task – start time of the day =

   If the result is greater than 0 then there is a space time, and if the result is equal to 0 and less then there is no space time .

   Then it will subtract the end time of the previous task from the start time of the next task

   Start time of next – end time of previous =

   If the result is greater than 0 then there is a space time, and if the result is equal to 0 and less then there is no space time .

And for the last task it will subtract the end time of the task from the end time of the day

End time of the day – end time of the task =

If the result is greater than 0 then there is a space time, and if the result is equal to 0 and less then there is no space time .

And each result will be stored in an object in the array, that the result represents the duration value and the first value of the equation represents the start and the second value of the equation represents the end .

9. Then the algorithm will compare between the duration of the non-specific task with the duration of each element in the array to find the appropriate one

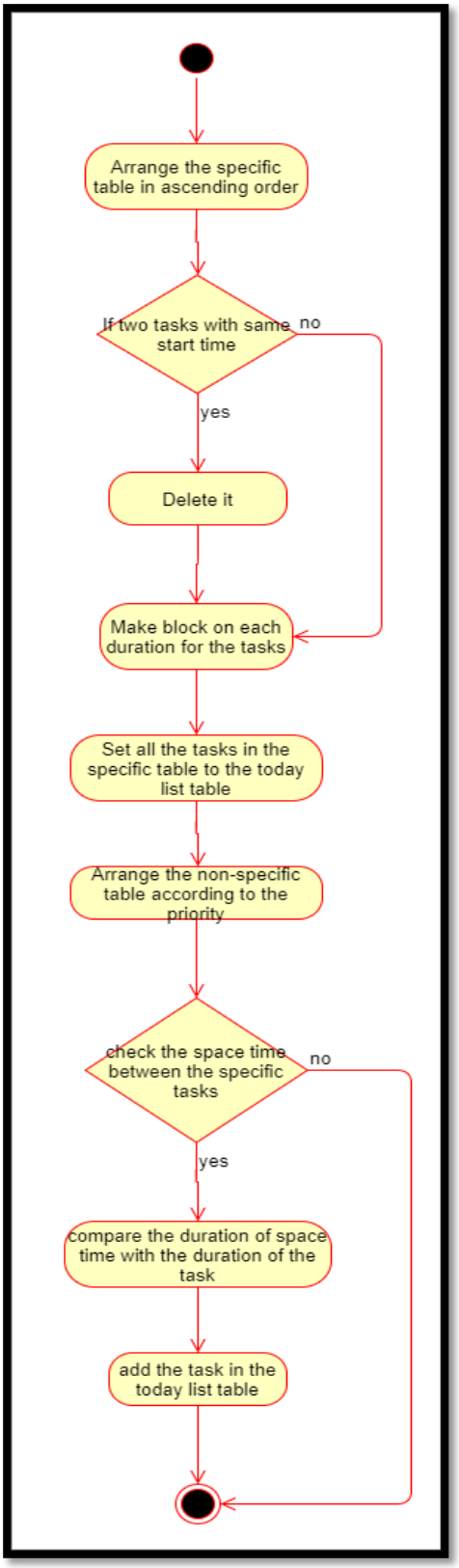
   If (duration of task <= duration in the object)

   In all cases if there is no duration in the array suits the duration of the task then the hours of the day are not sufficient for the task and an exception message will appear says there is no enough time for the task .

# Appendix D : Part of the code of the arrangement algorithm

```
    try {
        projectDB=this.openOrCreateDatabase("project",MODE_PRIVATE,null);
        projectDB.execSQL("CREATE TABLE IF NOT EXISTS specific (taskname VARCHAR(30),start-time INT(2),end-time INT(2)");
        projectDB.execSQL("CREATE TABLE IF NOT EXISTS non-specific (taskname VARCHAR(30),duration INT(2),priority VARCHAR(20)");
        projectDB.execSQL("CREATE TABLE IF NOT EXISTS today-list (id INT(3),taskname VARCHAR(30),start-time INT(2),end-time INT(2),type \
    }catch (Exception e){
        e.printStackTrace();
    }

}

public void DBoperations (){
    String taskName=task.getText().toString();
    int starttime=Integer.parseInt(start.getText().toString());
    int endtime=Integer.parseInt(end.getText().toString());
    int dura=Integer.parseInt(duration.getText().toString());
    String pri=spinner.getSelectedItem().toString();

    try{

        if (v.getId()==R.id.Specific) {
            projectDB.execSQL("INSERT INTO specific (taskname, start-time, end-time) values ("+taskName+","+starttime+","+endtime+")");
        }
        if (v.getId()==R.id.radioButton) {
            projectDB.execSQL("INSERT INTO non-specific (taskname, duration, priority) values ("+taskName+","+dura+","+pri+")");
        }
        projectDB.execSQL("ALTER TABLE specific ORDER BY start-time ase");
        Cursor c=projectDB.rawQuery("SELECT * FROM specific", null);
        int nameindex=c.getColumnIndex("taskname");
        int startindex=c.getColumnIndex("start-time");
        int endindex=c.getColumnIndex("end-time");
        c.moveToFirst();
        while (c!=null){
            int d=c.getInt(startindex);
            c.moveToNext();
            int f=c.getInt(startindex);
            if (d==f) {
                projectDB.execSQL("DELETE FROM specific WHERE taskname=" + c.getString(startindex));
                c.moveToFirst();// need to remove
            }
        }
        c=projectDB.rawQuery("SELECT * FROM specific", null);
        c.moveToFirst();
        while (c!=null){
            int g=c.getInt(endindex);
            int h=c.getInt(startindex);
            if (starttime<g&&c.getString(nameindex)!=taskName&&starttime>h){
                projectDB.execSQL("DELETE FROM specific WHERE taskname="+taskName);
                Toast.makeText(getApplicationContext(),"this task cannot be added",Toast.LENGTH_SHORT).show();
            }
            c.moveToNext();
        }

        c.moveToFirst();
        while (c!=null){
            int g=c.getInt(endindex);
            int h=c.getInt(startindex);
            if (endtime>h&&c.getString(nameindex)!=taskName&&endtime<g){
                projectDB.execSQL("DELETE FROM specific WHERE taskname="+taskName);
                Toast.makeText(getApplicationContext(),"this task cannot be added",Toast.LENGTH_SHORT).show();
            }
            c.moveToNext();
        }
        settingDay s=new settingDay();
        if (starttime<s.daystarttime){
            projectDB.execSQL("DELETE FROM specific WHERE taskname="+taskName);
            Toast.makeText(getApplicationContext(),"this task cannot be added",Toast.LENGTH_SHORT).show();
        }

        if (starttime>s.dayendtime||endtime>s.dayendtime){
            projectDB.execSQL("DELETE FROM specific WHERE taskname="+taskName);
            Toast.makeText(getApplicationContext(),"this task cannot be added",Toast.LENGTH_SHORT).show();
        }

    }catch (Exception e){
        e.printStackTrace();
    }
}
```

## Appendix E : Flow chart for the algorithm



**Appendix F : Evaluation questionnaire for end users about Life Scheduler in Arabic**

# استبيان

**عزيزي المستخدم :**

هذا الاستبيان يهدف إلى فهم مدى رضاك وقبولك Life scheduler كتطبيق يساعد في تنظيم حياتك اليومية ، مشاركتك في الإجابة عن هذا الاستبيان مهمة جدا ومقدرة، للتأكد من نجاح هذه الدراسة المعلومة المقدمة منك سيتم التعامل معها بأقصى درجات السرية .

شكرا جزيلا لتعاونك معنا ،،،

| أ. معلومات عامة | | | | |
|---|---|---|---|---|
| هذا الجزء عبارة عن معلومات أساسية عنك ، الرجاء تعبة الفراغات من خلال وضع اشارة ( √ ) في الخيار الصحيح (الرجاء تعبة فراغ واحد فقط ) | | | | |
| 1- ما هو جنسك | ☐ ذكر | ☐ أنثى | | |
| 2- ما هو عمرك | ☐ أقل من 20 | ☐ 20- 25 | ☐ 26 - 30 | ☐ أكثر من 30 |
| 3- تخصصك<br>☐ علوم ( ICT – IT – رياضيات – هندسة )<br>☐ تجارة ( ادارة اعمال – محاسبة )<br>☐ اداب ( لغات – تاريخ – جغرافيا – قانون )<br>☐ أخرى ( الرجاء تحديد تخصصك ..... ) | | | | |
| 4- عدد سنوات استخدامك للهواتف الذكية<br>☐ اقل من 5 سنوات ☐ 5-10 سنوات ☐ اكثر من 10 سنوات | | | | |

| ب . العوامل التي تجعلك راضيا عن التطبيق |
|---|

الرجاء تحديد درجة موافقتك على الجمل التالية ، بواسطة وضع دائرة على الخيار المناسب حسب المقياس التالي :

1= لا اوافق بشدة 2= لا اوافق 3= محايد 4= اوافق 5= اوافق بشدة

| | | | | | | |
|---|---|---|---|---|---|---|
| 1. هذا القسم يهدف إلى فهم درجة قابلية المستخدم في استخدام التطبيق | | | | | | |
| -1 | استخدام التطبيق يمكنني من انجاز المهمات والواجبات بسرعة | 1 | 2 | 3 | 4 | 5 |
| -2 | استخدام التطبيق يحسن من ادائي في حياتي اليومية | 1 | 2 | 3 | 4 | 5 |
| -3 | استخدام التطبيق يزود من انتاجيتي في حياتي اليومية | 1 | 2 | 3 | 4 | 5 |
| -4 | استخدام التطبيق يعزز من كفاءتي في عملي | 1 | 2 | 3 | 4 | 5 |
| -5 | استخدام التطبيق يمكنني من الاندماج في الحياة الاجتماعية | 1 | 2 | 3 | 4 | 5 |
| -6 | أنا اجد التطبيق مفيد في حياتي | 1 | 2 | 3 | 4 | 5 |
| 2. هذا القسم يهدف إلى فهم درجة خلو التطبيق من الجهود في الاستخدام | | | | | | |
| -1 | تعلم استخدام التطبيق سهل علي | 1 | 2 | 3 | 4 | 5 |
| -2 | اجد من السهل استخدام التطبيق لأجد ما اريد | 1 | 2 | 3 | 4 | 5 |
| -3 | تفاعلي مع التطبيق واضح وقابل للفهم | 1 | 2 | 3 | 4 | 5 |
| -4 | أنا أجد التطبيق مرن في التفاعل معه | 1 | 2 | 3 | 4 | 5 |

| | | | | | | |
|---|---|---|---|---|---|---|
| -5 | من السهل علي أن أصبح ماهر في استخدام التطبيق | 1 | 2 | 3 | 4 | 5 |
| -6 | عموما ، انا اجد التطبيق سهل الاستخدام | 1 | 2 | 3 | 4 | 5 |

| 3. هذا القسم يهدف إلى فهم درجة قدرة التطبيق على تعليمك كيفية استخدامه | | | | | | |
|---|---|---|---|---|---|---|
| -1 | كان من السهل علي تعلم استخدام التطبيق | 1 | 2 | 3 | 4 | 5 |
| -2 | كان هناك معلومات كثيرة لقراءتها قبل ان استطيع استخدام التطبيق | 1 | 2 | 3 | 4 | 5 |
| -3 | المعلومة التي يزودها التطبيق كانت سهلة للفهم | 1 | 2 | 3 | 4 | 5 |

| 4. هذا القسم يهدف الى فهم النتائج – الاستخدام المستقبلي للتطبيق | | | | | | |
|---|---|---|---|---|---|---|
| -1 | أنا كنت قادر على ان اكمل مهامي بسرعة باستخدام التطبيق | 1 | 2 | 3 | 4 | 5 |
| -2 | أنا كنت قادر على ان اكمل مهامي بكفاءة باستخدام التطبيق | 1 | 2 | 3 | 4 | 5 |
| -3 | أنا اعتقد انني استطيع أن اصبح سريع الاستفادة باستخدام التطبيق | 1 | 2 | 3 | 4 | 5 |
| -4 | من خلال خبرتي الحالية من استخدام التطبيق ، انا اعتقد انني استطيع ان استخدمه على نحو نظامي | 1 | 2 | 3 | 4 | 5 |
| -5 | انا اوصي الاخرين باستخدام التطبيق | 1 | 2 | 3 | 4 | 5 |

باختصار ، هل تملك أي تعليقات اخرى عن التطبيق ؟

.....
.....
.....

شكرا على جهودك وتعاونك ..

## Appendix G : Evaluation questionnaire for end users about Life Scheduler in English

# Questionnaire

**Dear user,**

This questionnaire aims to understand your satisfaction and acceptance about Life Scheduler as an app which helps people manage their personal life .

Your participation in answering this questionnaire is very much significant and appreciated to ensure the success of this study. The information provided by you will be handled with utmost confidentiality.

**Thank you very much for your cooperation.**

| **A. GENERAL INFORMATION** | | | | |
|---|---|---|---|---|
| This section is about your background information. *Please fill up the blanks and mark [√] the most appropriate (please tick **one only**).* | | | | |
| 1-What is your gender | ☐ Male | ☐ Female | | |
| 2-What is your age | ☐ Under 20 | ☐ 20-25 | ☐ 26-30 | ☐ More than 30 |
| 3-Your Education background (*Please Tick one only*):<br>☐ Science (*e.g., IT, ICT, Math, Engineering*)<br>☐ Business (*e.g., Accounting, Finance, Management*)<br>☐ Art Studies (*e.g., Languages, Law, History*)<br>☐ Other (*Please identify*) _____ | | | | |
| 4- your experience in using Mobile apps :<br>☐ Less than 5 years<br>☐ 5-10 years<br>☐ More than 10 years | | | | |

| B. ACCEPTANCE FACTORS OF MOODLE | | | | | | |
|---|---|---|---|---|---|---|
| Please indicate your degree of agreement on the following statement, by circling or tick the most appropriate choice using the scale below:<br><br>**1= *Strongly Disagree*** **2= *Disagree*** **3= *Neutral*** **4= *Agree*** **5= *Strongly Agree*** | | | | | | |
| **1-This section aims to understand the *Perceived Usefulness (PU)* of the app.** | | | | | | |
| 1- | Using the app would enable me to accomplish tasks more quickly. | 1 | 2 | 3 | 4 | 5 |
| 2- | Using the app would improve my performance in my daily life . | 1 | 2 | 3 | 4 | 5 |
| 3- | Using the app would increase my productivity in my life. | 1 | 2 | 3 | 4 | 5 |
| 4- | Using the app would enhance my effectiveness in my work. | 1 | 2 | 3 | 4 | 5 |
| 5- | Using the app would make it easier to engage me in the social life. | 1 | 2 | 3 | 4 | 5 |
| 6- | I find the app useful in my life. | 1 | 2 | 3 | 4 | 5 |
| **2-This section aims to understand the *Perceived Ease of Use (PEOU)* of the app** | | | | | | |
| 1- | Learning to use the app is easy for me. | 1 | 2 | 3 | 4 | 5 |
| 2- | I find it easy to use the app to find what I want. | 1 | 2 | 3 | 4 | 5 |
| 3- | My interaction with the app is clear and understandable. | 1 | 2 | 3 | 4 | 5 |
| 4- | I find the app are flexible to interact with. | 1 | 2 | 3 | 4 | 5 |
| 5- | It is easy for me to become skillful in using the app. | 1 | 2 | 3 | 4 | 5 |
| 6- | Overall, I find the app easy to use. | 1 | 2 | 3 | 4 | 5 |

| | | | | | | |
|---|---|---|---|---|---|---|
| **3-This section aims to understand the *Learnability (L)* of the app** | | | | | | |
| 1- | It was easy to learn to use the app. | 1 | 2 | 3 | 4 | 5 |
| 2- | There was too much information to read before I can use the app. | 1 | 2 | 3 | 4 | 5 |
| 3- | The information provided by the app was easy to understand. | 1 | 2 | 3 | 4 | 5 |
| **4-This section aims to understand the *Outcome / Future Use (FU)* of the app** | | | | | | |
| 1- | I was able to complete my tasks quickly using the app. | 1 | 2 | 3 | 4 | 5 |
| 2- | I could effectively complete my tasks using the app. | 1 | 2 | 3 | 4 | 5 |
| 3- | I was able to efficiently complete my tasks using the app. | 1 | 2 | 3 | 4 | 5 |
| 4- | I believe I could become productive quickly using the app. | 1 | 2 | 3 | 4 | 5 |
| 5- | From my current experience with using the app, I think I would use it regularly. | 1 | 2 | 3 | 4 | 5 |

Briefly, do you have any other comments about the app? ……
……

***Thanks for Your Cooperation and Efforts***

## Appendix H : use case scenarios

**Table 0-1 Add task use case scenario**

| **Use Case Name :** Add new task | **ID :** UC-2 | **Priority :** High |
|---|---|---|
| **Actor :** App User | | |
| **Description :** the user inserts his task and define the type of the task (specific time/non-specific time), if it was specific time he shall insert the start and end time for it, and if it was non-specific time he shall insert the duration for it and specify the degree of priority of it, then the app will create a list containing this task, and it will prioritize it according to an arranging algorithm . | | |
| **Preconditions :**<br>1- the user must specify the day start and end time values .<br>2- the user must specify the priority of longest or shortest task first . | | |
| **Normal Course :**<br>1- the user inserts the name of the task .<br>2- the user specify the type of the task .<br>3- the user inserts the start and time of the task or the duration and degree of priority . | | |
| **Alternative Course :**<br>1- the user reuse previous list . | | |
| **Post-conditions :**<br>1- the task must be stored in the app .<br>2- the app must create a list containing the added task or add the task to an existing task . | | |

**Table 0-2 Show current list use case task**

| **Use Case Name :** Show current list | **ID :** UC-3 | **Priority :** Low |
|---|---|---|
| **Actor :** App User | | |
| **Description :** the user can display the list of the current day only, and he can choose to share the tasks of it on the map . | | |
| **Preconditions :**<br>1- a list must be created and stored in the app . | | |
| **Normal Course :**<br>1- the user displays the today list .<br>2- the user choose to share his tasks on the map . | | |
| **Alternative Course :** | | |
| **Post-conditions :**<br>1- the app must show the task that the user is doing on the map if he choose this . | | |

**Table 0-3 Choose suggested tasks use case scenario**

| **Use Case Name :** Choose suggested tasks | **ID :** UC-4 | **Priority :** Medium |
|---|---|---|
| **Actor :** App User | | |
| **Description :** the user can selects tasks from a group of pre-inserted tasks, and he can reuse previous list he created, and he can selects a place on the map to visit . | | |
| **Preconditions :**<br>1- all lists of the user must be stored in the app .<br>2- the WiFi must be enabled so the places can be displayed on the map . | | |
| **Normal Course :**<br>1- the user selects a task from a group of pre-inserted tasks .<br>2- the user reuses any old list .<br>3- the user picks up a place from the app and add it as task . | | |
| **Alternative Course :**<br>1- the user add his tasks normally from his own mind . | | |
| **Post-conditions :**<br>1- the user must specify the values of the task when he selects a task from the group of pre-inserted tasks or when he selects a place from the map . | | |

**Table 0-4 Join a friend from the map use case scenario**

| **Use Case Name :** Join a friend from the map | **ID :** UC-5 | **Priority :** Medium |
|---|---|---|
| **Actor :** App User | | |
| **Description :** from the map that illustrates the other friends users who use the same app and the task that each one is doing, the user can select a friend and send him a notification that he wants to join the task with him . | | |
| **Preconditions :**<br>1- the WiFi must be enabled so friends can be displayed on the map . | | |
| **Normal Course :**<br>1- the user picks up a friend from the app and add it as task . | | |
| **Alternative Course :**<br>1- the user chooses not to share any friend and stay with his tasks . | | |
| **Post-conditions :**<br>1- the user must specify the values of the task when he selects a friend . | | |

**Table 0-5 Edit today list use case scenario**

| **Use Case Name :** Edit today list | **ID :** UC-6 | **Priority :** High |
|---|---|---|
| **Actor :** App User | | |
| **Description :** the user can deletes or delay a task from the today list . | | |
| **Preconditions :**<br>1- a list must be created and stored in the app . | | |
| **Normal Course :**<br>1- the user selects a task to delete .<br>2- the user selects a task to delay . | | |
| **Alternative Course :**<br>1- the user keeps the task as they are . | | |
| **Post-conditions :**<br>1- the app must rearrange the list and save the changes . | | |

# Appendix I : Test cases

**Table 0-1 Show today list screen test case**

| **TC2-Show today list screen-check successful of displaying the list .** |
|---|
| **Description :** the app should successfully display the today list . |
| **Preconditions :** the user must have added his tasks so a list is created and stored in the app . |
| **Assumptions :** a supported device is being used . |
| **Test steps :**<br>1- open the app .<br>2- choose Show today list button .<br>3- display the list . |
| **Expected result :** the screen of the today list should be displayed containing the list with the added tasks showing its number, name, start time, and end time . |

**Table 0-2 Suggested tasks screen test case**

| **TC3-Suggested tasks screen-check successful of adding task from this screen .** | | |
|---|---|---|
| **Description :** the app should successfully add the user choice as a task in the list . | | |
| **Preconditions :** the WiFi must be enabled . | | |
| **Assumptions :** a supported device is being used . | | |
| **Test steps :** | | |
| 1- open the app .<br>2- choose Suggested tasks button .<br>3- select Choose task button .<br>4- choose one of the tasks .<br>5- add values to the task .<br>6- click add task button . | 1- open the app .<br>2- choose Suggested tasks button .<br>3- select previous lists button<br>4- select a list .<br>5- click on reuse button . | 1- open the app .<br>2- choose Suggested tasks button .<br>3- select Go to map button .<br>4- select a place on the map .<br>5- add values to the task .<br>6- click add task button . |
| **Expected result :** the screen of the today list should be displayed containing the list with the added task showing its number, name, start time, and end time . | | |

**Table 0-3 Join a friend screen test case**

| **TC4-join a friend screen-check successful of displaying the map .** |
|---|
| **Description :** the app should successfully display the map showing on it the friends users who use the same app and the task that each one is doing, and when the user selects a friend and sends him a notification and the acceptance is received the app should add this friend as a task . |
| **Preconditions :**<br>1- the WiFi must be enabled .<br>2- the phone number of the friend must be stored in the SIM . |
| **Assumptions :** a supported device is being used . |
| **Test steps :**<br>1- open the app .<br>2- choose Join a friend button .<br>3- select a user .<br>4- send a notification message .<br>5- if the other user accepted the request, then add values for the task .<br>6- click add task button . |
| **Expected result :** the screen of the today list should be displayed containing the list with the added task showing its number, name, start time, and end time . |

**Table 0-4 Edit today list screen test case**

| **TC5-Edit today list screen-check successful of displaying the list .** |
|---|
| **Description :** the app should successfully delete or delay a task from the today list . |
| **Preconditions :** the user must have added his tasks so a list is created and stored in the app . |
| **Assumptions :** a supported device is being used . |
| **Test steps :**<br>1- open the app .<br>2- choose Edit button .<br>3- display the list .<br>4- select a task .<br>5-click delete/delay button . |
| **Expected result :** the screen of Edit list should be displayed containing the list without the task that has been deleted or disabled if it has been delayed . |