

Motazen

Prepared by

Reema AlSaif, 441201280
Manar AlMalki, 441201327
Reem AlRossais, 441203397
Randa Bakhshwain, 441203867

Supervised by
Dr. Alhanoof AlThnian
Dr. Khulood AlYahya



Table of Contents

1	Introduction	8
1.1	The Problem	10
1.2	The Solution	11
1.3	Product	12
1.3.1	Product Vision	12
1.3.2	Product Roadmap	13
1.3.3	Objectives	14
1.3.4	Scope	16
1.3.5	Hardware/Software Tools and Costs	17
1.4	Scrum Team	18
1.4.1	Skill Set Requirements	18
1.4.2	Learning	18
1.4.3	Roles and Responsibilities	19
2	Background	21
2.1	Life Coaching Tools	21
2.2	The Wheel of Life tool	22
2.2.1	The Concept of Journaling	24
2.2.2	Gamification	25
2.2.3	Ranking algorithms in multi- criteria decision solving techniques	26
3	Literature Review	29
3.1	Competitive Product Analysis	29
3.1.1	LifeWheel—Goal Setting Tracker	29
3.1.2	Happy: Wheel of Life	31
3.1.3	Fabulous	32
3.1.4	Evaluation Wheel	33
3.1.5	Ambitionist	34
3.1.6	Goal Setting Tracker Planner	35
4	System Requirements	41
4.1	System Users	41
4.2	Requirements Elicitation and Analysis	41
4.2.1	Summary of The Questionnaire	46
4.3	User Interactions	47
4.4	Product Backlog	48
5	System Design	57



5.1	Architectural Diagram	57
5.2	Class Diagram	59
5.3	Component Level Design	60
5.3.1	Generate Wheel of Life Flowchart	60
5.3.2	Add Goals	62
5.3.3	Ranking Tasks	63
5.4	Data Design	64
5.4.1	Data Models	64
5.5	Interface Design	67
5.5.1	UX Guidelines used	67
5.5.2	Site Map	69
6	System Implementation	71
6.1	Initialization	71
6.2	Connecting to the database	71
6.3	Core Functions	72
6.3.1	Ranked Tasks	72
6.3.2	Community	92
6.4	Integration	95
6.5	Challenges	95
7	System Testing	100
7.1	User Acceptance Testing	100
7.1.1	Demographics of Participants	101
7.1.2	Questionnaire/Interview Results	101
7.2	Quality Attributes (Non-functional testing)	112
7.3	Discussion	113
8	Conclusions and Future Work	115
8.1	Local Impact	115
8.2	Global Impact	115
8.3	Problems and challenges	116
8.4	Limitations of the system	116
8.5	Main contribution of the project	116
8.6	Future Work	117
9	References	119
10	Appendix A	124
10.1	Questionnaire's Questions	124
10.2	Questionnaire's responses	125



11 Appendix B	129
11.1 Usability Testing Questions	129
12 Appendix C	130
12.1 Assessment Questions	130
13 Appendix D	132
13.1 Ranking Algorithm	132
Normalize the value by dividing over the maximum T over all tasks	132
$NTi = TimaxT$	132
Where maxT is the maximum Ti over all tasks	132
• Dependency is calculated as follows:	132
$Dependency_i = \frac{\text{num of tasks dependent on task } i}{\text{total num of tasks}}$	132
13.2 Ranking Test Cases	133

Table of Tables

Table 1 Software Tools.....	17
Table 2 Skill Requirements.....	18
Table 3 Roles and responsibilities	19
Table 4 Competitive analysis.....	37
Table 5 Definition of ready.....	48
Table 6 project backlog.....	49
Table 7 Demographics of Participants.....	101
Table 8 NFR Testing.....	112
Table9 Appendix Questionnaire's response	125



Table of Figures

Figure 1 the proposed roadmap.....	13
Figure 2 a completed wheel of life [13].....	24
Figure 3 a blank wheel of life [11].....	24
Figure 4 Averages of positive PANAS before and after the journaling [18].....	25
Figure 5 LifeWheel's interface	30
Figure 6 Happy:WheelofLife interface	31
Figure 7 Fabulous' interface	32
Figure 8 EvaluationWheel's interface	33
Figure 9 Ambitionist's interface.....	34
Figure 10 Goal Setting Tracker Planner's interface	35
<i>Figure 11 Having a hard time to set my goals effectively.</i>	42
Figure 12Having trouble for adopting new habits	42
Figure 13 Find it difficult to prioritize my daily tasks.....	42
Figure 14 Suffering from task procrastination.....	43
Figure 15 Rely on a tool to remind me of my daily tasks.....	43
Figure 16 Sharing goals with others helps to achieve them.....	43
Figure 17 What method do you follow to organize your daily tasks and achieve your goals?.....	45
Figure 18 Have you ever tried daily journaling?	45
Figure 19 how much do you prefer to use key phrases as a motivator for journaling?	45
Figure 20 how many tasks can you complete in a day without feeling exhausted or stressed?	45
Figure 21 when you are setting your goals, what information do you want to include about them?.....	46
Figure 22 Motazen's use case diagram	47
Figure 23 Motazen System Architecture [33].....	58
Figure 24 Class Diagram	59
Figure 25 Generate Wheel Flowchart	61
Figure 26 Add Goal Flowchart	62
Figure 27 Ranking Tasks	63
Figure 28 ER Diagram	65
Figure 29 Non-relational data model.....	66
Figure 30 Site Map.....	69
Figure 31 The method for calculating normalized time.....	73
Figure 32 the method for creating a directed graph	75
Figure 33 the method for calculating the dependency	75
Figure 34 the calculate rank method.....	76
Figure 35 Post Model with variables	92
Figure 36 realtime database of post channels: Communities.....	93
Figure 37 realtime database of posts: Messages	93
Figure 38 Message Example	94
Figure 39 Feature Request in Firebase.....	97
Figure 40 histogram of first question's results	102
Figure 41 histogram of second question's results	102



Figure 42 histogram of the second question's results.....	102
Figure 43 histogram of the third question's results	103
Figure 44 histogram of the fourth question's results	103
Figure 45 histogram of the fifth question's results	104
Figure 46 histogram of the sixth question's results	104
Figure 47 histogram of the seventh question's results.....	105
Figure 48 histogram of the eighth question's results	105
Figure 49 histogram of the ninth question's results	106
Figure 50 histogram of the tenth question's results	106
Figure 51 histogram of the eleventh question's results	107
Figure 52 histogram of the twelfth question's results.....	107
Figure 53 histogram of the thirteenth question's results.....	108
Figure 54 histogram of the fourteenth question's results.....	108
Figure 55 histogram of the fifteenth question's results	109
Figure 56 histogram of sixteenth question's results	109
Figure 57 histogram of seventeenth question's results.....	110
Figure 58 histogram of eighteenth question's results.....	110
Figure 59 histogram of nineteenth question's results.....	111
Figure 60 Questionnaire Q1.....	125
Figure 61 Questionnaire Q2.....	125
Figure 62 Questionnaire Q3.....	125
Figure 63 Questionnaire Q4.....	126
Figure 64 Questionnaire Q5.....	126
Figure 65 Questionnaire Q6.....	126
Figure 66 Questionnaire Q7.....	127
Figure67 Questionnaire Q8.....	127
Figure68 Questionnaire Q10.....	127
Figure69 Questionnaire Q11	128
Figure70 Questionnaire Q12.....	128
Figure71 Questionnaire Q13.....	128

Chapter 1:

Introduction



1 Introduction

In today's fast-paced world, individuals have evolved to become more ambitious and goal-driven to keep up with the times, but have their daily habits been adjusted to help achieve their dreams? Studies have shown that out of approximately 41% of Americans who make New Year's resolutions, only 9% of them were successful in keeping them [1]. It is without a doubt that we are living in an age of rapid growth and development, where maintaining resolutions and goals is extremely challenging.

Moreover, balancing various goals has only gotten harder with time as individuals feel compelled to attempt improving multiple aspects of their lives all at once, such as finance, family, health, career, and their social lives. Finding the time, effort, and resources to juggle improving all these aspects is never a simple task, and therefore burdens the attempter's and negatively impacts their well-being [2].

As a result, most individuals today are facing difficulties in forging effective paths to reach their goals and monitor their progress towards them, while also deciding and prioritizing their daily tasks that contribute to these goals according to some factors, such as the goals' due dates/urgency, importance, and complexity based on a Forbes Magazine study [3].

Issues like these are deeply integrated in our daily lives, and a technological solution would be a great step towards overcoming them. Since individuals tend to carry their mobile phones with them at most times, an application that allows its users to be notified about tasks and track their progress towards achieving them would be a convenient way to aid users in creating habits and tasks that help them achieve their goals. Such technology could also be used to improve efficiency by automatically ranking tasks on a user's to-do lists using a simple multi-objective ranking algorithm. Furthermore, people can use the application to share common goals with each other to spark inspiration. The application would also allow users to write down notes through a virtual journal within the application [4]. This section presents an introduction about life coaching and the wheel of life tool, then the project problem description followed by the solution, the product vision, and the product roadmap, then finally the objectives of the project in the following detailed levels:



Product (customer focus-value), Project (solution focus-plan), and Learning (student focus).
Later, we explore the problem's scope, the hardware/software tools and cost, and finally the Scrum team with the skill set requirements and roles with responsibilities.



1.1 The Problem

As people search for success in life, it can be easy to find themselves focusing on achieving goals related to some aspects at the expense of other, more important aspects. The success of life, however, is determined by all its components, not just one or a few, as it is an integrated system.

Besides attempting to juggle multiple goals at once, people face difficulties monitoring their progress towards these goals. As well as deciding and prioritizing their daily tasks that contribute to these goals, according to multiple factors such as goals' urgency, importance, complexity, dependency, etc. According to Forbes Magazine, only 8% of all people complete their new year's resolutions each year [3]. This is largely the outcome of having great expectations for these goals, struggling to uphold them, and watching them fade away after the initial thrill has worn off.



1.2 The Solution

This project aims to help people achieve life balance and accomplish their goals. It applies The Wheel of Life tool to give the user a visual representation of the way their life is currently, against a completely balanced wheel. Some of the main features of this application are as follows:

- The application starts with an assessment for the user to find out which life aspects (e.g., health, family/friends, career, financial, etc.) are balanced and which aspects need improvement.
- The answers are then analyzed, and a visual representation of the user's life wheel is displayed with a diagnosis (balanced/imbalanced).
- The user is then prompted to create a new goal or habit, focusing on areas that need improvement.
- The user has private goals and shared goals with public and private communities.
- Each goal belongs to an aspect and has a to-do list (tasks), due date (sharp/flexible), importance (high, medium, low), and dependency on other goals.
- The user enters his/her priorities for goals' aspects (e.g., health, family, career, etc.) and goal's factors mentioned in the previous point.
- The user can graphically monitor his/her progress towards their goals and view their daily tasks.
- The user can create private communities between friends, create public communities, join communities with common goals, and view the community's progress.
- Given the user's priorities and goal aspect, the application automatically ranks a to-do list using a simple multi-objective ranking algorithm.
- The application has space for daily journaling prompted with ideas related to the user's goals.



1.3 Product

1.3.1 Product Vision

Product Vision:

For users eighteen-years-old and above and speak Arabic

Who want to improve their lives and achieve their goals,

Motazen **is a** life coaching mobile application

That relies on the “Wheel of Life” tool to help visualize different aspects of the user’s life, ranks daily tasks based on their goals, and provides them with a pre-determined journal.

Unlike Life Wheel

Our product offers the ability to invite friends and join communities to achieve a certain goal while engaging in a friendly competition, helps user’s ability to track progress through a pre-determined prompted journal, rank the user’s daily tasks, and most importantly supports the Arabic language.



1.3.2 Product Roadmap

In this section, the product roadmap will be demonstrated as shown in figure 1.1. The First release will include sprints 0 and 1. In sprint 0, our increment will be a document that expands upon the project's proposal by adding the following sections: background, literature review, system description, and the product's backlog. Then, in sprint 1, objectives 1 and 2 will be completed. After the first release, sprint 2 will include sprints 2 through 4. With each sprint, two product objectives will be completed in the numerical order presented in this document.

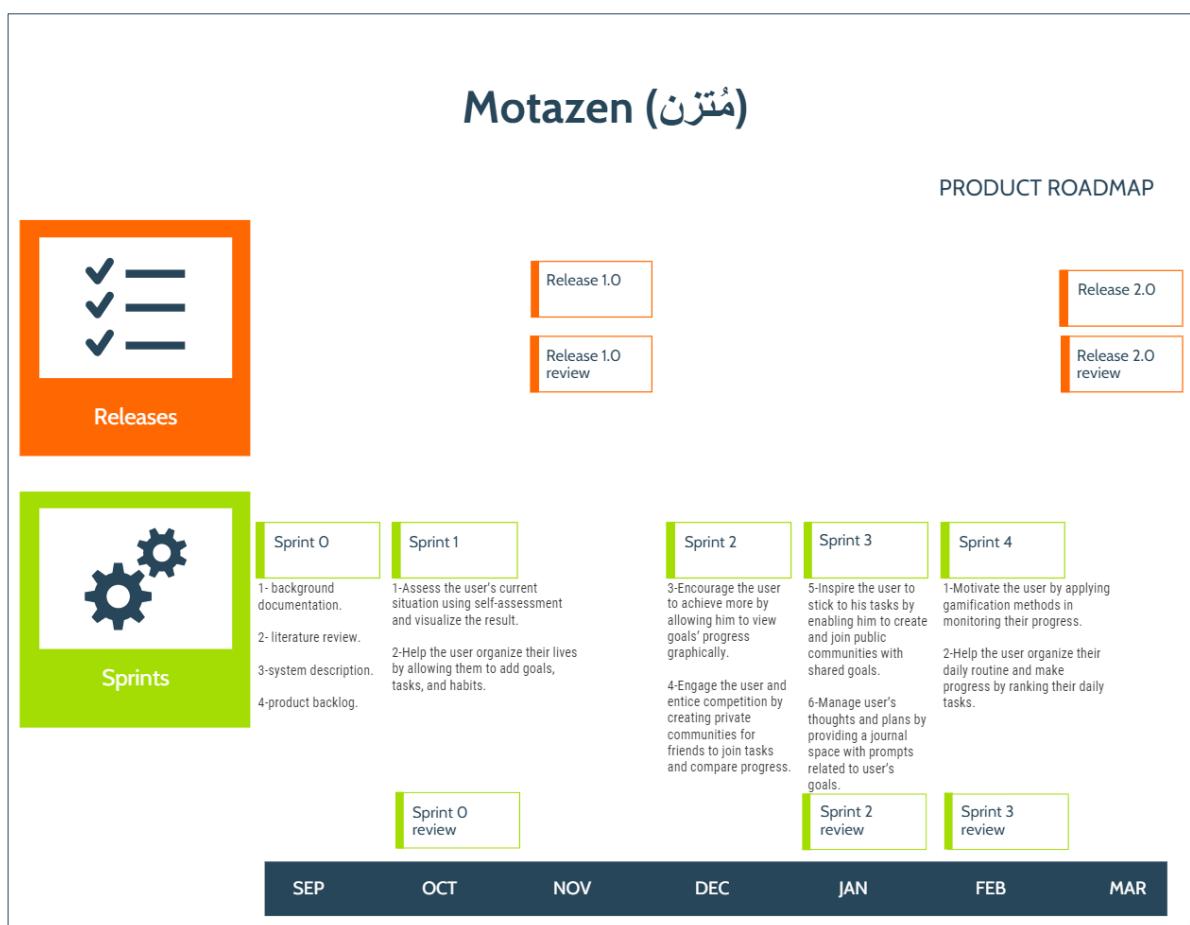


Figure 1 the proposed roadmap



1.3.3 Objectives

- **Product (customer focus-value):**

Motazen mobile application strives to help those who lack consistency by providing a virtual life coaching alternative. Hiring a life coach and attending multiple courses might not be a feasible option for many people yet creating goals and maintaining them is a vital part of self-development, so Motazen aims to fill that gap. It will help those who have a hard time building good habits by applying the Wheel of Life methodology and assessing each user's needs based on a short survey, and create a clear, direct path as to how to reach their goals.

Motazen will provide the following features:

1. Assess the user's current situation using self-assessment and visualize the result.
2. Help the users organize their lives by allowing them to add goals, tasks, and habits.
3. Encourage the user to achieve more by allowing them to view goals' progress graphically.
4. Engage the user and entice competition by creating private communities for friends to join goals and compare progress.
5. Inspire the user to stick to their tasks by enabling them to create and join public communities with shared goals.
6. Manage user's thoughts and plans by providing a journal space with prompts related to user's goals.
7. Motivate the user by applying gamification methods in monitoring their progress.
8. Help the user organize their daily routine and make progress by ranking their daily tasks.



- **Project (solution focus-plan):**

To be able to present an accurate depiction of the imbalances in people's lives and find a working solution, we need to acquire requirements by using questionnaires. We will first collect data from the Wheel of Life assessment and analyze it based on a Wheel of Life school. Then, by sending out public questionnaires as part of gathering requirements, we will carry out an analytic method of the survey's results. We will also design the necessary gamification techniques to engage our users, as well as design the ranking algorithm, which would be accomplished by using a weighted sum formula where we will multiply each number by its weight, then add the results. The weight value will vary depending on the user's priorities. Later, we will design a user-friendly interface by applying UX principles, and finally developing the software and testing it.

- **Learning (student focus):**

We will first learn about Xcode environment, followed by how to build and program mobile applications using Flutter's Dart. Beyond that, we will learn about how gamification can be used in IT projects to increase participation, engagement, and interact with users by connecting it to motivation theory. In addition to that, we will learn how to rank objects based on multiple criteria by applying a simple multi-objective ranking algorithm.



1.3.4 Scope

Motazen is a life coaching iOS mobile application developed using Flutter's Dart framework that supports the Arabic language. It aims to help users that are 18 and above who need guidance in improving their quality of life, and struggle to maintain goals and habits by assessing the user's current life balance in accordance with the Wheel of Life's main aspects which are: Finance, Health, Family & Friends , Significant Other, Personal Growth, Physical Environment, Career, and Entertainment [5]. The user can select either some or all of the aspects out of the previously mentioned 8 main aspects, based on what is deemed more important to them.

The application will rank the user's daily tasks based on the user's priority. The accommodating journal space will be used with a pre-determined list of prompts. The application will allow the creation and joining of public and private communities where members can exchange messages. Sub aspects and adding personal aspects to the Wheel of Life are out of this project's scope, as well as setting schedules and appointments in the mobile's calendar.



1.3.5 Hardware/Software Tools and Costs

Table 1 describes all hardware and software tools that will be used during this project.

Table 1 Software Tools

Software Tools	
Name	Description
Xcode	Apple's IDE (Integrated Development Environment) that is used to develop software for Mac and iOS applications. It includes the iOS SDK (Software development kit), tools, compilers, and the frameworks needed to design, develop, and debug iOS applications. [6]
Flutter: Programming Language	Google's open-source Flutter framework enables you to create multiplatform, natively built applications from a single codebase. The six platform targets supported by Flutter are web applications, Windows, macOS, iOS, Android, and Linux. Flutter consists of two important parts: <ul style="list-style-type: none">• An SDK: toolkit that application developers use to build applications via prebuilt components instead of building each component individually. This includes tools to compile code into native machine code (code for iOS and Android).• A Framework: a tool that offers pre-made elements or solutions that can be tweaked to hasten the development process.[7]



1.4 Scrum Team

1.4.1 Skill Set Requirements

Table 2 demonstrates the skills required of the development team to be able to complete the requirements of the project.

Table 2 Skill Requirements

Technical Skill Required	What is the current level of the team for each skill? How will the gap be bridged? (if necessary) Learning plan
Programming Skills using Flutter	Level of the team: Beginner. Learning plan: Learn basics of Dart programming, practice building apps using Flutter framework.
Ranking Algorithm	Level of the team: Beginner Learning plan: Understand the concept behind the weighted sum, how to apply it using real-world examples.
Gamification	Level of the team: Beginner Learning plan: Understand Maslow's Hierarchy of Needs, how it applies to gamification, and how to effectively gamify the application.

1.4.2 Learning

We have conducted several activities and steps to ensure that this learning experience is as smooth and efficient as possible, the steps are as follows:

- Joining a Flutter course
- Joining a Flutter learning mailing list
- Experimenting with a small Flutter project
- Exploring the XCode environment and Flutter libraries



1.4.3 Roles and Responsibilities

Table 3 will cover the roles and the responsibilities of all members of the scrum team.

Table 3 Roles and responsibilities

Scrum Team	
Product Owner:	Dr. Alhanoof AlThnian Dr. Khulood AlYahya
Developers:	Reema AlSaif Randa Bakhshwain Reem AlRossais Manar AlMalki
Scrum Master (SM):	Dr. Maha Alyahya Dr. Rana Alkadhi Dr. Hend Alrasheed
Stakeholders:	Users Development Team

Chapter 2:

Background



2 Background

In this section, we discuss certain information related to topics of interest in the field of our application to gain better insight and have a more comprehensive view of necessary knowledge for the application's implementation. We begin with a brief overview of our field, that is, life coaching, then we move on to discuss the tools and techniques we plan to use, i.e., the Wheel of Life tool, pre-determined journaling technique, ranking algorithm, and gamification.

2.1 Life Coaching Tools

With the help of various tools, methods, and strategies, life coaches work to assist people enhance the quality of their lives. Life coaching typically focuses on assisting individuals with problems that interfere with their everyday lives by offering a fresh perspective on them or helping them develop positive habits to experience greater fulfillment or accomplish a specific goal. Although often mistaken to be the same, life coaching is not therapy. Unlike therapy, life coaching is not meant to help with any psychological issues, rather it is meant to help people with other issues regarding changing and improving general aspects in their lives, such as finances, their career, etc.[\[8\]](#)



2.2 The Wheel of Life tool

The Wheel of life is a circular diagram that is divided into sections based on different aspects of a person's life, these aspects usually represent an individual's top priorities during that time and what they plan on improve. The different aspects of the wheel are subject to change with the changing lifestyle and priorities of its user. For example, if the person was once a student, their main aspect would be studying, but after that period of their life has passed, or after they graduate, their main aspect might change to career. [9]

The concept of the Wheel of Life was first introduced in Buddhism, and was called "The Bhavachakra in Sanskrit", in which the wheel was divided into six different major sections that can be understood as the different states of mind. [10] During the 1960s, the Wheel of Life has resurfaced as a life coaching and business tool for development and growth, as it was used as a method for measuring balance. Some reasons that attributed to the popularity of this tool is its briefness and simplicity; as it does not need any extra or specialized tools to work, all that is required is for the individual to take a short assessment and fill the based on the results of the test, which will be discussed in more detail later in this section.

As mentioned before, the Wheel of Life is divided into several sections that represent different aspects. Although the aspects differ from one individual to another, and from school to another, there are always fixed common aspects that are used when creating a Wheel of Life and they include: career, finance, spiritual, physical, intellectual, family, social.

- **Career:** Includes all types of careers, whether the individual is business owner, a freelancer, works a 9-5 job, or is retired. The aim of adding career to the wheel is usually to have an idea of how a person is doing in their career in terms of earnings, success, etc.
- **Finance:** It is actively involved with the career aspect and focuses on goals related to personal finances rather than earnings. For example, a finance driven goal would be to save a certain amount of money or reach a certain revenue in investments.
- **Spiritual:** The spiritual aspect is not meant to represent a particular system, rather it is meant to represent any system whether it is a religion or a type of meditation. This aspect focuses on how individuals are doing spiritually.



- Physical: This aspect focuses on health and wellness of the individual in general and how it is maintained. An example of goal in this aspect would be walking a certain number of steps in a day or maintaining a healthy diet.
- Intellectual: This aspect is interested in the pursuit on knowledge, and the stimulation of one's intellect through any form of media. An example of goals in this aspect would be to finish reading a book in a certain amount of time or to research an interesting subject.
- Family: Includes immediate and extended family; it focuses on an individual's relation with his family and how it is going. A good example for a goal in this aspect would be visiting one's grandparents once a week.
- Social: Focuses on social interactions outside of one's family members. These groups include friends, coworkers, classmates, etc. An example of this is joining a club or going to a conference. [11]

Finally, we will delve more into detail about how the Wheel of Life is created. To begin, approximately five to nine aspects that hold the highest priority in a person's life are identified, the number of aspects differ according to the Wheel of Life school used. After identifying these aspects, we move on to draw a circle divided into the number of the identified aspects, Figure 2.2 shows what an empty wheel of life should look like. Then, the satisfaction level of each aspect is analyzed by comparing it to what the "ideal situation" would be for that aspect, thus making this analysis personal and unique to each person. After completing the analysis, the current situation of each aspect is reviewed against the ideal situation and is given a tangible indication (as a percentage or points) of how far the current situation is from the ideal situation. Finally, each field is filled respectively from the center to the periphery according to the acquired indications, Figure 2.1 is an example of a completed wheel of life. [12]

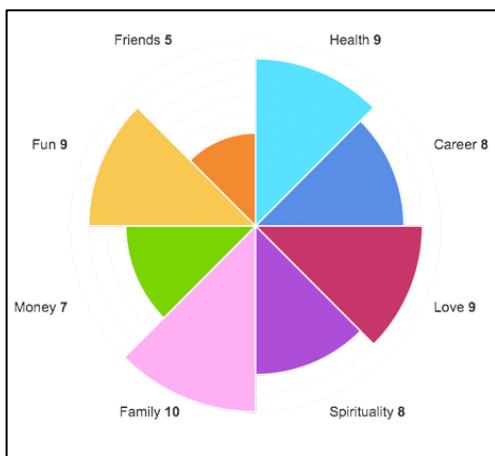


Figure 2 a completed wheel of life [13]

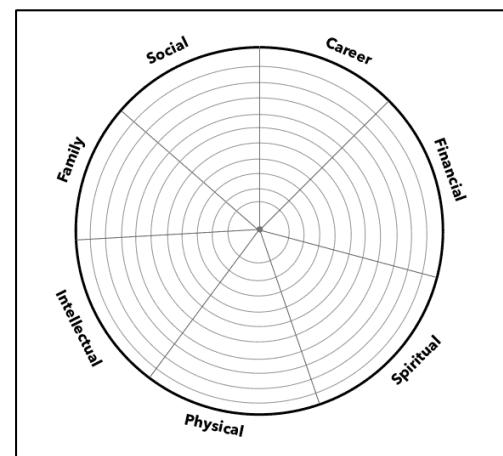


Figure 3 a blank wheel of life [11]

2.2.1 The Concept of Journaling

The concept of journaling is much deeper than simply keeping a straightforward log of the events of a day. A person might journal to express his thoughts, get in touch with his emotions, set goals and monitor his recent progress [14]. The benefits of journaling for mental health and wellness are numerous. Frequent journaling can lead to a decrease in stress, anxiety, and depression. Using a journal is a great way to label unhealthy aspects of a person's life, free anxious thoughts, and be more open to alternatives. In addition to improving the overall health of a person's mental state, journaling is an effective way to provide a sanctuary self-space where a person can prioritize problems and worries, talk positively to himself, and convert negativity into creativity [15].

A study has been conducted on college students to illustrate which technique is the most effective for stress management in such demographic, and the result was that journaling was the most effective technique due to its feasibility, acceptability, and convenience [16]. For these and similar reasons, as well as the result of people getting busier, more distracted, and impatient, people now prefer journaling online rather than using the traditional ways. Journaling online also has the advantage of availability, where a person can journal anywhere at any time [17].

There are different ways of journaling such as reflective journaling, prompted journaling, and freestyle journaling. Reflective journaling is a reflection on a learning experience, it is used to

analyze growth and understanding of the learned material. Freestyle journaling is the most common journaling technique where a person can just write about anything that comes to mind, unlike prompted journaling, where the person will be responding to a given prompt. There have not been sufficient studies regarding best practices for journaling. However, Rachel Goluboff, a student at Berkeley Carroll school, has studied the impact of prompted and unprompted journaling on high school students' moods. Rachel used the Positive and Negative Affect Schedule (PANAS) survey to determine their baseline mood and their mood after 4 weeks of journaling. The outcomes showed that the prompted journaling group indicated a larger positive change in mood than the unprompted journaling while the overall result didn't show a significant shift. See figure 2.1 [18].

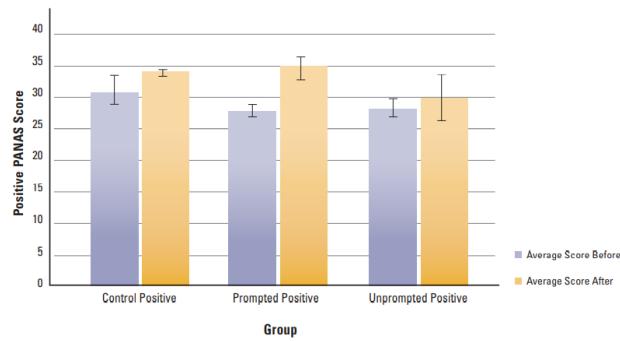


Figure 4 Averages of positive PANAS before and after the journaling [18]

2.2.2 Gamification

Put simply, gamification is the process of “gamifying” non-game contexts to increase user engagement. What that means is bringing game mechanics, elements, and incentives to tasks that are usually not deemed enjoyable. The way gamification works is it leverages human’s innate tendencies for competition, success, and achievement. In order to encourage people to reach their goals or improve performance, game design techniques like badges, “leveling up,” and rewards for accomplishments are translated into the real world. This works due to the fact that it combines trigger-action-feedback in a closed loop with routine-based activities, leaving the end user yearning for more. When implemented correctly, gamification has proven to be a success across different industries [19].

Successful gamification combines features that work well as intrinsic motivation. This results in users interacting with the system happily and willingly. For instance, Foursquare/Swarm



allows users to compete for first place when dining, shopping, watching movies, and other activities by promoting them to "Mayors" of venues after a certain number of visits.

By incorporating gamification into an existing system, we can motivate users by bringing fun, and game like structure to "serious" or boring tasks. There are multiple ways to gamify systems, from countdowns to badges to rewards for completing a task up to a certain percentage, which works well to monitor progress and motivate the user to keep pushing.

2.2.3 Ranking algorithms in multi- criteria decision solving techniques

Multiple applications in various fields use methodologies that help them make decisions. One of these methodologies is using the MCDM (multi-criteria decision-making) ranking algorithm that classifies alternatives based on the sum of weighted attributes, where each weight reflects importance and priority, and eventually makes the decisions of how much that attribute contributes to the weighted sum ranking. [20]

The main reason for using the ranking method is to prioritize the alternatives based on users' needs. MCDM's strong suit is that it is involved with constructing and resolving multiple-criteria decision and planning problems. The goal is to assist decision-makers who are dealing with these issues. For such challenges, there typically isn't a single best answer, hence it's vital to compare ideas using the decision-makers' preferences.

Moreover, we can also apply the ranking method by combining several different methods that have been previously developed to take multiple criteria priorities and take the users' needs into consideration [21] . Additionally, in its nature, MCDM solves the bias case problem for the attributes in the data, so that the user does not make a biased decision.

Users can also use various libraries and packages in different programming languages that will help with multi-objective optimization [22]. And there are multiple methodologies called Generic Methodologies, which are customary between many problem solvers and scientists that the user may also easily use:

- Maximize and minimize one attribute at a time to get the total score.
- Assigning weights to each attribute to get an optimized weighted score for all attributes.

- Combine the attributes for each score (weighted scores) to finalize and get the final score as a whole number.

Finally, every decision made can be developed using many criteria with its corresponding importance assigned by each criterion. [23]

Chapter 3:

Literature Review



3 Literature Review

In this section, we reviewed similar applications in the life coaching field and reviewed the features that add to their market value, then we proceeded to compare the rivaling applications against Motazen to learn more about this market, improve our functionalities, and avoid some of the pitfalls other similar applications face.

3.1 Competitive Product Analysis

In this section we discuss rivaling application in terms of their main features, discuss some of their drawbacks, and compare them to Motazen.



3.1.1 LifeWheel—Goal Setting Tracker

LifeWheel is our primary competitor in this field. The application visualizes the user's life wheel that is composed of several pre-generated wheels based on an assessment. After that, it allows the user to edit their life wheels, add habits, goals, and tasks related to each wheel, as well as set reminder for them. From then on, they can add them to the calendar and search through them. [24]

It also allows the user to record their progress and recommends classes related to the pre-generated wheels. However, Life wheel, like many other life coaching applications, requires a monthly paid subscription to access all its features which might not be suitable for all users.

LifeWheel's main features:

- Visualization: Users can view a visualization of the current balance of their Life spheres.
- View progress: Users who own accounts can view their progress in each life sphere and the regularity of their habits represented as charts over the week, month, or year.

- Edit life wheels: Users can personalize life spheres by changing their icon or color, as well as adding entirely new wheels or deleting old ones.
- Journaling: Users can journal starting from an empty page or from a template. Moreover, users can edit templates by changing its contents and icon, along with being able to delete and add templates.
- Adding habits and goals: Users can add habits, goals, and subtasks under each goal. They can also add suggested habits.
- To-do list: Users can add to, view, and edit their to-do list

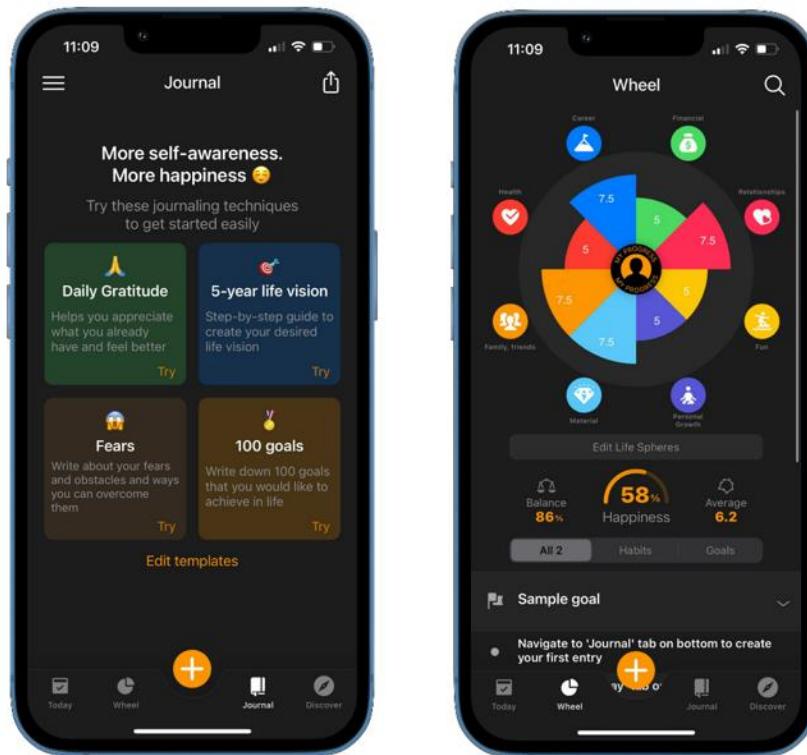


Figure 5 LifeWheel's interface



3.1.2 Happy: Wheel of Life

Happy is a life coaching application that uses the “Wheel of Life” methodology to help users maintain and track goals and habits. It visualizes a life wheel based on certain roles the user associates themselves with. Users can edit their roles, add tasks related to roles, and set reminders for them. Users can also create daily routines. [25]

Happy’s main features:

- Visualization: Users can view a visualization of the current balance of their roles.
- Edit roles: Users can personalize roles by changing their title or color, as well as adding an entirely new role or deleting an old one.
- Add tasks: Users can add tasks related to each role, each task can have a specific date and time. Tasks can also have reminders or be added as routines.
- Library: Users can access a library that contains helpful materials about the concept of the wheel of life and other topics related to improving the quality of life.
- To-do list: Users can view their to-do list

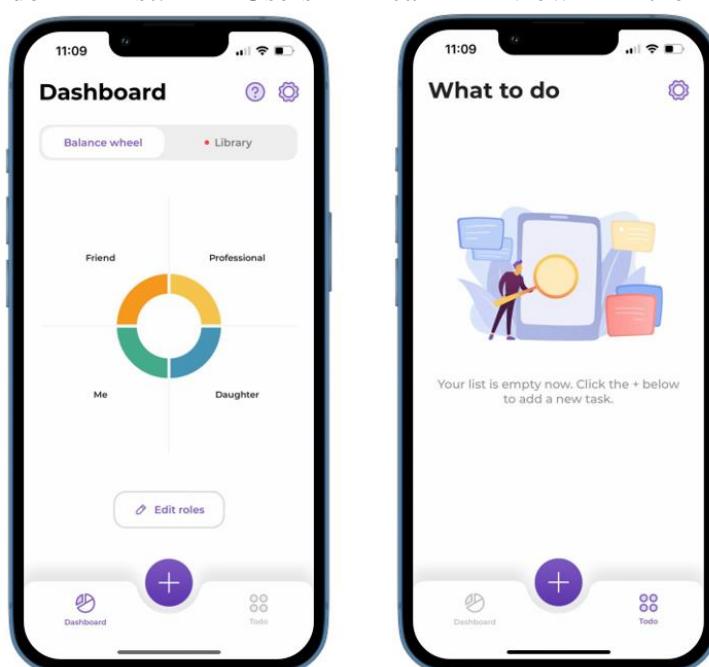


Figure 6 Happy:WheelofLife interface



3.1.3 Fabulous

The Fabulous app is a self-care and life coaching application that harnesses the power of behavior based on science. It helps users develop lasting healthy habit by allowing them to create meaningful daily rituals and stack habits to create routines that guide the user towards achieving their goals. [26]

Fabulous' main features:

- To-do list: Fabulous allows the user to select several tasks for each part of their daily routine, such as studying, working, exercising, etc. A list of suggested habits will be provided, but the user can still create personal habits. [27]
- Community: Users with common goals can connect with each other through this feature to share stories and ask each other questions. Users should first select a certain number of goals before joining a community. Communities also include challenges to entice and encourage users. [27]
- Keeping Track: Fabulous allows the user to keep track of their progress and users can monitor it through their calendar. As a user, you can see your current progress. [27]
- Motivational templates: To motivate the user more, the application shows inspirational templates and infographics. They can be used to keep the user's productivity in check, excite the user to swap out bad habits with better ones. [27]

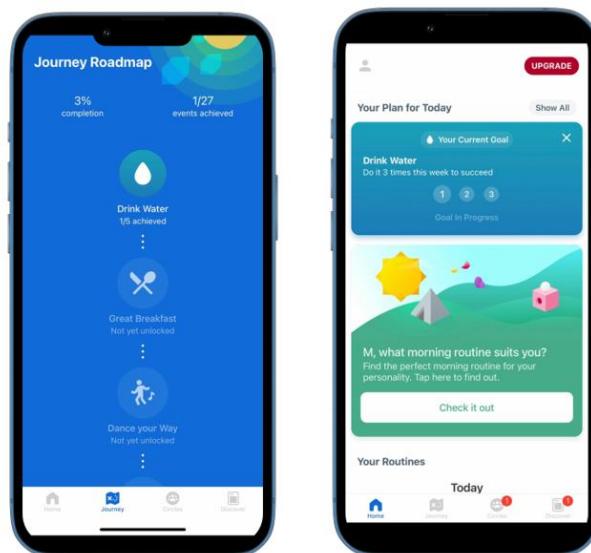


Figure 7 Fabulous' interface



3.1.4 Evaluation Wheel

Figure 8 showcases the simple functionalities of the Evaluation Wheel mobile application. To begin, it is only a visualization tool, where the different aspects of the wheel are modifiable, both in aspect title and balance level. It contains a home page that works by simply dragging your finger across each aspect of the evaluation wheel to provide a score between 1 and 10. Then, the user may copy the wheel to your preferred notes app or keep it in their photos application for later reflection. The user can also share the wheel with a friend or coach.

Evaluation Wheel main features:

- Visualization tool: In Evaluation Wheel, the user can both view and edit their Wheel of Life manually, which can be viewed later as well for reflection.
- Share with friends: Users can share their wheels as a .png image with their family, friends, or life coach.
- Modify Aspects: Because every person has different priorities, their chosen aspects would differ as well, so the application allows users to modify the aspects of the wheel in case the original wheel do not resonate with user's goals.



Figure 8 EvaluationWheel's interface



3.1.5 Ambitionist

Figure 9 is the interface of the Ambitionist App, an application for tracking personal resolutions. It serves as a long-term planner for personal development and tool for creating general bucket lists [28]. It offers two versions, a free version where users can choose up to three life goals, and a premium version with an unlimited number of life goals. The application directs the user towards gaining a better perspective of their life and derive motivation from recording their accomplishments. Ambitionist allows the user to create a personalized life area in case the user doesn't find their chosen aspect amongst the choices. After selecting a life area, the user can either select a goal from the ready, modifiable template or manually add goals. Under each goal is a set of tasks, displayed as a simple to do list without extensive task features such as due date, frequency, dependency etc.

Ambitionist main features:

- Focus Goals: Users can pick a “focus goal” that shows up in a separate tab to emphasize it and helps the user prioritize it over other goals [28].
- To-do list: Ambitionist allows the user to add tasks under each goal and mark them as checked/unchecked, “in focus” or not, as defined previously [28].
- Achievement Tab: Contains latest achievements and sorted by the year, as well as sidebar tab where user can either manually enter achievements or choose from list [28].
- Color Coding: Setting a color for each life area which gets inherited to goals, tasks, and achievements so user can visually distinguish them [28].

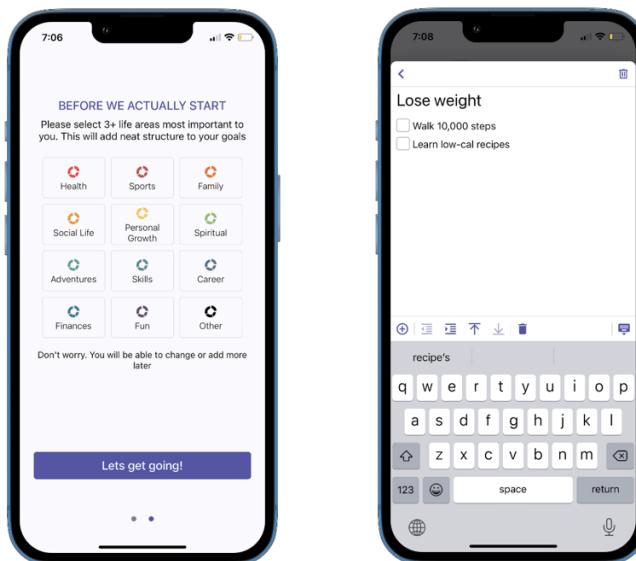


Figure 9 Ambitionist's interface



3.1.6 Goal Setting Tracker Planner

This application assists the users to discover, plan, and track their goals. It offers a step-by-step guide to assess the user's life and determine the areas that need development.

In Goal Setting Tracker Planner the user will be able to:

- Create a life chart by rating different aspects of their life.
- Set Long-Term goals, Recurring goals, and plan their day.
- Track their goal progress.
- Journal using general journal prompts and questions or freestyle writing.

The application has a premium subscription that allows the user to add specific questions in the journal and share their life chart, goals, and plan with a supportive team.

The application features a user-friendly interface that facilitates the planning and tracking process. Also, it gives the user related suggestion when setting a new goal or activity. See Figure 10 [29].

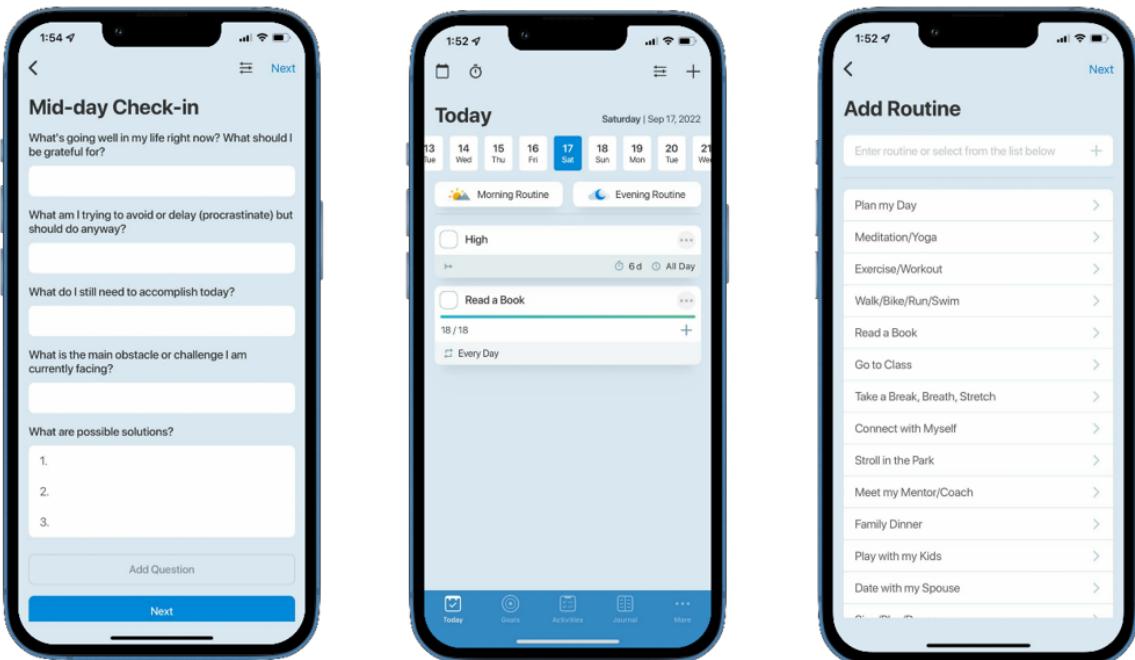


Figure 10 Goal Setting Tracker Planner's interface



The Wheel of Life tool is used in numerous applications to help assess and improve a user's unbalanced life areas by setting new goals, tasks, habits, and monitoring their progress along the way. In addition to that, motivate users by joining communities with common goals, sharing these goals with friends, and journaling. However, most, if not all, of these applications do not support the Arabic language, and only one of the applications mentioned above assist the user in setting a goal. Additionally, we can see that all these programs rely on the users to sort their daily tasks to help them reach and maintain their goals; due to that, users feel overwhelmed by the amount of work they must do, which could lead them to concentrate on less important tasks, and neglect more valuable ones. By ranking tasks, the user can comprehend their plan as well as the order in which they will carry out their tasks, so that they can tell what needs immediate attention and what can be left for another time. This will allow the user to be more organized and reliable, which will in turn cause them to feel less stressed and more productive [30]. In Motazen, the user's job is to simply provide the goal and tasks with the necessary information, and the application will guide the user by ranking their daily tasks based on priority. Table 4 illustrates a comparison between Motazen and all its competitors.

Table 4 Competitive analysis

Application Features	Motazen	Life Wheel Goal Setting Tracker	Happy: Wheel of Life	Fabulous	EvaluationWheel	Ambitionist	Goal Setting Tracker Planner
Wheel of life assessment and visualize	Yes	Yes	Yes	Yes	Yes	No	No
Add goals, habits, and subtasks	Yes	Yes	Yes	Yes	No	Yes	Yes
Add goal's detailed information such as: duration, due date, and steps	Yes	Yes	Yes	No	No	Yes	Yes
Monitor the user's progress	Yes	Yes	Yes	Yes	No	Yes	Yes
Create communities	Yes	No	No	No	No	No	No

Application Features	Motazen	Life Wheel Goal Setting Tracker	Happy: Wheel of Life	Fabulous	EvaluationWheel	Ambitionist	Goal Setting Tracker Planner
Join Public Communities	Yes	No	No	Yes	No	No	No
Join Private Communities	Yes	No	No	No	No	No	Yes
View community's member progress	Yes	No	No	No	No	No	No
Journaling	Yes	Yes	No	No	No	No	Yes
To-do List	Yes	Yes	No	No	No	Yes	Yes

Application Features	Motazen	Life Wheel Goal Setting Tracker	Happy: Wheel of Life	Fabulous	EvaluationWheel	Ambitionist	Goal Setting Tracker Planner
Ranking Daily Task	Yes	No	No	No	No	No	No
Language	Arabic	English	English	English	English	English	English, French, German, and Portuguese

Chapter 4:

System

Requirements



4 System Requirements

This section will review users of the application by defining each user within the system using their educational level and technical expertise. It will also include the requirements elicitation where we describe the requirement elicitation methods used to formulate our requirements, along with a summary of the requirements elicitation, including the user case diagram.

4.1 System Users

Although life coaching and the Wheel of Life tool are applicable to users of all ages, every range of ages has a set of life goals, different quality of life, and diverse life aspects to focus on. Therefore, based on the aspects of the Wheel of Life provided in Motazen application, we targeted users aged 18 and above.

The users do not need to be highly technically skilled of any sort. Operating the application only requires the ability to download the app from the App Store and start using the mobile application on a device that supports iOS. The user's fluency level in Arabic should range from medium to high, so that they can interact with the application's Arabic interface and use its features such as taking the assessment, creating goals and habits, participating in the communities, interacting with the prompting journal, etc.

4.2 Requirements Elicitation and Analysis

We designed an electronic questionnaire using Google Forms to guide our requirements elicitation process. The questionnaire consists of 13 questions and was distributed among 159 Arabic speakers that are eighteen years old and above. The questionnaire is available in Appendix A. Presented below are the responses we received, displayed graphically alongside a graph key (legend).

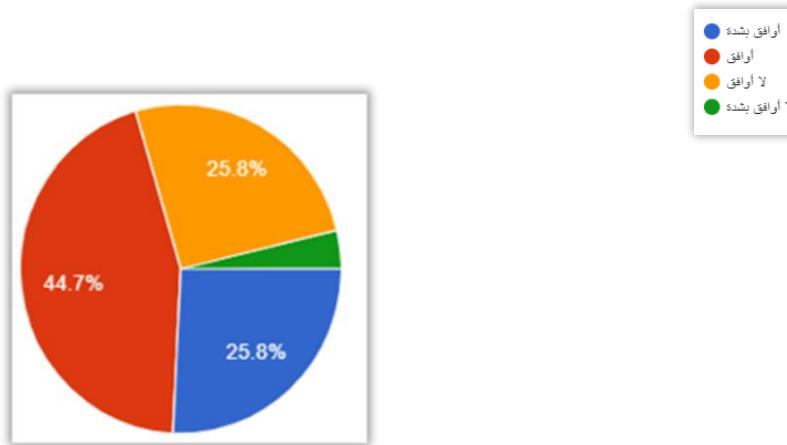


Figure 11 Having a hard time to set my goals effectively.

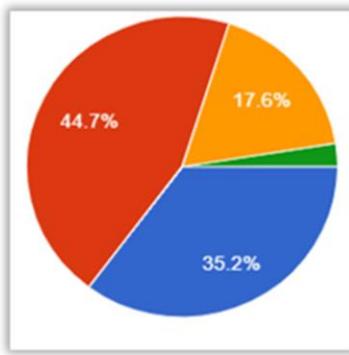


Figure 12 Having trouble for adopting new habits

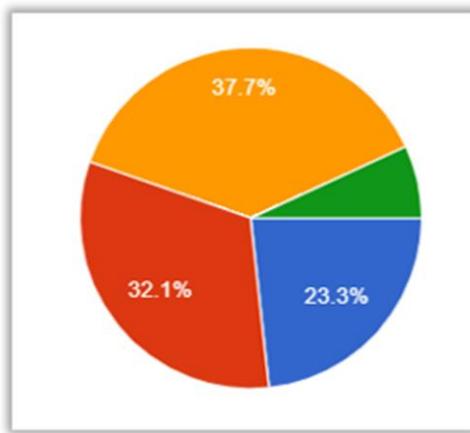


Figure 13 Find it difficult to prioritize my daily tasks

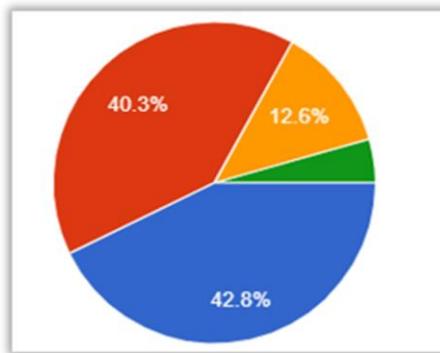


Figure 14 Suffering from task procrastination

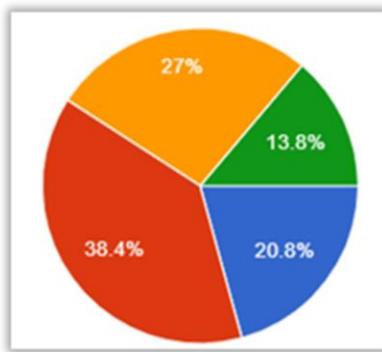


Figure 15 Rely on a tool to remind me of my daily tasks

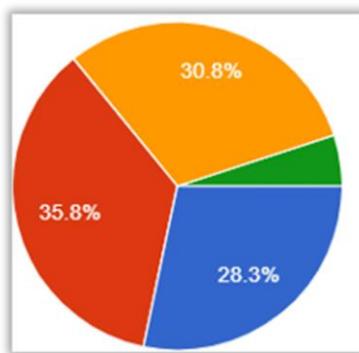


Figure 16 Sharing goals with others helps to achieve them



We can see in figure 12 that about 44.7% of users face difficulty in determining their goals effectively, therefore users need a way to determine which life aspect need improving to create an effective goal. In figure 13, about 44.7% of users agree and 35.2% strongly agree that they face difficulties in adopting new habits. Therefore, using gamification methods like a progress bar or celebratory sound will entice users and encourage them to stay on track. As shown in figure 14, about 37.7% of users didn't agree with the statement and believe that prioritizing their tasks is not difficult. Alternatively, 55.4% (32.1% + 23.3%) of users agreed (and strongly agreed) that it is indeed difficult to prioritize tasks. Therefore, automatically ranking tasks based on importance leads to better productivity and time management for those who face difficulty making such decisions. Figure 15 shows that about 42.8% of users strongly agree and 40.3% agree, respectively, that they lack discipline and suffer from procrastination when it comes to doing their required tasks. Meanwhile, only 12.6% of users do not suffer from this issue. Thus, it is substantial to have deadlines and reminders that notify users to perform their tasks. An extension of the previous question, figure 16 demonstrates the results of question 5 where 64.1% of users are already using applications that track their tasks, such as iPhone reminders, To-do, etc. On the other hand, only about 30.8% of users are not using any task tracking tool. Hence, most users heavily rely on technical solutions and methods for creating and tracking tasks. Figure 17 shows that about 59.2% of users agree, 20.8% of them strongly so, that sharing their goals with other people is a helpful way to accomplish and achieve goals. In contrast, about 27% of users do not believe that. Since most users agreed with the statement, it can be concluded that being a part of community can help users motivate each other.

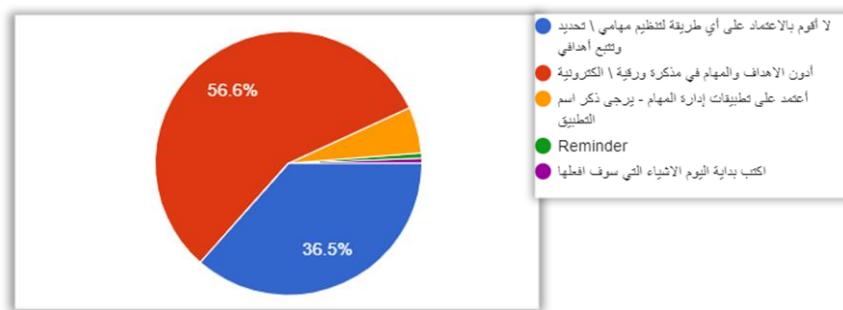


Figure 17 What method do you follow to organize your daily tasks and achieve your goals?

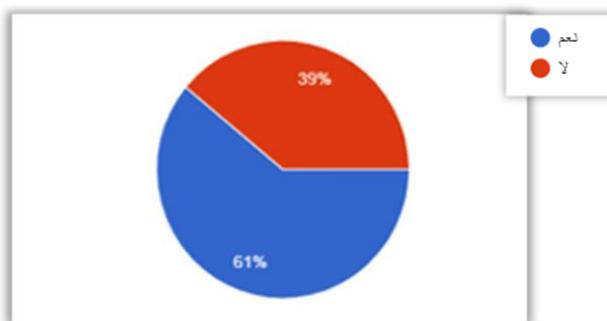


Figure 18 Have you ever tried daily journaling?

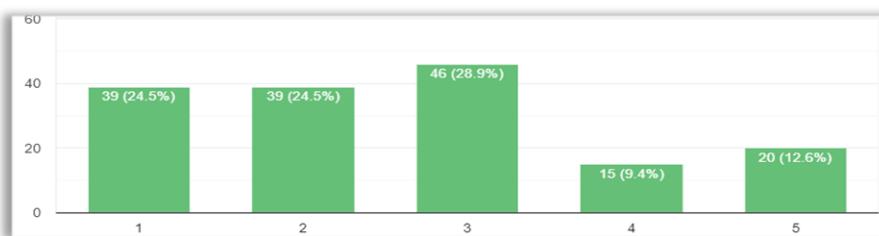


Figure 19 how much do you prefer to use key phrases as a motivator for journaling?

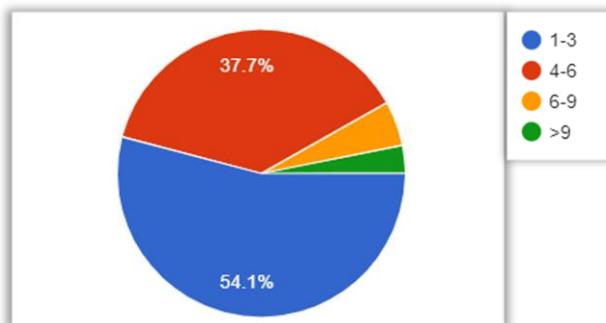


Figure 20 how many tasks can you complete in a day without feeling exhausted or stressed?

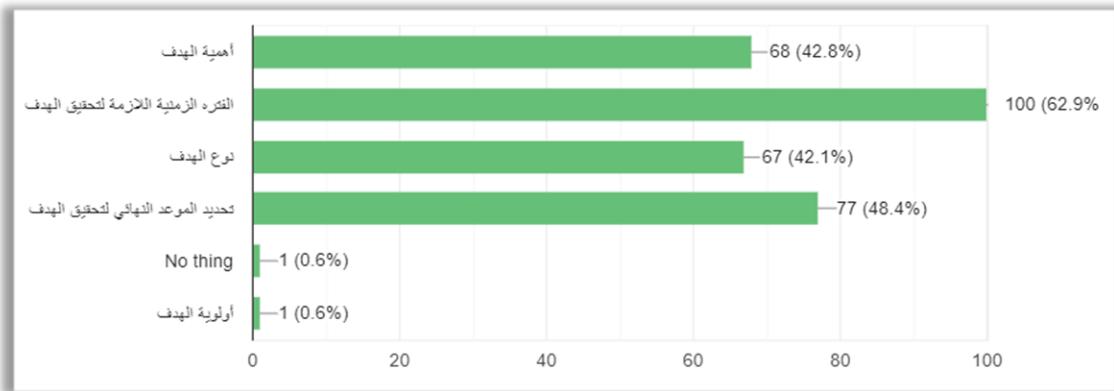


Figure 21 when you are setting your goals, what information do you want to include about them?

4.2.1 Summary of The Questionnaire

Based on the analysis above, Motazen will contain the following features: the user is prompted to create a new goal or habit based on unbalanced areas that need improvement. Each task has a duration, due date, and priority. Users can graphically monitor their progress towards their goals using a progress bar. Users can also view their daily tasks, which are automatically ranked in a list according to the user's priority. Furthermore, users can create or join private and public communities. Motazen will include a space for daily journaling with pre-determined prompts.

Finally, [figure 27 \(Appendix A: questionnaires responses\)](#) shows apps are sharing similar features as Motazen but, unlike Motazen, most of them do not support the Arabic language.

4.3 User Interactions

The figure below shows the user using Motazen and its main functionalities with a clear view of the steps the user will take to accomplish a specific task.

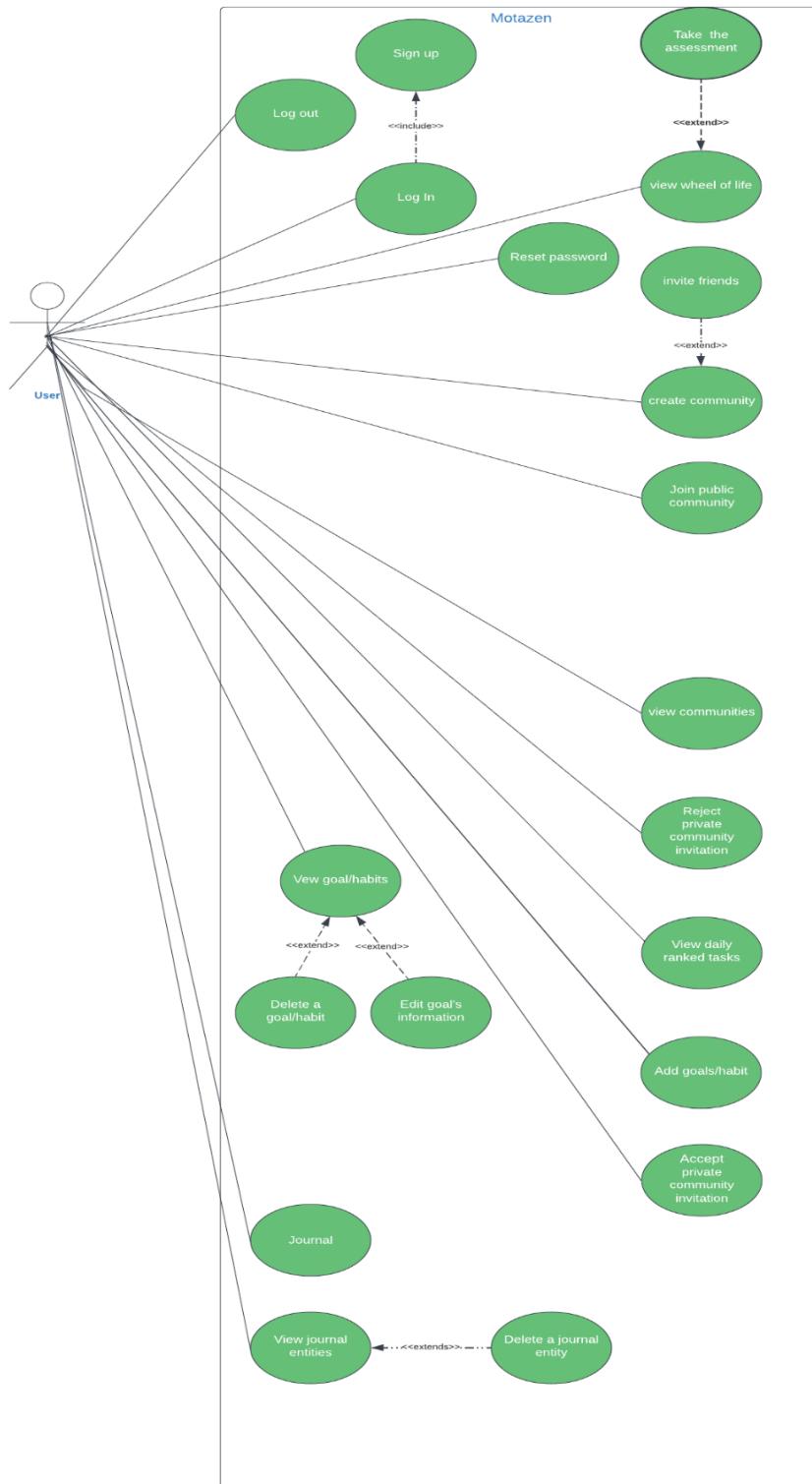


Figure 22 Motazen's use case diagram



4.4 Product Backlog

The table below describes the requirements that must be met for a story to move from the backlog into development.

Table 5 Definition of ready

Definition of Ready	
<input type="checkbox"/>	Business value is clearly articulated
<input type="checkbox"/>	Details are sufficiently understood
<input type="checkbox"/>	If there are dependencies, they were refined and separated
<input type="checkbox"/>	Team is appropriately staffed relative to the PBI
<input type="checkbox"/>	Estimated and small enough to be completed during sprint
<input type="checkbox"/>	Acceptance criteria are clear, testable, and defined.
<input type="checkbox"/>	Performance criteria, if any, are defined and testable
<input type="checkbox"/>	The user story has been estimated and is under a certain size
<input type="checkbox"/>	The team has what they need to start working on the story, but some details will be discovered during the development.

Table 6 will review the product backlog of Motazen, which includes the user stories arranged in order according to priority, as well as the acceptance criteria that had to be met in order for each user story to be considered complete.

Table 6 project backlog

PBI	Size	Type (Feature, defect, technical work, knowledge acquisition)	Acceptance Criteria The conditions of satisfaction that must be met for that item to be accepted.
1. As a new user, I want to sign-up my account so that my information is saved, and I can start using the application.	2	Feature	As a new user, <ul style="list-style-type: none"> – If I open the sign-up page, then I will be able to enter my username, email, and password. – If I open the sign-up page and enter my username, email, and password, and click on the “sign up” button, then my information should be saved, and I can start using the application – If I open the sign-up page and enter any of the following information incorrectly: username, email, or password, then register fails and a detailed error message will be displayed.
2. As a user, I want to log in, so that I can start using the application and grant access to my account.	2	Feature	As a registered user, <ul style="list-style-type: none"> – If I launch the system, then username and password fields should appear. – If I fill the username and password fields with the correct information, and click on the "log in" button, then I will be able to start using the application. – If I enter either my username or password incorrectly, then log in fails and a detailed error message will be displayed.
3. As a user, I want to log out of my account so that I prevent anyone from accessing my account.	2	Feature	As a registered user, <ul style="list-style-type: none"> – If I click on “Log out” button, then a confirmation message with two options “Confirm” and “Cancel” appears. – If I click on the confirm button on the confirmation message, then the log out process succeeds. – If I click on cancel button on the confirmation message, then the log out process fails.
4. As a user, I want to reset my password so that I can still log into my account if I forget my password.	2	Feature	As a registered user, <ul style="list-style-type: none"> – If I click on “Forgot password?” in the log in page, then I will be redirected to reset password page and I will be able

PBI	Size	Type (Feature, defect, technical work, knowledge acquisition)	Acceptance Criteria The conditions of satisfaction that must be met for that item to be accepted.
			to enter my email and receive a reset password code. - If I create a new password and click on “Next” button, then I will be directed to the log in page again
5. As a user, I want to take the wheel of life assessment, so that I assess my current situation.	2	Feature	As a registered user, - If I signed up/logged in successfully, then I will be able to select my life aspects. - If I select life aspects, then I will be redirected to a set of questions regarding the aspects I have selected. - If I answer all questions on the assessment, then I can click on “Done”
6. As a user, I want to view my wheel of life, so that it is easier for me to comprehend my current situation.	2	Feature	As a registered user, - If I completed the wheel of life assessment questions and click on “Done” button, then I will be redirected to my wheel of life reflecting my answers.
7. As a user, I want to add goals, so that I can have a clear list of what my goals are.	2	Feature	As a registered user, - If I view my wheel of life, then I can click on “add goal” button. - If I click on “Add Goal”, then the system will display a new tab containing input fields. - If I input the goal details in the field, then the system will save my goal and add it to my list of goals. - If my goal is saved, then the system will show a progress bar. - If I add the new goal and the goal’s required information haven’t been fulfilled, then I should get an error message indicating the failure of the adding process.

PBI	Size	Type (Feature, defect, technical work, knowledge acquisition)	Acceptance Criteria The conditions of satisfaction that must be met for that item to be accepted.
8. As a user, I want to add habits, so that I can maintain a positive lifestyle.	2	Feature	<p>As a registered user,</p> <ul style="list-style-type: none"> – If I view my wheel of life containing imbalanced areas, then I can click on “Add Habit” button. – If I click on “Add Habit” then the system will display a new tab containing input fields. – If I input the habit details in the field, then the system will save my habit and add it to my list of habits. – If add the new habit and the habit’s required information hasn’t been fulfilled, then I should get an error message indicating the failure of the adding process.
9. As a user, I want to view my goals/habits, so that I can be reminded of my list of goals/habits.	2	Feature	<p>As a registered user,</p> <ul style="list-style-type: none"> – If I view the goals/habits tab, then I can view my list of goals and habits.
10. As a user, I want to delete goal/habit, so that I can remove the goal/habit that no longer fulfills my vision.	2	Feature	<p>As a registered user,</p> <ul style="list-style-type: none"> – If I view my list of goals/habits, then I can click on the goal/habit from the list. – If I click on my desired goal/habit, then the system will display a modifiable version of that goal/habit. – If I click the “More” button, then the system will display “Edit/Delete” to my goal/habit. – If I click on the “delete” button, then I should receive a confirmation message whether I want to proceed with the process or not. – If I click on the agree button in the confirmation message, then the goal /habit should disappear, and I should be informed of the success of the process. – If I delete a goal/habit, then the system will alter my progress bar will change as needed.
11. As a user, I want to edit goal/habit, so that I can alter my lifestyle as needed.	2	Feature	<p>As a registered user,</p> <ul style="list-style-type: none"> – If I click the “More” button on my desired goal/habit, then the system will display “Edit/Delete” to my goal/habit. – If I click on “Edit”, then the system will allow me to edit the selected goal/habit.

PBI	Size	Type (Feature, defect, technical work, knowledge acquisition)	Acceptance Criteria The conditions of satisfaction that must be met for that item to be accepted.
			<ul style="list-style-type: none"> - If I save my changes after editing the goal/habit, then the system will update to the latest version of goal/habit. - If I edit a goal/habit, then the system will alter my progress bar will change as needed.
12. As a user, I want to view my progress, so that I can know how far I've gotten and encouraged to keep going.	3	Feature	<p>As a registered user,</p> <ul style="list-style-type: none"> - If I view my list of goals, then I can view the progress bar for each goal.
13. As user I want to view my daily ranked tasks so that I do not get overwhelmed organizing my daily routine.	4	Feature	<p>As a registered user,</p> <ul style="list-style-type: none"> - If I add a task to my goal/habit, then the system will display it in a ranked list based on priority and importance.
14. As a user, I want to check off tasks/habits from my daily tasks list so that I get a sense of progress toward my goals.	3	Feature	<p>As a registered user,</p> <ul style="list-style-type: none"> - If I view my daily ranked tasks, then I should be provided with a check box next to each task/habit. - If I check off a completed task, then the task should go to the bottom of the list.
15. As a user, I want to invite friends to a private community so that I can get encouraged and be more committed to the goal.	4	Feature	<p>As a registered user,</p> <ul style="list-style-type: none"> - If I create a community, then I can click on "Private" button. - If I click on "Invite Friends" button, then I can invite specific friends with shared goals to my community.
16. As a user, I want to accept a private community's invitation, so that I can see friends' progress and motivate each other.	3	Feature	<p>As a registered user,</p> <ul style="list-style-type: none"> - If I received a private community's invitation, then a notification message with the sender username and the community's name will appear. - If I click on the notification message, then a detailed description of the community will appear along with an accept and reject buttons. - If I click on the accept button, then I will join the community and be directed to the community page

PBI	Size	Type (Feature, defect, technical work, knowledge acquisition)	Acceptance Criteria The conditions of satisfaction that must be met for that item to be accepted.
17. As a user, I want to reject a private community's invitation, so that I can control which goal is more suitable for me.	3	Feature	<p>As a registered user,</p> <ul style="list-style-type: none"> - If I received a private community's invitation, then a notification message with the sender username and the community's name will appear. - If I click on the notification message, then a detailed description of the community will appear along with an accept and reject buttons. - If I click on the reject button, then I will not join the community.
18. As a user I want to create public communities, so that I can share my goals with a community that has the same goal.	3	Feature	<p>As a registered user,</p> <ul style="list-style-type: none"> - If I create a community, then I can click on "Public" button. - If I make my community public, then members with the same goal can join my community.
19. As a user, I want to join communities, so that I can be more motivated to achieve my goal.	3	Feature	<p>As a registered user,</p> <ul style="list-style-type: none"> - If I view the communities tab, then I can browse through communities. - If I browse through communities, then I can select a community I want to join. - If I join a community, then it will be added to my communities.
20. As a user, I want to view community members' progress so that I can entice competition and get motivated to achieve my goal.	4	Feature	<p>As a registered user,</p> <ul style="list-style-type: none"> - If I view joined communities, then I can see all the members progress ranked from highest to lowest.
21. As a user, I want to view a list of communities and my joined communities, so that I can easily browse through them and find them.	3	Feature	<p>As a registered user,</p> <ul style="list-style-type: none"> - If I view the communities tab, then I can see a list of all public communities and my communities
22. As a user, I want to be able to filter the list of communities based on certain characteristics to find the most desired community easily	4	Feature	<p>As a registered user,</p> <ul style="list-style-type: none"> - If I view the communities tab, then I can see a list of all public communities and my communities. - If I click on the filter icon, then I can choose what aspects I want in the community.

PBI	Size	Type (Feature, defect, technical work, knowledge acquisition)	Acceptance Criteria The conditions of satisfaction that must be met for that item to be accepted.
23. As a user, I want to journal so that I can express my thoughts or ideas.	3	Feature	As a registered user, <ul style="list-style-type: none"> – If I click on the “Journal” tab, then the system will display all my previous entries. – If I click on “Add New Entry”, then I can write a new entry.
24. As a user, I want to view my journal's entry so that I can reflect and gain more insight about my progress	3	Feature	As a registered user, <ul style="list-style-type: none"> – If I click on the “Journal” tab, then the system will display all my previous entries. – If I click on a previous entry, then I can view the entry.
25. As a user, I want to delete my journal's entry so that I can get rid of entities that I no longer want to view	3	Feature	As a registered user, <ul style="list-style-type: none"> – If I click on the “Journal” tab, then the system will display all my previous entries with a delete button next to them. – If I click on the “delete” button, then I should receive a confirmation message whether I want to proceed with the process or not. – If I click on the agree button in the confirmation message, then entry disappears, and I should be informed of the success of the process.
26. As a user, I want to be able to successfully perform application functions, so that my average number of errors per task ¹ is less than 0.7 ² . (Usability - error)		Feature	As a user, If I interact with the app, then my average number of errors per task is less than 0.7.
27. As a user, I want to be able to learn all application functions in		Feature	As a new user, If I launch the app for the first time, then I will be able to learn all application functions in less than 3 minutes.

¹ The equation used for measuring the number of error per task is as follows: total number of error/ total number of tasks, where the total number of errors is the total number of errors that occur during testing.

² <0.7 is known to be the standard for the error per task result [34]

PBI	Size	Type (Feature, defect, technical work, knowledge acquisition)	Acceptance Criteria The conditions of satisfaction that must be met for that item to be accepted.
less than 3 minutes ³ , so that I do not get irritated and exit the app. (Usability - Learnability)			
28. As a user, I want the system to be available 99.999% ⁴ of the time I try to access it, so that I do not get irritated and find another app to use. (Availability)		Feature	As a user, If I use the application, then I will be able to access system functionalities 99.999% of the time.

³ The results were benchmarked from other applications

⁴ 99.999% also known as the five nines is what most vendors use for their products, it indicates a down time of 6.05 seconds per week. [35]

Chapter 5:

System Design



5 System Design

This section will cover the architectural diagram, class diagram, component level design, data design, and interface design.

5.1 Architectural Diagram

Motazen's system will be implemented for iOS devices, and we will be using Flutter's Dart framework. The Flutter architecture has three development levels. The embedder, flutter engine, and flutter framework are listed from top to bottom. To our advantage, Flutter is flexible enough to allow us to adopt any pattern we have used before. Therefore, considering it is one of the most popular iOS architecture patterns, we will be adopting the MVC pattern, also known as the Model View Controller[31]. It is the default application architecture pattern used in all iOS, MacOS, and watchOS projects. One of the goals of this pattern is to divide all our code to Presentation and Domain Model that serve their own purpose. The user interacts with the controller, which acts a middleman between the other two sections: model and view. The model and view do not interact with each other. The model is responsible for handling all the data logic of the user's request, meaning the model interacts with the database, as well as handles the slice of the data that is relevant at-the-moment, meaning all the data our application needs to run, especially as an in-memory, partial and local representation of the data that lives in our database [36]. It receives and contains the data it needs from our database's file storage to our local memory, Isar. Isar is an open-source, NoSQL database that is made specifically for Flutter. [37]

The process goes as follows: First, the client sends a request to the server. Once the server receives the request, the Controller will respond it, as it is responsible for handling all request flow. To perform the requested operation, the Controller does not directly interact with the data logic, but instead uses the Model to perform these interactions. After the Model sends its response back from the database or local memory to the controller, the Controller interacts with the View to render the data to the user, which only responsible for handling data presentation and dynamical rendering. Just as the Controller requests data from the Model, it can also work in the opposite direction. It can receive UI events from the View, process them, and send data to the Model if necessary (e.g. adding data to the Model from user input) [36]. The main reason of choosing this architecture pattern is because the MVC paradigm allows us to easily modify

the application when needed. MVC pattern gives us a far more ordered, readable, and controllable architecture. Adding new types of views is simplified in the MVC pattern; therefore, changes made to one part of the application will never modify the architecture as a whole. As a result, this will help increase the flexibility and scalability of the application.[32]

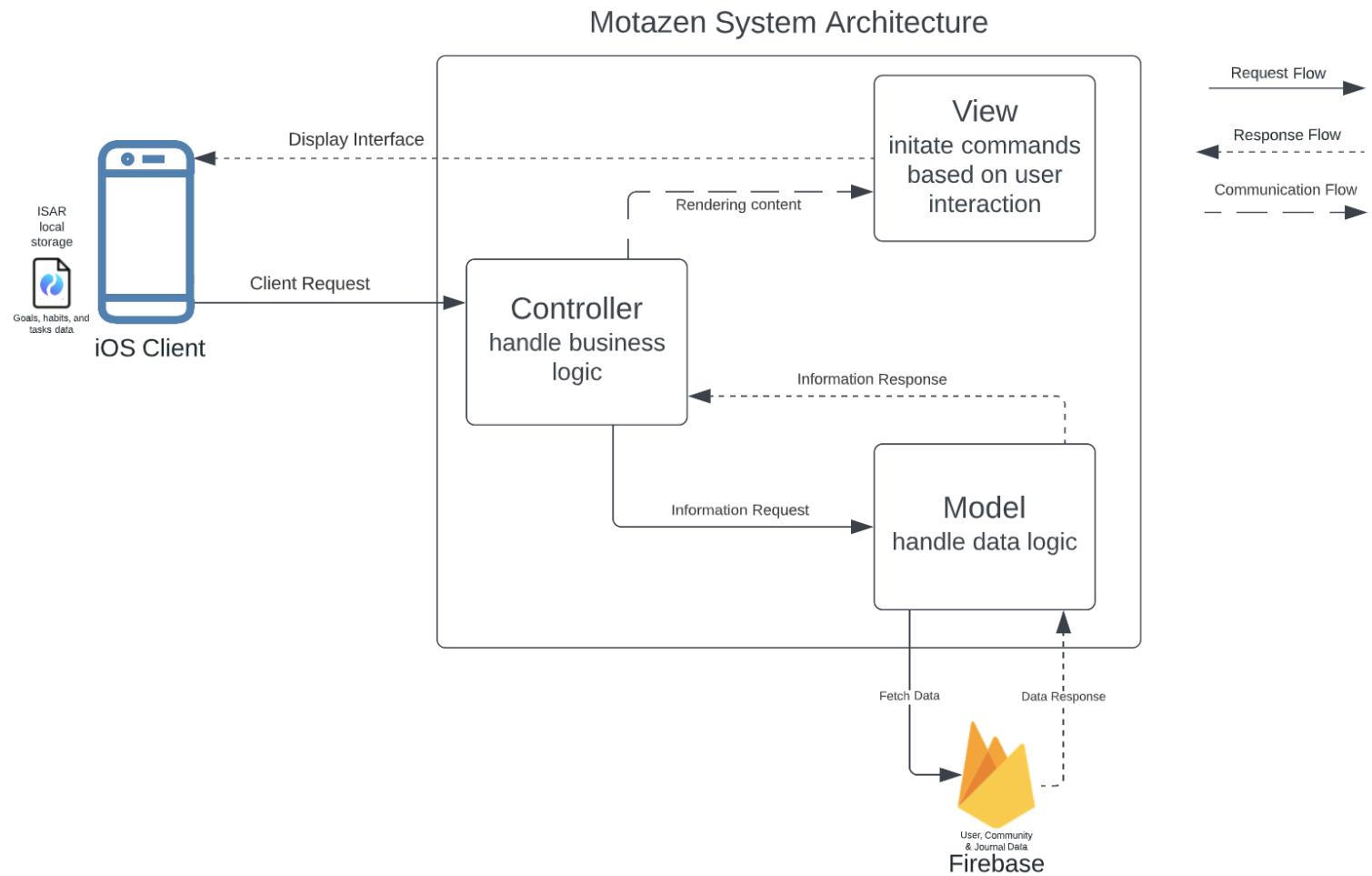


Figure 23 Motazen System Architecture [33]

5.2 Class Diagram

The main purpose here is to gain a general understanding of how the system is decomposed, and how the individual parts work together to provide the desired functionality.

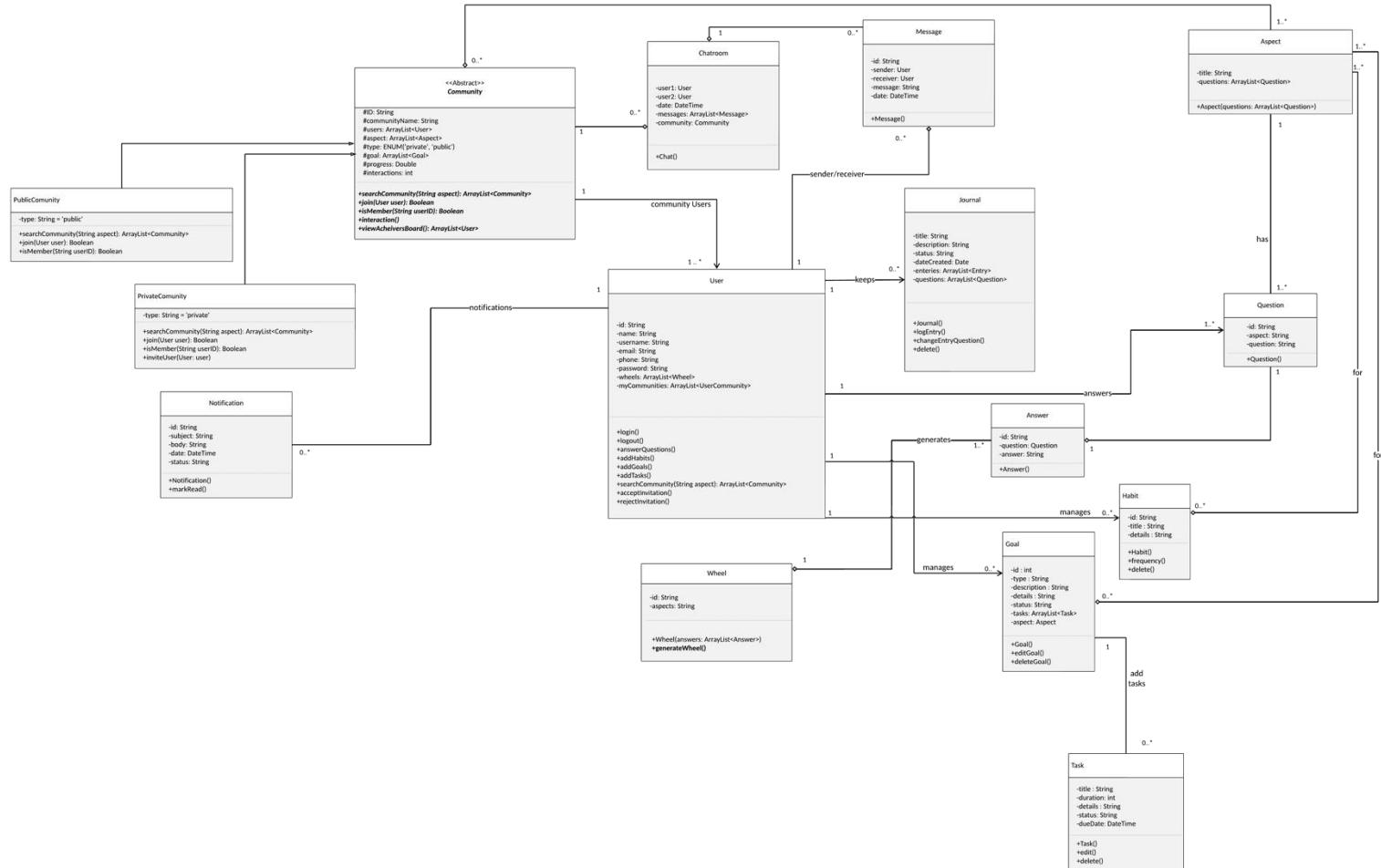


Figure 24 Class Diagram



5.3 Component Level Design

In this section, we will present three diagrams of some of Motazen's key components: Log into the Motazen account, add a goal, and the analysis method of how every user's results are calculated and transformed into a wheel using a specific Wheel of Life school, McLean's Institute of Coaching, Coaches Training Institute.

5.3.1 Generate Wheel of Life Flowchart

Figure 25 is a flowchart representation of the process to generate the user's Wheel of Life. It analyzes the user's answers in the assessment and converts them into visual data in the form of a wheel. The wheel is divided into eight parts, with each part representing an aspect, and further segmented radially using radial displacement (offset) into 10 parts. Each radar represents the balance level of the current aspect. The analytic method was initially created by CTI, Institute of Coaches, a Harvard affiliate medical school, and further researched by Atina Diffley, a public speaker and award-winning author. [38]

Simply put, the analytic method consists of three main steps. The first step is reading the user's answers and storing them in a variable. The answers are then divided by the number of questions in the assessment with respect to each aspect (Each aspect has a different number of questions) multiplied by 10. Then, the result is multiplied by 100 to get a percentage that can be reflected into the radial wheel.

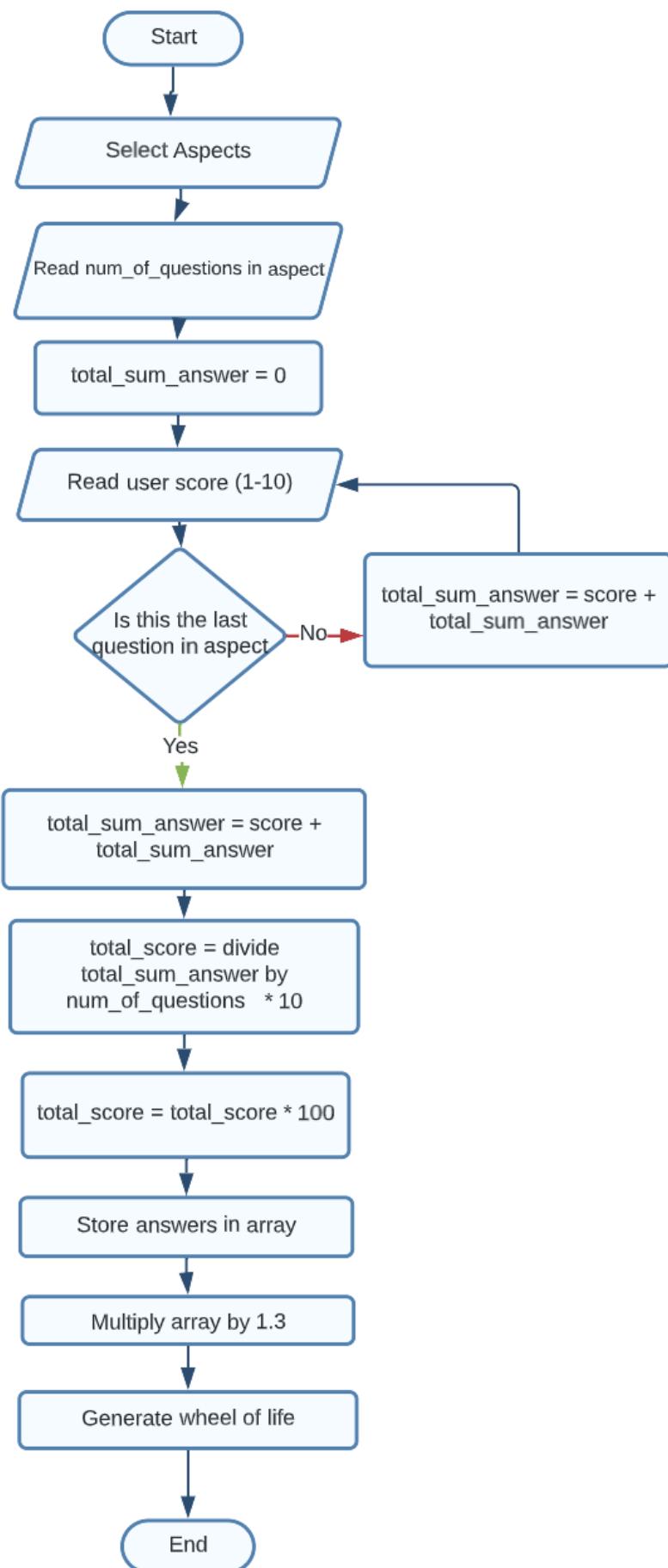


Figure 25 Generate Wheel Flowchart



5.3.2 Add Goals

Figure 26 displays the adding goals process in a flowchart.

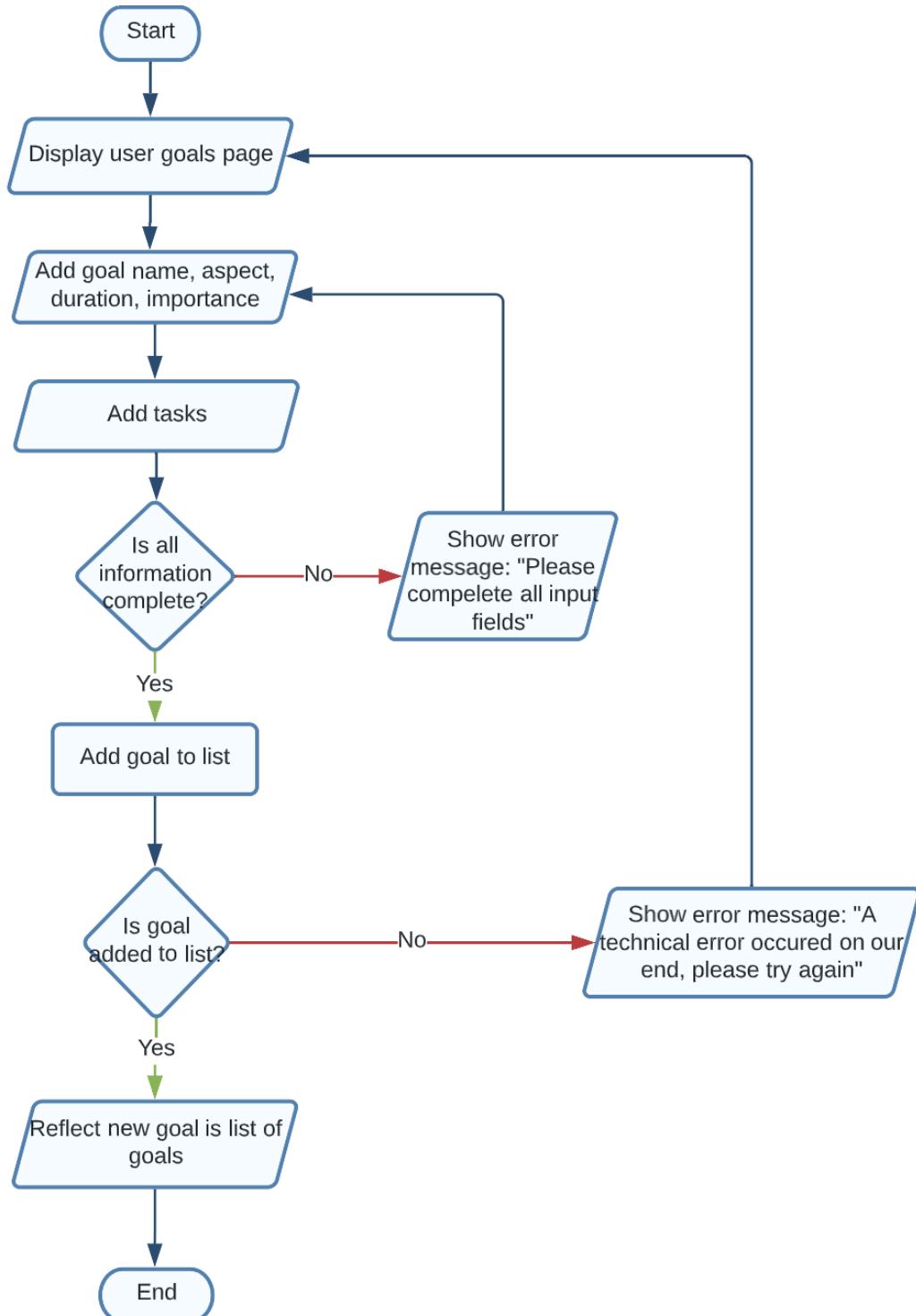


Figure 26 Add Goal Flowchart



5.3.3 Ranking Tasks

The following flowchart was made to provide a visual representation of the decision-making process involved in ranking tasks based on their priority. The algorithm uses the weighted criteria to rank the tasks in order of priority. The tasks that are most important or urgent are ranked higher than tasks that are less important or urgent.

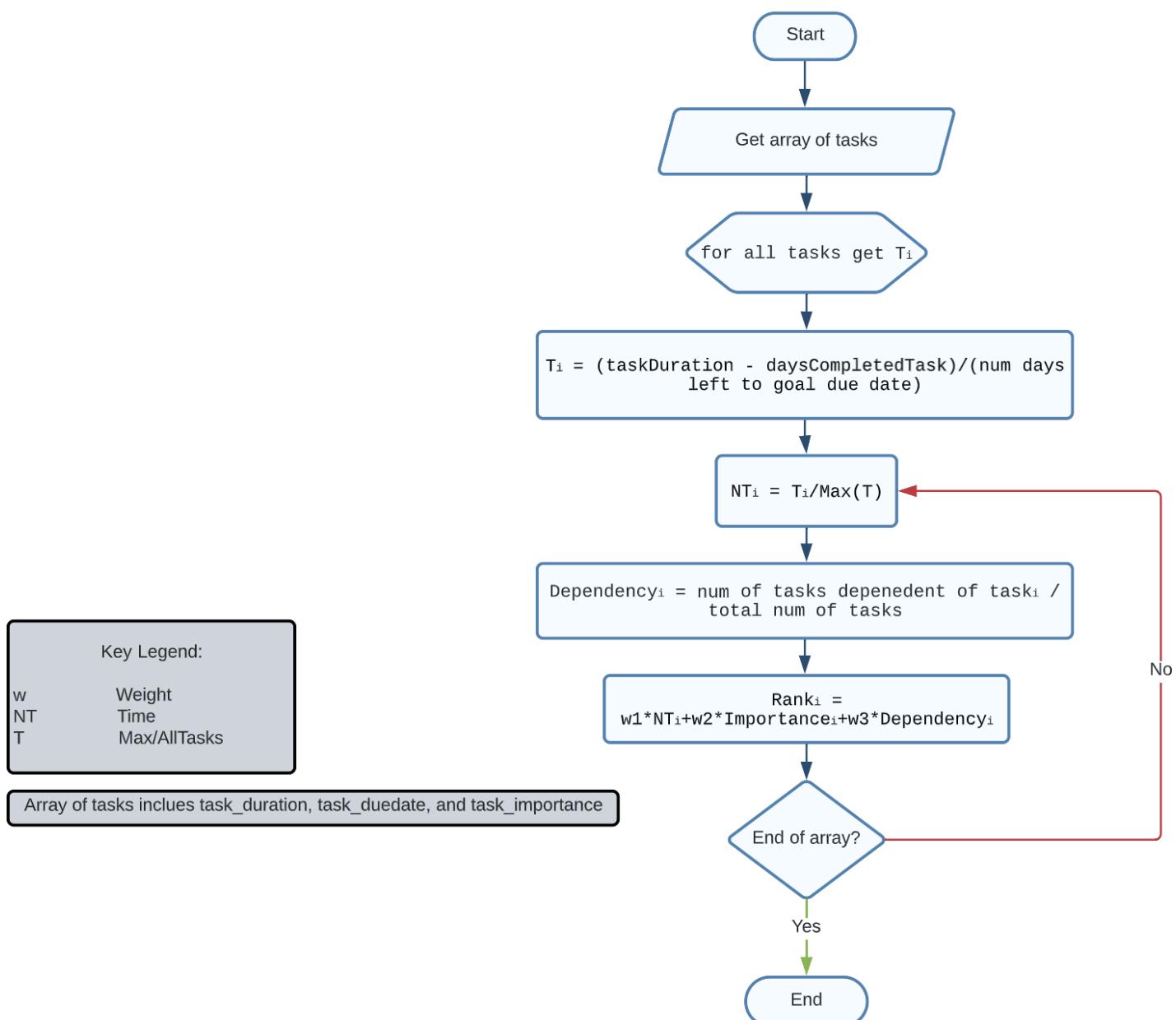


Figure 27 Ranking Tasks



5.4 Data Design

5.4.1 Data Models

In this part, we will show how our system and data are designed and stored, as well as how information is gathered during the system analysis process and is transformed into appropriate data structures for developing our application. We utilized Firebase to store our application.

Firebase has two types of databases: Realtime Database and Cloud Firestore. Both databases are NoSQL databases, so the database is structured as key-value pairs. The reason we have chosen Firebase as the hosting service for our database is because Flutter offers the FlutterFire plugin which gives us the ability to connect our application to Flutter easily & reliably [39]. Moreover, the entire authentication process can be set up quickly with Firebase's user-friendly SDK (Software Development Kit), pre-made UI libraries, and backend services. It also provides a wealth of capabilities that position it as the preferred backend development solution for most websites and mobile applications. It reduces the workload and duration of development, and is a simple, lightweight, recognized tool for prototyping [40].

As for the locally stored data, it is without a doubt that a local database is necessary for mobile app development to maintain and preserve users' data until the application is deleted, or to load data even when the app isn't connected to the Internet. Therefore, we have chosen Isar database to locally store our data since it has been made specially for Flutter. In contrast to other options such as Hive and ObjectBox, Isar is able to load a large amount of data in milliseconds. [41] It is highly scalable, cross-platform, and provides us with a tool to visualize our database.

5.4.1.1 ER Diagram

In figure 28, we illustrated Motazen's ER diagram that displays the relationship of entity sets stored in our Firebase database.

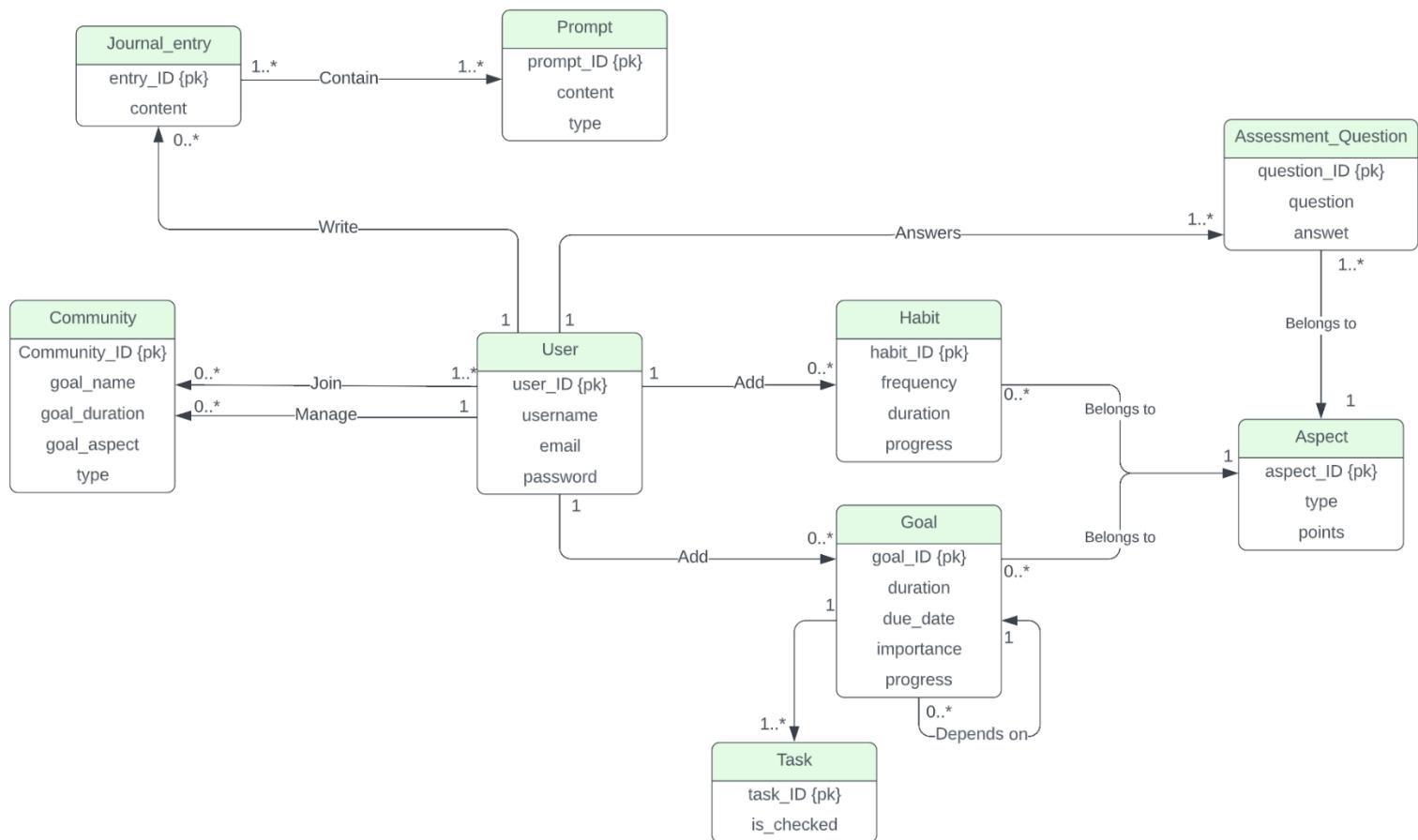


Figure 28 ER Diagram

5.4.1.2 Non-relational data model

Unlike relational data models, NoSQL databases scale much better, since every item in the database stands on its own. This simple modification states that they're essentially key-value stores: meaning each item in the database only has 2 fields, a key and a value; hence why they scale better, and why big organizations such as Apple run NoSQL databases. At Motazen, we opted for a NoSQL database for the same reason, as well as the fact that NoSQL databases are schemaless, meaning items in our database don't need to have the same structure, and thus gives us more flexibility and freedom.

The following figure illustrates our NoSQL Firebase/Firestore database system, a tree-like structure consisting of our collections, documents, and their attributes:

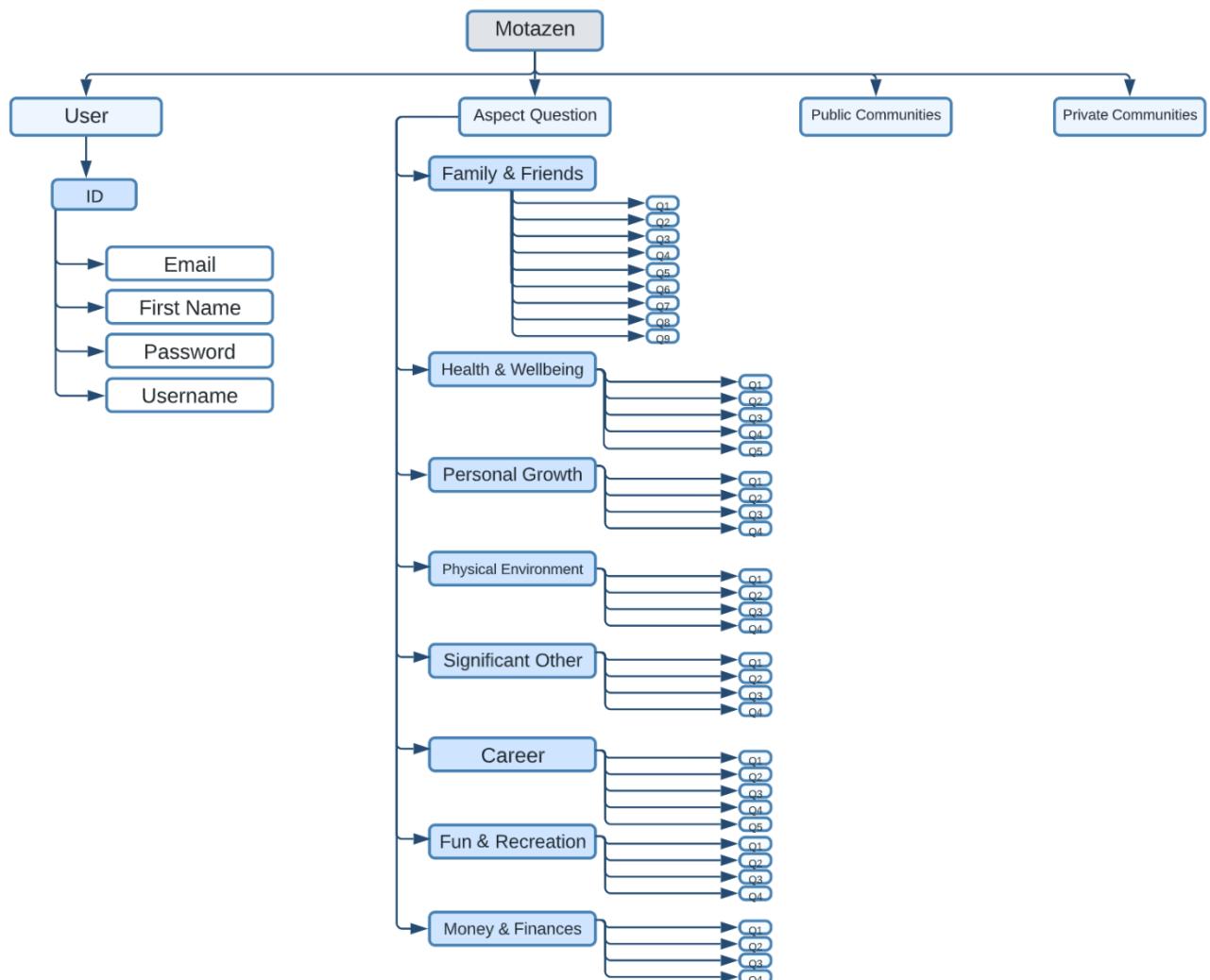


Figure 29 Non-relational data model.



5.5 Interface Design

In the realm of usability engineering, computer scientist Jakob Nielsen proposed 10 principles that have since been adopted as best practices for human-computer interaction, they are currently the most most-used usability heuristics for user experience design. The following are principles we have applied in Motazen.

5.5.1 UX Guidelines used

5.5.1.1 Match between system and the real world

Motazen aims to speak the user's language by not using any unfamiliar jargon or icons. We apply real life metaphors in a technical sense, i.e.: Using a bin to delete, using a pencil to edit, etc. Moreover, it is also important that components be arranged logically so that they make sense to users based on their prior knowledge and life experiences. For example, when adding goals: the user begins by naming the goal then adding goal details later [\[42\]](#).

5.5.1.2 Consistency and Standards

Especially with welcoming a new concept, users should not keep guessing whether two terms mean the same thing. Therefore, we made sure to unify all our terms, i.e.: Between "aspect" and "life area", we've chosen to stay consistent and use "aspect" [\[42\]](#).

5.5.1.3 Error Prevention

A thoughtful design that averts an issue in the first place is preferable to good error messages. We attempt to eliminate error-prone situations or check for them and provide users the chance to affirm before committing to an activity [\[42\]](#).

5.5.1.4 Aesthetic and minimalistic design

As designers and developers, we must live up to a certain standard of both functionality and design, but once the design impacts the function, we immediately swap it out for a simpler,



more minimalistic design. This helps us create interactions that only include necessary information, while keeping away from extraneous visuals that could overwhelm and distract users. In Motazen, we used icons and colors to represent the aspects to give users an overall better experience [\[42\]](#).

5.5.1.5 Help users recognize, diagnose, and recover from errors

To aid our users, Motazen's error messages are written in plain language, completely code free. Generic error messages that do not entail what the issue is usually cause frustration, which is why we aimed to specify the type of error. Telling the user that their login credentials are incorrect, for example, would be a better error message than simply stating a login failure [\[42\]](#).

5.5.1.6 Help and documentation

Motazen gives first time users contextual help by first explaining the Wheel of Life method and what to expect in the application. This will help users better understand the goal of the application and understand how to use its functions [\[42\]](#).

5.5.2 Site Map

Figure 30 showcases Motazen's site map

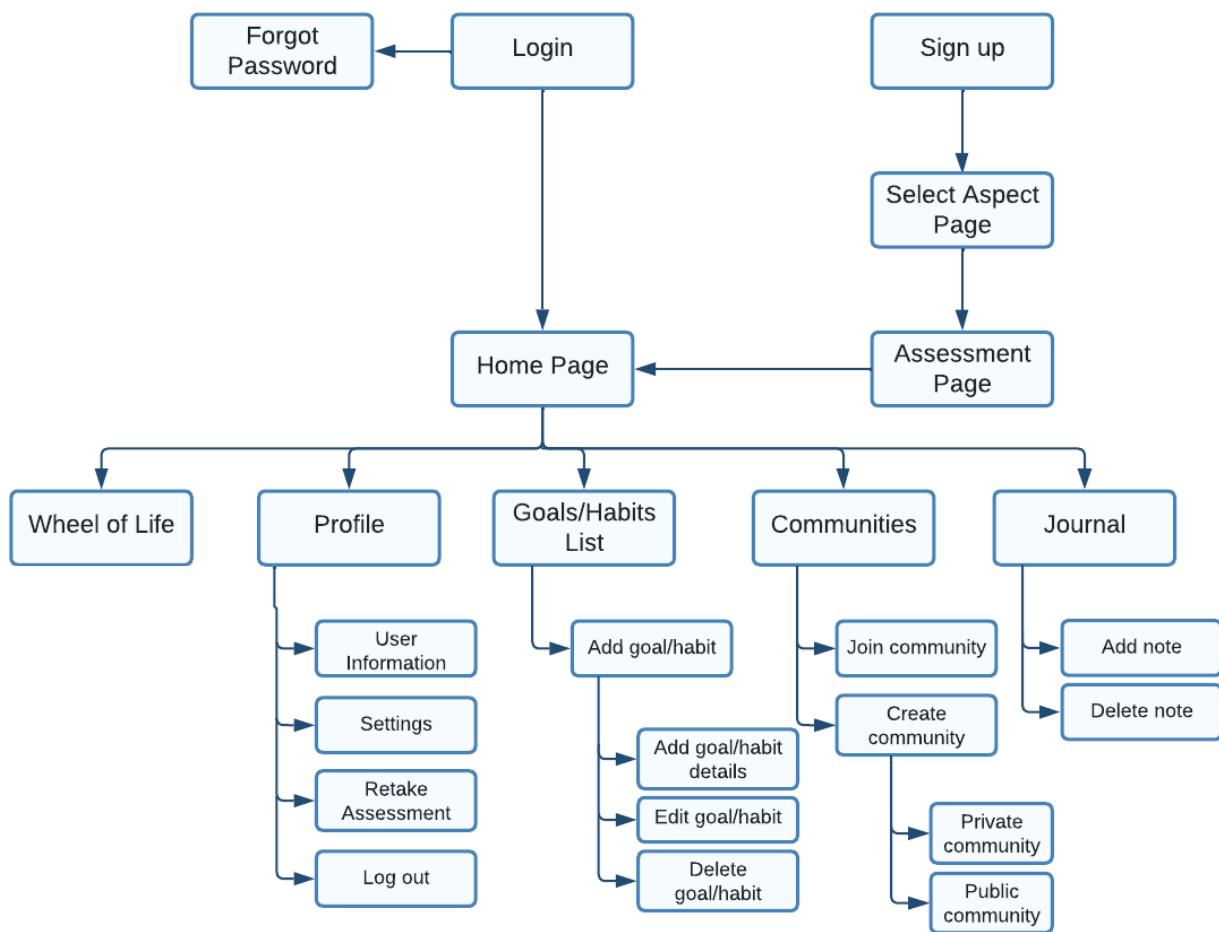


Figure 30 Site Map.

Chapter 6:

System

Implementation



6 System Implementation

6.1 Initialization

In this section, we will explain how we set up Flutter and its Dart plugin to the team's devices.

As part of preparation to begin our development process, the team began by initially installing Flutter SDK depending on each of our operating systems, then set up an editor to run and implement Flutter onto. Our chosen environments depended on the OS used, Android Studio for Windows and Visual Studio Code for MacOS. Once we started our preferred IDEs, we invoked the preferences/command palette to install all necessary extensions and plugins, which included the Dart plugin. Next, we tested our Flutter setup out by running the `flutter doctor` command. Later, we wanted to experiment with our first demo app using the simulator/emulator, so we ran the Flutter Demo Home Page by using the `open -a flutter` command, which automatically chooses the latest device available. Once we could see the simulator/emulator, we debugged and ran the app to check for any errors, and as soon as it ran as expected we were set to begin implementing Motazen.

6.2 Connecting to the database

In this section, we will discuss how we connected our Flutter project to the Firebase for an external database, and how we used Isar to store offline data, more commonly known as local storage.

Before connecting Motazen to Firebase, we had to first create a Google account that would be then used to create a Firebase account. Next, we logged into our Firebase account installed the FlutterFire CLI by running `dart pub global activate flutterfire_cli`. FlutterFire is a set of Flutter plugins which will connect our Flutter application to Firebase. Then, in the Firebase page under “Docs”, we chose the “Get started for Android” option. This does not impact whether our application will be implemented for iOS devices or Android devices, as it is only used to



redirect us to the launch page. Then we went to the “Use Firebase with a Framework” in which Flutter is the only option.

Once we chose our preferred OS, we had to create a demo Firebase project to connect our application to. After that, we need to register Motazen in the Firebase console to connect them together. To do that, we had to access the AndroidManifest.xml file in our project and copy the full package name, which could either be created automatically or written manually. We then feed it to Firebase by pasting the package name in the correct input field and give our application any chosen nickname. After clicking “Register App”, we must now download the configuration file, which is done by first modifying our root level Gradle file. We then used flutterfire configure to start the app configuration workflow. To initialize Firebase in Motazen, we had to install the core plugins, and run them with each step to ensure that the configuration is up to date. In our main class, we had to import the configuration file created earlier since the main class would be the link to all other classes. As a final step, we added all the necessary plugins including firebase_auth and firebase_database.

6.3 Core Functions

6.3.1 Ranked Tasks

Motazen uses a weighted sum MCDM, where each task is assigned a number (rank) in the range [0,1] and larger numbers indicate a higher rank/ more important task. To do so, we consider three factors: importance, time, dependency.

As for the weights, they must be in the range [0,1] and their sum must be equal to 1. The default value for all weights is 0.33, as we consider that all terms in the equation are equally important, however, users have the option to edit the weights according to their needs.

Each factor has a term in the equation. Each term must have a number in the range [0,1], as follows:

- **Importance:**

Where the higher the importance, the higher the rank. This value is set and saved in the local database when the user creates a goal, it has three possible values:

- The goal importance is high = 0.75
- The goal importance is medium = 0.5
- The goal importance is low = 0.25

- **Time:**

Where the lesser days left, the higher the rank. To achieve that, we will perform the following equation on each task where T_i is time:

$$T_i = \frac{\text{TaskDuration} - \text{DaysCompleted_Task}}{\text{num days left to goal due date}} \quad \text{Equation 1}$$

Next, we need to normalize the time value for each task by dividing T_i of each task over the maximum T of all tasks, the normalized time NT_i for a task is calculated as follows:

$$NT_i = \frac{T_i}{\max T} \quad \text{Equation 2}$$

```
//Calculate NT(the normalized time criteria value) for each task
void calculateNT(List<Item> tasks) {
    double timeLeft;
    double maxTime;      You, 2 months ago • added first ver of ranking ...
    double diff;
    if (tasks.isEmpty) {
        return;
    }
    for (var task in tasks) {
        //find how many days are left till the due date
        if (task.dueDate!.difference(DateTime.now()).inDays.toDouble() == 0) {
            diff = task.dueDate!.difference(DateTime.now()).inMinutes.toDouble()
                / 24 /
                60;
        } else {
            diff = task.dueDate!.difference(DateTime.now()).inDays.toDouble();
        }

        //Calculate the time left for each task
        timeLeft = (task.duration! - task.daysCompletedTask!) / diff;
        //save the time left
        task.timeLeft = timeLeft;
    }

    //Find the task with the most time left
    Item taskWithMaxT = tasks.reduce(
        (item1, item2) => item1.timeLeft > item2.timeLeft ? item1 : item2);
    //save the maximum time left
    maxTime = taskWithMaxT.timeLeft;
    //Next we need to normalize the data
    for (var task in tasks) {
        task.nt = task.timeLeft / maxTime;
    }
}
```

Figure 31 The method for calculating normalized time



As shown in figure (31), we implemented the time term calculations in a method called `calculateNT`, we first started by declaring `timeLeft`, `maxTime`, and `diff`, we also added a conditional statement that checks if the task list is empty, if it is we will return from the method, if not we will loop over each task to find the difference between the task's due date and the current date in days by using the build in method `difference`⁵.

Next, we'll calculate `timeLeft` using equation (1) and then set the value for the task. After the loop is complete, we will use the `reduce` method to compare the `timeLeft` value for each task and return the task with the maximum `timeLeft`. We will store the maximum time value in `maxTime`.

Finally, we will loop over each task and divide its `timeLeft` value over `maxTime` and set the result as the task's `NT`.

- **Dependency:**

Where the more tasks dependent on the task, the higher the rank. To achieve that, we will perform the following equation on each task where $Depedency_i$ is the task's dependency:

$$Depedency_i = \frac{\text{num of tasks dependent on task } i}{\text{total num of tasks}} \quad \text{Equation 3}$$

The first step for applying the equation is to find the *num of all tasks dependent on task i* directly or indirectly, this can be calculated by creating a directed acyclic graph (DAG)⁶ where each node represents a task and an edge is created between the task and its dependent tasks, the number of incoming edges for a given task is *num of all tasks dependent on task i*.

⁵ When the time difference is less than a day, the automatic conversion method `inDays` always returns zero, hence a conditional statement is added to check if the value is equal to zero, and the difference is automatically calculated in minutes, then manually calculated in days.

⁶ The code does not allow the task to add itself as a dependency, in turn making the graph acyclic

```

static DirectedGraph<String> graph = DirectedGraph<{}>;
void createDepGraph(LocalTask task) {
    ///check if the graph contains the task, if not
    ///add the task to the graph
    if (!graph.contains('${task.id}')) {
        graph.addEdge('${task.id}', {});
    }
    You, 3 months ago • latest ver. ...
    //add the dependencies of the task
    for (var element in task.taskDependency.toList()) {
        if (!graph.edgeExists('${task.id}', '${element.id}')) {
            graph.addEdge('${task.id}', '${element.id}');
        }
    }
}
  
```

Figure 32 the method for creating a directed graph

Figure (32) shows the implementation of the graph, the graph itself is a static variable, the graph is initialized inside the `createDepGraph` method. The method receives a task as a parameter and then checks if the graph contains the task node exists in the graph, if not, it adds the node to the graph. Next, we loop over each task which it is dependent on and create an edge between them.

```

void calculateDependency(List<Item> tasks) {
    for (var task in tasks) {
        double dependency =
            graph.inDegree('${task.id}')!.toDouble() / tasks.length;
        task.depandancies = dependency;
    }
}
  
```

Figure 33 the method for calculating the dependency

In figure (33), we can see the `calculateDependency` method that receives a list of type `Item` and then loops over each item applying equation (3) to find the task's dependency, and then assigning that value to the task.

Finally, we can now calculate the rank for each task, using the following equation:

$$Rank_i = w_1 * NT_i + w_2 * importance_i + w_3 * Dependency_i \quad \text{Equation 4}$$

```

List<Item> calculateRank(List<Item> tasks) {
    const double weight = 0.33;
    final now = DateTime.now();
    //we need to calculate the NT first
    calculateNT(tasks);

    //calculate dependency
    calculateDependency(tasks);

    //calculate the rank
    for (var task in tasks) {
        //check items should be displayed at the bottom of the list
        if (task.completed) {
            task.rank = 0;
            continue;
        }
        //items past due should be displayed at the top of the list
        if (task.dueDate!.isBefore(now)) {
            task.rank = 1;
        } else {
            task.rank = (weight * task.importance!) +
                (weight * task.nt) +
                (weight * task.depandancies);
        }
    }

    graph.clear();      You, 2 months ago • latest changes ...
    //sort list
    return tasks.sorted(
        [SortedComparable<Item, double>((task) => task.rank!, invert: true)]);
}

```

Figure 34 the calculate rank method

As seen in Figure (34), the calculate rank method receives and returns a list of tasks of type `Item`, which is the type defined for items to be displayed on the list. Then the method defines the constant `weight`, and a final variable of type `DateTime now` that contains the current Date and time.

After initializing `weight` and `now`, we'll call the `calculateNT` and `calculateDependency` methods respectively to calculate and set the NT and Dependency factors.

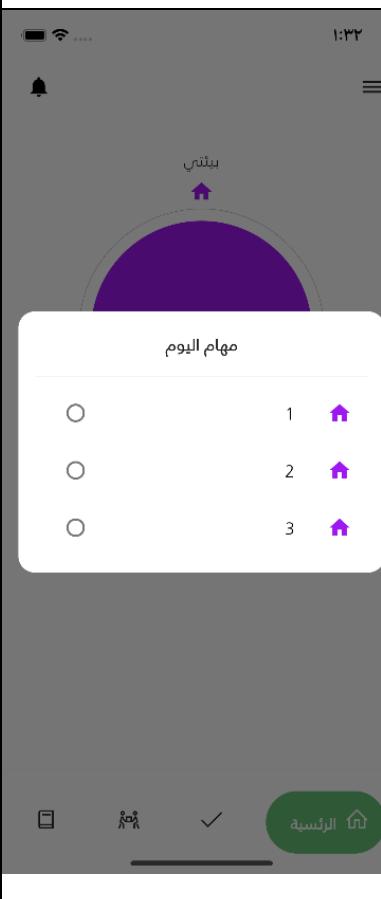


Next, we check if the task has been completed for the day (checked) if it has, then its rank is set to zero so that it would be displayed at the bottom of the list, if not then we will check if the task has passed its due date by checking if the task's due date is before the current date using the variable **now**, if it is past its due date then its rank is set to one so that it would be displayed at the top of the list. After that we will calculate the rank of the task using equation (4) and assign the value to the task. Finally, we'll clear the graph and return the list of items sorted by rank⁷.

6.3.1.1 Ranking Algorithm Test Cases (Experimental Results)

Test case	Variable values	Expected result	Actual Result	Pass?	Comments
1. Adding an item with high Importance, and all other list items are of low importance	<p>Importance: Item1 = 0.75 Items2&3 = 0.25</p> <p>Due date: Items 1&2&3 = due four days from current date</p> <p>Duration: Items 1&2&3 = 4</p> <p>daysCompleted: Items 1&2&3 = 0</p> <p>Dependency : Items 1&2&3 = 0</p>	$\text{RankItem1} = 0.33 * 1 + 0.33 * 0.75 + 0.33 * 0 = 0.5775$ $\text{RankItems2\&3} = 0.33 * 1 + 0.33 * 0.25 + 0.33 * 0 = 0.4125$ <p>Thus, the order of the tasks should be: Item1 Item2 Item3</p>		pass	Note: items with equal rank are displayed by the order of creation
2. Adding an item with high	Importance: Item1 = 0.75	<ul style="list-style-type: none"> $\text{RankItem1} = 0.33 * 1 + 0.33 * 0.75 + 0.33 * 0 = 0.5775$ 		pass	No comments

⁷ The sorted comparable method, by default, sorts items in ascending order, hence it was inverted to return the list in descending order.

Importance, and all other list items are of medium importance.	<p>Items2&3 = 0.5</p> <p>Due date: Items 1&2&3 = due four days from current date</p> <p>Duration: Items 1&2&3 = 4</p> <p>daysCompleted: Items 1&2&3 = 0</p> <p>Dependency : Items 1&2&3 = 0</p>	$= \mathbf{0.5775}$ <ul style="list-style-type: none"> Rank_{Items2&3} = $0.33 * 1 + 0.33 * 0.5 + 0.33 * 0 = 0.4950$ <p>Thus, the order of the tasks should be: Item1 Item2 Item3</p>		
3. Adding an item with high Importance, and all other list items are of high importance	<p>Importance: Items1&2&3 = 0.75</p> <p>Due date: Items 1&2&3 = due four days from current date</p> <p>Duration: Items 1&2&3 = 4</p> <p>daysCompleted: Items 1&2&3 = 0</p> <p>Dependency : Items 1&2&3 = 0</p>	<ul style="list-style-type: none"> Rank_{Items1&2&3} = $0.33 * 0.75 + 0.33 * 0 = 0.5775$ <p>Thus, the order of the tasks should be: Item1 Item2 Item3</p>		pass No comment

4. Adding an item with high Importance, and other list items have either medium and low importance	Importance: Item1 = 0.75 Item2 = 0.25 Item3 = 0.5 Due date: Items 1& 2&3 = due four days from current date Duration: Items 1& 2&3 = 4 daysCompleted: Items 1& 2&3 = 0 Dependency: : Items 1& 2&3 = 0	<ul style="list-style-type: none"> • $Rank_{Item1}=0.33*1+0.33*0.75+0.33*0 = \mathbf{0.5775}$ • $Rank_{Item2}=0.33*1+0.33*0.25+0.33*0 = \mathbf{0.4125}$ • $Rank_{Item3}=0.33*1+0.33*0.5+0.33*0 = \mathbf{0.4950}$ <p>Thus, the order of the tasks should be:</p> <p>Item1 Item3 Item2</p>		pass	No comments
5. Adding an item with high Importance, and all other list items have either high, medium, and low importance	Importance: Item1&4 = 0.75 Item2 = 0.25 Item3 = 0.5 Due date: Items 1& 2&3&4 = due four days from current date Duration: Items 1& 2&3&4 = 4 daysCompleted: Items 1& 2&3&4 = 0 Dependency: : Items 1& 2&3&4 = 0	<ul style="list-style-type: none"> • $Rank_{Items1&4}=0.33*1+0.33*0.75+0.33*0 = \mathbf{0.5775}$ • $Rank_{Item2}=0.33*1+0.33*0.25+0.33*0 = \mathbf{0.4125}$ • $Rank_{Item3}=0.33*1+0.33*0.5+0.33*0 = \mathbf{0.4950}$ <p>Thus, the order of the tasks should be:</p> <p>Item1 Item4 Item3 Item2</p>		pass	No comments
6. Adding an item that is due in 1 day, and all other	Importance: Items1&2&3 = 0.5 Due date:	<ul style="list-style-type: none"> • $Rank_{Item1}=0.33*1+0.33*0.5+0.33*0 = \mathbf{0.5775}$ 		pass	No comments

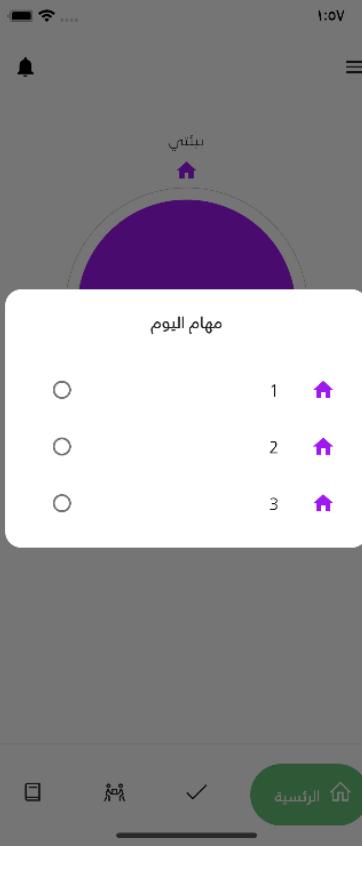
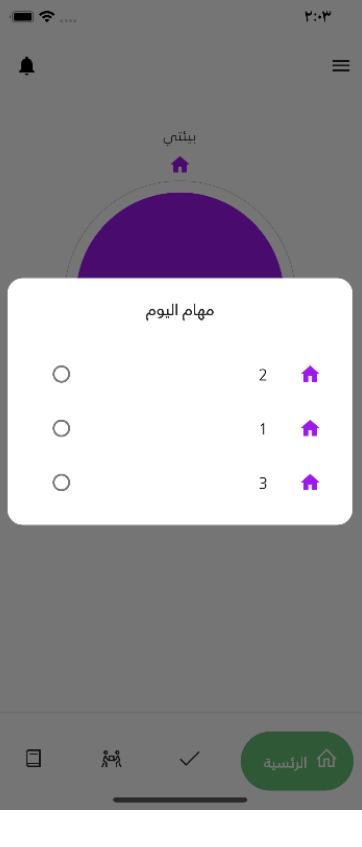
list items are due in 4 days	Item 1 = due one day from current date Items 2&3 = due four days from current date Duration: Items 1&2&3 = 1 daysCompleted: Items 1&2&3 = 0 Dependency : Items 1&2&3 = 0	$\begin{aligned} &= \mathbf{0.4950} \\ &\bullet \text{ Rank}_{\text{Items2\&3}} = 0.33 * 0.25 + 0.33 * 0.5 + 0.33 * 0 \\ &= \mathbf{0.2475} \end{aligned}$ <p>Thus, the order of the tasks should be:</p> <p>Item1 Item2 Item3</p>			
7. Adding an item that is due in 1 day, and all other list items are due in 1 day	Importance: Items 1&2&3 = 0.5 Due date: Items 1&2&3 = due one day from current date Duration: Items 1&2&3 = 1 daysCompleted: Items 1&2&3 = 0 Dependency : Items 1&2&3 = 0	$\bullet \text{ Rank}_{\text{Items1\&2\&3}} = 0.33 * 1 + 0.33 * 0.5 + 0.33 * 0 \\ = \mathbf{0.4950}$ <p>Thus, the order of the tasks should be:</p> <p>Item1 Item2 Item3</p>		pass	No comments
	Importance:				No comments

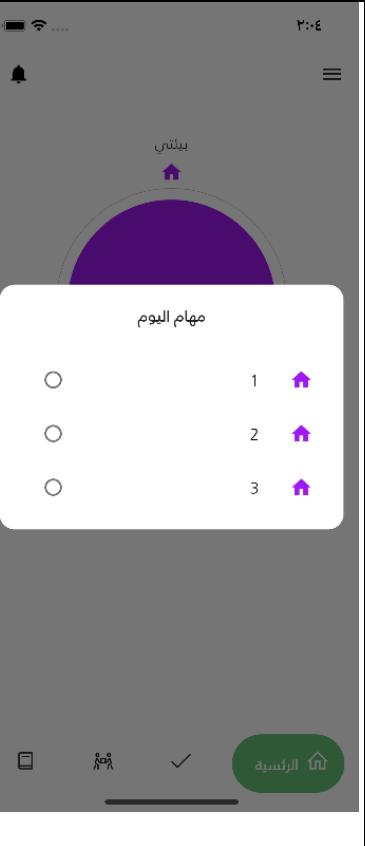
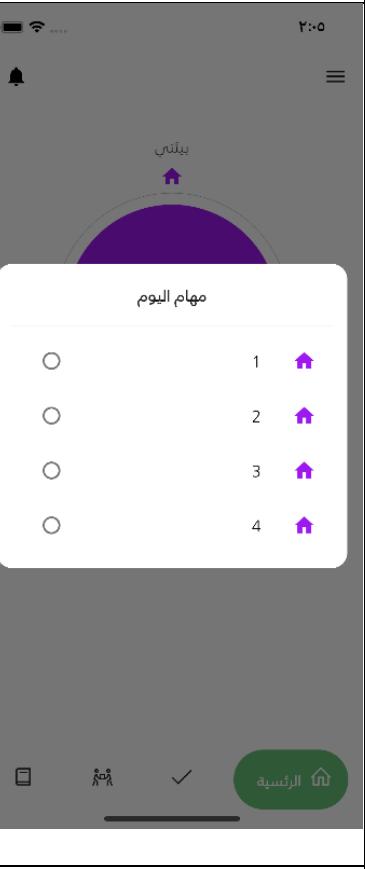
8. Adding an item that is due in 4 days, and all other list items are due in 1 day	Items 1&2&3 = 0.5 Due date: Item 1 = due four days from current date Items 2&3 = due one day from current date Duration: Items 1&2&3 = 1 daysCompleted: Items 1&2&3 = 0 Dependency : Items 1&2&3 = 0	<ul style="list-style-type: none"> $Rank_{Item1} = 0.33 * 0.25 + 0.33 * 0.5 + 0.33 * 0 = 0.2475$ $Rank_{Items2&3} = 0.33 * 1 + 0.33 * 0.5 + 0.33 * 0 \quad \text{????}$ $\diamondsuit Items2&3 = 0.33 * 1 + 0.33 * 0.5 + 0.33 * 0 = 0.4950$ <p>Thus, the order of the tasks should be: Item2 Item3 Item1</p>		pass
9. Adding an item that is due in 1 day, an item that is due in 2 days, and an item that is due in 4 days	Importance: Items 1&2&3 = 0.5 Due date: Item 1 = due one day from current date Item 2 = due two days from current date Item 3 = due four days from current date Duration: Items 1&2&3 = 1 daysCompleted: Items 1&2&3 = 0 Dependency : Items 1&2&3 = 0	<ul style="list-style-type: none"> $Rank_{Item1} = 0.33 * 1 + 0.33 * 0.5 + 0.33 * 0 = 0.4950$ $Rank_{Item2} = 0.33 * 0.5 + 0.33 * 0.5 + 0.33 * 0 = 0.3300$ $Rank_{Item3} = 0.33 * 0.25 + 0.33 * 0.5 + 0.33 * 0 = 0.2475$ <p>Thus, the order of the tasks should be: Item1 Item2 Item3</p>		pass No comments

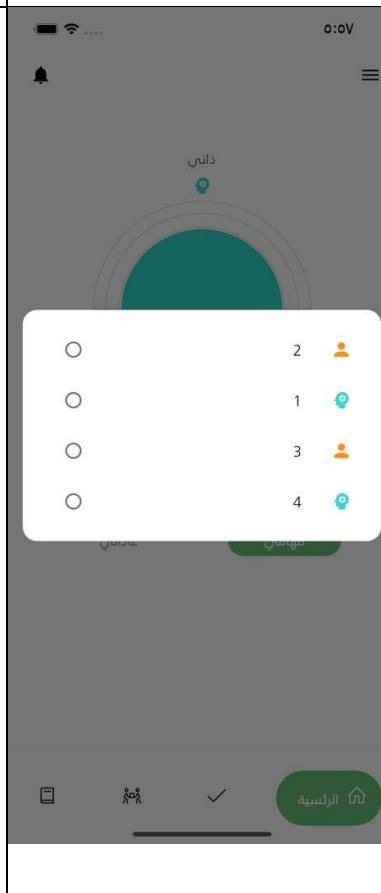
10. Adding an item with 1 day duration, and all other items have 4-day duration	Importance: Items 1&2&3 = 0.5 Due date: Item 1&2&3 = due four days from current date Duration: Item 1 = 1 Items 2&3 = 4 daysCompleted: Items 1&2&3 = 0 Dependency: : Items 1&2&3 = 0	<ul style="list-style-type: none"> $Rank_{Item1} = 0.33 * 0.25 + 0.33 * 0.5 + 0.33 * 0 = 0.2475$ $Rank_{Items2&3} = 0.33 * 1 + 0.33 * 0.5 + 0.33 * 0 = 0.4950$ <p>Thus, the order of the tasks should be:</p> <p>Item2 Item3 Item1</p>		pass	No comments
11. Adding an item with 4-day duration, and all other items have 4-day duration	Importance: Items 1&2&3 = 0.5 Due date: Item 1&2&3 = due four days from current date Duration: Items 1&2&3 = 4 daysCompleted: Items 1&2&3 = 0 Dependency: : Items 1&2&3 = 0	<ul style="list-style-type: none"> $Rank_{Item1&2&3} = 0.33 * 1 + 0.33 * 0.5 + 0.33 * 0 = 0.4950$ <p>Thus, the order of the tasks should be:</p> <p>Item1 Item2 Item3</p>		pass	No comments
12. Adding an item with 4-day duration	Importance: Items 1&2&3 = 0.5	<ul style="list-style-type: none"> $Rank_{Item1} = 0.33 * 1 + 0.33 * 0.5 + 0.33 * 0$ 		pass	No comments

duration, and all other items have 1-day duration	Due date: Item 1&2&3 = due four days from current date Duration: Item 1 = 4 Items 2&3 = 1 daysCompleted: Items 1&2&3 = 0 Dependency: : Items 1&2&3 = 0	$\begin{aligned} em1 &= 0.33 * 1 + 0.33 * 0.5 + 0.33 * 0 \\ &= \mathbf{0.4950} \\ \bullet \quad Rank_{Items2\&3} &= 0.33 * 0.25 + 0.33 * 0.5 + 0.33 * 0 \\ &= \mathbf{0.2475} \end{aligned}$ <p>Thus, the order of the tasks should be:</p> <p>Item1 Item2 Item3</p>		
13.Completing (checking) an item, and all other items are incomplete (unchecked)	Importance: Items 1&2&3 = 0.5 Due date: Item 1&2&3 = due four days from current date Duration: Items 1&2&3 = 4 daysCompleted: Item 1 = 1 Items 2&3 = 0 Dependency: : Items 1&2&3 = 0	$\begin{aligned} \bullet \quad Rank_{Item1} &= 0.33 * 0.75 + 0.33 * 0.5 + 0.33 * 0 \quad \text{◆◆◆} \\ \text{◆◆} \quad Item1 &= 0.33 * 0.75 + 0.33 * 0.5 + 0.33 * 0 \\ &= \mathbf{0.4125} \\ \bullet \quad Rank_{Items2\&3} &= 0.33 * 1 + 0.33 * 0.5 + 0.33 * 0 \\ &= \mathbf{0.4950} \end{aligned}$ <p>Thus, the order of the tasks should be:</p> <p>be: Item2 Item3 Item1</p>		<p>Note: For enhancing the usability and user experience, completed items are moved to the bottom of the list.</p> <p>pass</p>
	Importance:			

14. Uncheck an item, and all other items are checked	<p>Items 1&2&3 = 0.5</p> <p>Due date: Item 1&2&3 = due four days from current date</p> <p>Duration: Items 1&2&3 = 4</p> <p>daysCompleted: Items 1&2 = 1 Item 3 = 0</p> <p>Dependency : Items 1&2&3 = 0</p>	<ul style="list-style-type: none"> Rank_{Items1&2}=$0.33*0.5+0.33*0.5+0.33*0=0.4125$ Rank_{Item3}=$0.33*1+0.33*0.5+0.33*0=0.4950$ <p>Thus, the order of the tasks should be: Item3 Item1 Item2</p>	 <p>مهام اليوم</p> <table border="1"> <thead> <tr> <th>رتبة</th> <th>المهمة</th> </tr> </thead> <tbody> <tr> <td>3</td> <td>Item3</td> </tr> <tr> <td>4</td> <td>Item1</td> </tr> <tr> <td>2</td> <td>Item2</td> </tr> </tbody> </table>	رتبة	المهمة	3	Item3	4	Item1	2	Item2	pass
رتبة	المهمة											
3	Item3											
4	Item1											
2	Item2											
15. Checking an item, and all other items are checked	<p>Importance: Items 1&2&3 = 0.5</p> <p>Due date: Item 1&2&3 = due four days from current date</p> <p>Duration: Items 1&2&3 = 4</p> <p>daysCompleted: Items 1&2&3 = 1</p> <p>Dependency : Items 1&2&3 = 0</p>	<ul style="list-style-type: none"> Rank_{Items1&2&3}=$0.33*1+0.33*0.5+0.33*0=0.4950$ <p>Thus, the order of the tasks should be: Item1 Item2 Item3</p>	 <p>مهام اليوم</p> <table border="1"> <thead> <tr> <th>رتبة</th> <th>المهمة</th> </tr> </thead> <tbody> <tr> <td>1</td> <td>Item1</td> </tr> <tr> <td>2</td> <td>Item2</td> </tr> <tr> <td>3</td> <td>Item3</td> </tr> </tbody> </table>	رتبة	المهمة	1	Item1	2	Item2	3	Item3	pass
رتبة	المهمة											
1	Item1											
2	Item2											
3	Item3											

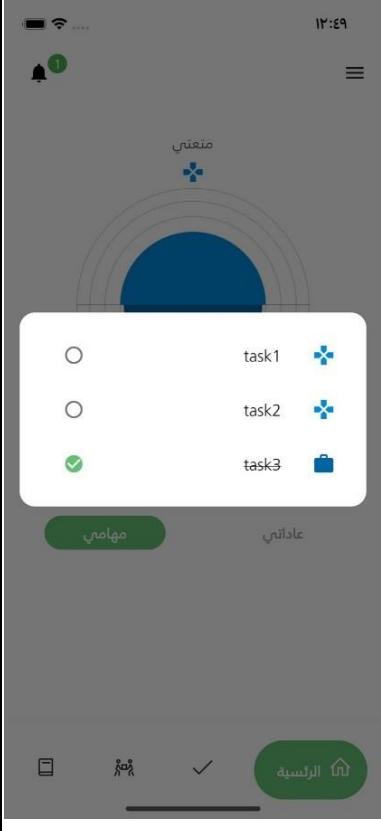
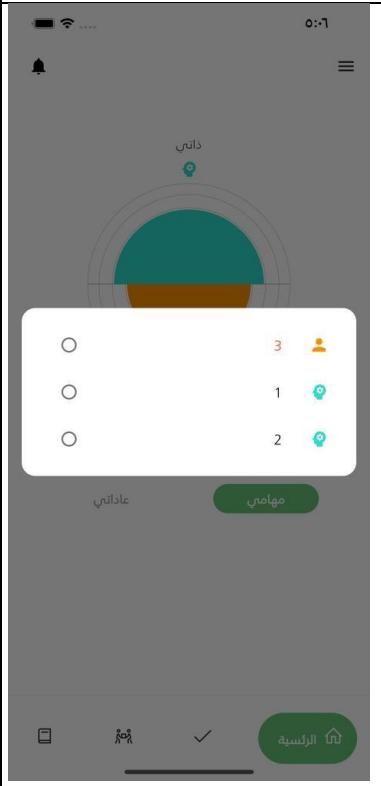
16. Uncheckin g an item, and all other items are unchecked	Importance: Items 1&2& $3 = 0.5$ Due date: Item 1&2&3 = due four days from current date Duration: Items $1 \& 2 \& 3 = 4$ daysComple ted: Items $1 \& 2 \& 3 = 0$ Dependency : Items $1 \& 2 \& 3 = 0$	<ul style="list-style-type: none"> $Rank_{Items1\&2\&3} = 0.33 * 1 + 0.33 * 0.5 + 0.33 * 0 = 0.4950$ <p>Thus, the order of the tasks should be: Item1 Item2 Item3</p>			
17. Adding an item with 1 dependency, and all other items have 0 dependencies	Importance: Items 1&2& $3 = 0.5$ Due date: Item 1&2&3 = due four days from current date Duration: Items $1 \& 2 \& 3 = 4$ daysComple ted: Items $1 \& 2 \& 3 = 0$ Dependency : Items 1&3 = 0 Item 2 = 1	<ul style="list-style-type: none"> $Rank_{Item2} = 0.33 * 1 + 0.33 * 0.5 + 0.33 * 0.33 = 0.605$ $Rank_{Items1\&3} = 0.33 * 1 + 0.33 * 0.5 + 0.33 * 0 = 0.495$ <p>Thus, the order of the tasks should be: Item2 Item1 Item3</p>		Pass	No comment
	Importance:				No comment

18. Adding an item with 0 dependencies, and all other items have 1 dependency	Items 1&2&3 = 0.5 Due date: Item 1&2&3 = due four days from current date Duration: Items 1&2&3 = 4 daysCompleted: Items 1&2&3 = 0 Dependency : Items 1&2 = 1 Item 3 = 0	<ul style="list-style-type: none"> $Rank_{Item3} = 0.33 * 1 + 0.33 * 0.5 + 0.33 * 0 = 0.495$ $Rank_{Items2&1} = 0.33 * 1 + 0.33 * 0.5 + 0.33 * 0.33 = 0.605$ <p>Thus, the order of the tasks should be:</p> <p>Item1 Item2 Item3</p>		Pass	
19. Adding an item with 1 dependency, and all other items have 1 dependency	Importance: Items 1&2&3 = 0.5 Due date: Item 1&2&3 = due four days from current date Duration: Items 1&2&3 = 4 daysCompleted: Items 1&2&3 = 0 Dependency : Items 1&2&3 = 1	<ul style="list-style-type: none"> $Rank_{Item1&2&3} = 0.33 * 1 + 0.33 * 0.5 + 0.33 * 0.33 = 0.605$ <p>Thus, the order of the tasks should be:</p> <p>Item2 Item3 Item1</p>		Pass	No comment
	Importance:				No comment

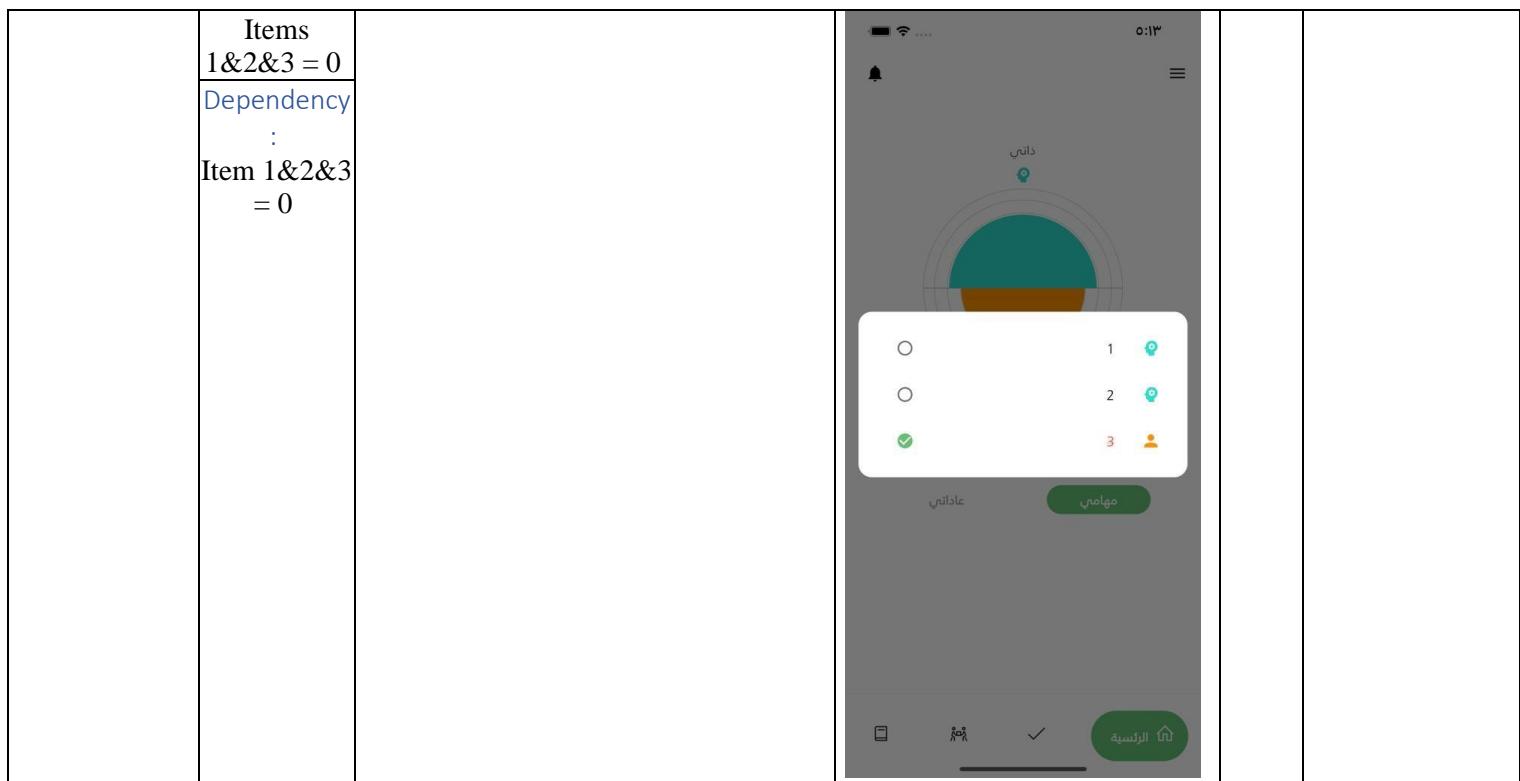
20. Adding an item with 0 dependencies, an item with 1 dependency , an item with 2 dependencies, and an item with 3 dependencies	Items 1&2&3 = 0.5 Due date: Item 1&2&3 = due four days from current date Duration: Items 1&2&3 = 4 daysComPLETED: Items 1&2&3 = 0 Dependency : Item 1 = 2 Item 2 = 3 Item 3 = 1 Item 4 = 0	<ul style="list-style-type: none"> Rank_{Item1}=0.33*1+0.33*0.5+0.33*0.5 = 0.660 Rank_{Item2}=0.33*1+0.33*0.5+0.33*0.75 = 0.7425 Rank_{Item3}=0.33*1+0.33*0.5+0.33*0.25 = 0.5775 Rank_{Item4}=0.33*1+0.33*0.5+0.33*0 = 0.4950 <p>Thus, the order of the tasks should be:</p> <table style="margin-left: 100px;"> <tr><td>Item2</td></tr> <tr><td>Item1</td></tr> <tr><td>Item3</td></tr> <tr><td>Item4</td></tr> </table>	Item2	Item1	Item3	Item4		Pass
Item2								
Item1								
Item3								
Item4								
21. Adding items with different values in each term	Importance: Item 1 = 0.75 Items 2&3 = 0.5 Item 4 = 0.25 Due date: Item 1 = due nine days from current date Item 2&3 = due ten days from current date Item 4 = due five days from current date Duration: Item 1 = 4 Item 2 = 6	<ul style="list-style-type: none"> Rank_{Item1}=0.33*0.74+0.33*0.75+0.33*0 = 0.4919 Rank_{Item2}=0.33*1+0.33*0.5+0.33*0.25 = 0.5775 Rank_{Item3}=0.33*0.5+0.33*0.5+0.33*0.25 = 0.4125 Rank_{Item4}=0.33*0.67+0.33*0.25+0.33*0 = 0.3025 <p>Thus, the order of the tasks should be:</p> <table style="margin-left: 100px;"> <tr><td>Item2</td></tr> <tr><td>Item1</td></tr> <tr><td>Item4</td></tr> <tr><td>Item3</td></tr> </table>	Item2	Item1	Item4	Item3		No comment
Item2								
Item1								
Item4								
Item3								

	Item 3 = 3 Item 4 = 2 daysCompleted: Items 1&2&3&4 = 0 Dependency: : Item 1&4 = 0 Item 2&3 = 1			
22. Adding items with different values in each term	<p>Importance: Items 1&2&3 = 0.5 Item 4 = 0.75</p> <p>Due date: Item 1 = due nine days from current date Item 2&3 = due ten days from current date Item 4 = due three days from current date</p> <p>Duration: Item 1 = 1 Item 2 = 5 Item 3 = 4 Item 4 = 2</p> <p>daysCompleted: Items 1&2&3&4 = 0</p> <p>Dependency: : Item 1&4 = 0</p>	<ul style="list-style-type: none"> • $Rank_{Item1}=0.33*0.60 +0.33*0.5+0.33*0 = \mathbf{0.363}$ • $Rank_{Item2}=0.33*0.75 +0.33*0.5+0.33*0.25 = \mathbf{0.4950}$ • $Rank_{Item3}=0.33*0.60 +0.33*0.5+0.33*0.25 = \mathbf{0.4455}$ • $Rank_{Item4}=0.33*1+0.33*0.75+0.33*0 = \mathbf{0.5775}$ <p>Thus, the order of the tasks should be: Item4 Item2 Item3 Item1</p>		Pass No comment

	Item 2&3 = 1			
23. Adding items with different values in each term	<p>Importance: Items 1&2&3 = 0.5 Item 4 = 0.75</p> <p>Due date: Item 1 = due nine days from current date</p> <p>Item 2&3 = due ten days from current date</p> <p>Item 4 = due three days from current date</p> <p>Duration: Item 1 = 1 Item 2 = 5 Item 3 = 4 Item 4 = 2</p> <p>daysCompleted: Items 1&3&4 = 0 Item 2 = 1</p> <p>Dependency : Item 1&4 = 0 Item 2&3 = 1</p>	<ul style="list-style-type: none"> • $Rank_{Item1}=0.33*0.17+0.33*0.5+0.33*0 = 0.22$ • $Rank_{Item2}=0.33*0.6+0.33*0.5+0.33*0.25 = 0.4455$ • $Rank_{Item3}=0.33*0.6+0.33*0.5+0.33*0.25 = 0.4455$ • $Rank_{Item4}=0.33*1+0.33*0.75+0.33*0 = 0.5775$ <p>Thus, the order of the tasks should be: Item4 Item3 Item2 Item1</p>		Pass No comment
24. Checking an Item that has over all higher ranking than other items	<p>Importance: Items 1&2 = 0.5 Item 3 = 0.75</p> <p>Due date: Items 1&2 = due four days</p>	<ul style="list-style-type: none"> • $Rank_{Item3}=0$ • $Rank_{Items2&1}=0.33*0.5+0.33*0.5+0.33*0 = 0.33$ 		Note: in terms of the equation, Item3 has higher overall rank even after being checked; but for enhancing the usability and user experience, checked items

	<p>from current date Item 3 = due one day from current date</p> <p>Duration: Items 1&2&3 = 2 daysComplet ed: Items 1&2 = 0 Item 3 = 1</p> <p>Dependency : Item 1&2&3 = 0</p>	<p>Thus, the order of the tasks should be: Item1 Item2 Item3</p>		<p>now move to the bottom of the list</p>
25. One item is past due	<p>Importance: Items 1&2 = 0.5 Item 3 = 0.75</p> <p>Due date: Items 1&2 = due four days from current date Item 3 = due one day before the current date</p> <p>Duration: Items 1&2&3 = 2</p> <p>daysComplet ed: Items 1&2&3 = 0</p> <p>Dependency :</p>	<ul style="list-style-type: none"> • $Rank_{Item3}=1$ • $Rank_{Items2\&1}=0.33*0.5+0.33*0.5+0.33*0=0.33$ <p>Thus, the order of the tasks should be: Item3 Item1 Item2</p>		<p>Pass</p> <p>Note: For enhancing the usability and user experience, due items are moved to the top of the list.</p>

	Item 1&2&3 = 0			
26. Two items are past due	<p>Importance: Items 1&2 = 0.5 Item 3 = 0.75</p> <p>Due date: Items 1 = due four days from current date Item 2&3 = due one day from current date</p> <p>Duration: Items 1&2&3 = 2</p> <p>daysCompleted: Items 1&2&3 = 0</p> <p>Dependency: Item 1&2&3 = 0</p>	<ul style="list-style-type: none"> • $Rank_{Item1} = 0.33 * 1 + 0.33 * 0.75 + 0.33 * 0 = 0.5775$ • $Rank_{Items2&3} = 1$ <p>Thus, the order of the tasks should be: Item2 Item3 Item1</p>		Pass
27. Checking an Item that is past due	<p>Importance: Items 1&2 = 0.5 Item 3 = 0.75</p> <p>Due date: Items 1&2 = due four days from current date Item 3 = due one day before the current date</p> <p>Duration: Items 1&2&3 = 2</p> <p>daysCompleted:</p>	<ul style="list-style-type: none"> • $Rank_{Item3} = 0$ • $Rank_{Items2&1} = 0.33 * 0.5 + 0.33 * 0.5 + 0.33 * 0 = 0.33$ <p>Thus, the order of the tasks should be: Item1 Item2 Item3</p>		<p>Note: For enhancing the usability and user experience, due items are moved to the top of the list.</p>
				<p>Importance: Items 1&2 = 0.5 Item 3 = 0.75</p> <p>Due date: Items 1&2 = due four days from current date Item 3 = due one day before the current date</p> <p>Duration: Items 1&2&3 = 2</p> <p>daysCompleted:</p>



6.3.2 Community

Motazen's community feature was implemented using a realtime database. Within the database, an array of objects was used, with the first part of each object being the millisecond of when it was created, concatenated with the user's unique ID (UID) from the database. The same method is applied when a user creates a community, ensuring that every community had a UID. The post channel has one object named "post" that stores messages, including the sender's name (author), ID, message type (text or image), the text itself, and time the message

```
class PostModel {
    String? author;
    String? authorId;
    final time;
    final text;
    final comments;
    final replyingPost;
    final likes;
    final progreebar;
    String? postType;
    final imageURL;
```

Figure 35 Post Model with variables

was sent. The time is used for sorting to display the messages in the correct order on the interface.



Figure 36 realtime database of post channels: Communities



Figure 37 realtime database of posts: Messages

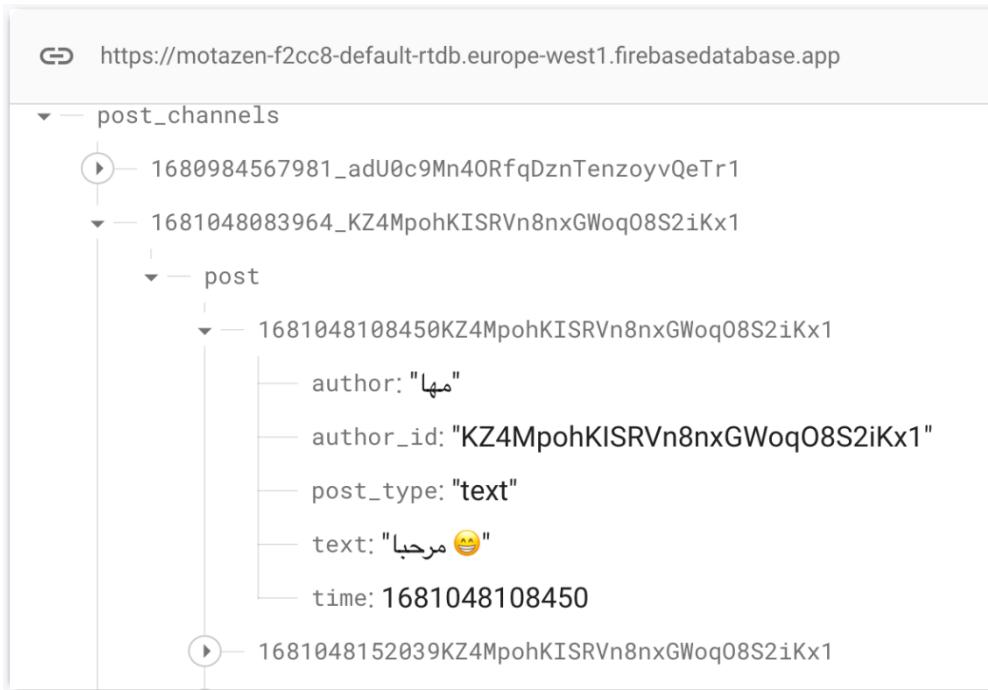


Figure 38 Message Example

In the firestore database, every community has a community UID that is structured in the same way as in the realtime database (time and creator ID). Once a user joins a community, the data is fetched from the realtime database via the UID in the post channel. The users communicate with each other by fetching each other's data from the "joined communities" array. Since each community has a UID, we can determine whether a user is part of the community or not. When a user is now a member of the community, the app looks through each user to see if they have a matching community UID. If there is a match, the user is now considered part of this community, and all users in the community could be fetched. Which leads to how the users communicate with each other.

Every post that is made by a member in the community is stored in the realtime database and then sent to all members that were previously fetched. This meant that it was necessary to have a constant internet connection for the community feature to work. The implementation of this feature was designed to ensure that each community had a UID and that users could communicate with each other effectively. By fetching each other's data from the "joined communities" array, users could stay up to date with the latest posts and discussions within the community



6.4 Integration

System integration is the process of connecting many components, or sub-systems, into a single, bigger system that functions as a single unit. Through integration, these components are intended to communicate across the Motazen's multiple files, accelerating information flows and lowering expenses.

We employed a bottom-up approach to integrate our system. Bottom-up testing is a type of incremental integration testing approach in which testing is carried out by integrating or combining two or more components as they move upward through the control flow of the architecture structure from bottom to top. In bottom-up testing, the low-level components are tested first, followed by high-level modules, which is a more user-friendly testing that leads to better overall software development. First, we linked the components that were related to one another, such as the sign-up form and the onboarding page. Then we moved on to less related, larger components, and so forth. This method aided in the integration process and allowed us to manage our application in smaller doses.

6.5 Challenges

During software development, we encountered our first challenge in Firebase authentication. In contrast to our prior experience with web development where authentication was simply validating with the database, the mobile application's authentication process involved several time-consuming steps. No matter which platform we decided for our program to operate on, we first had to link it to the database so that it would work on both iOS and Android.

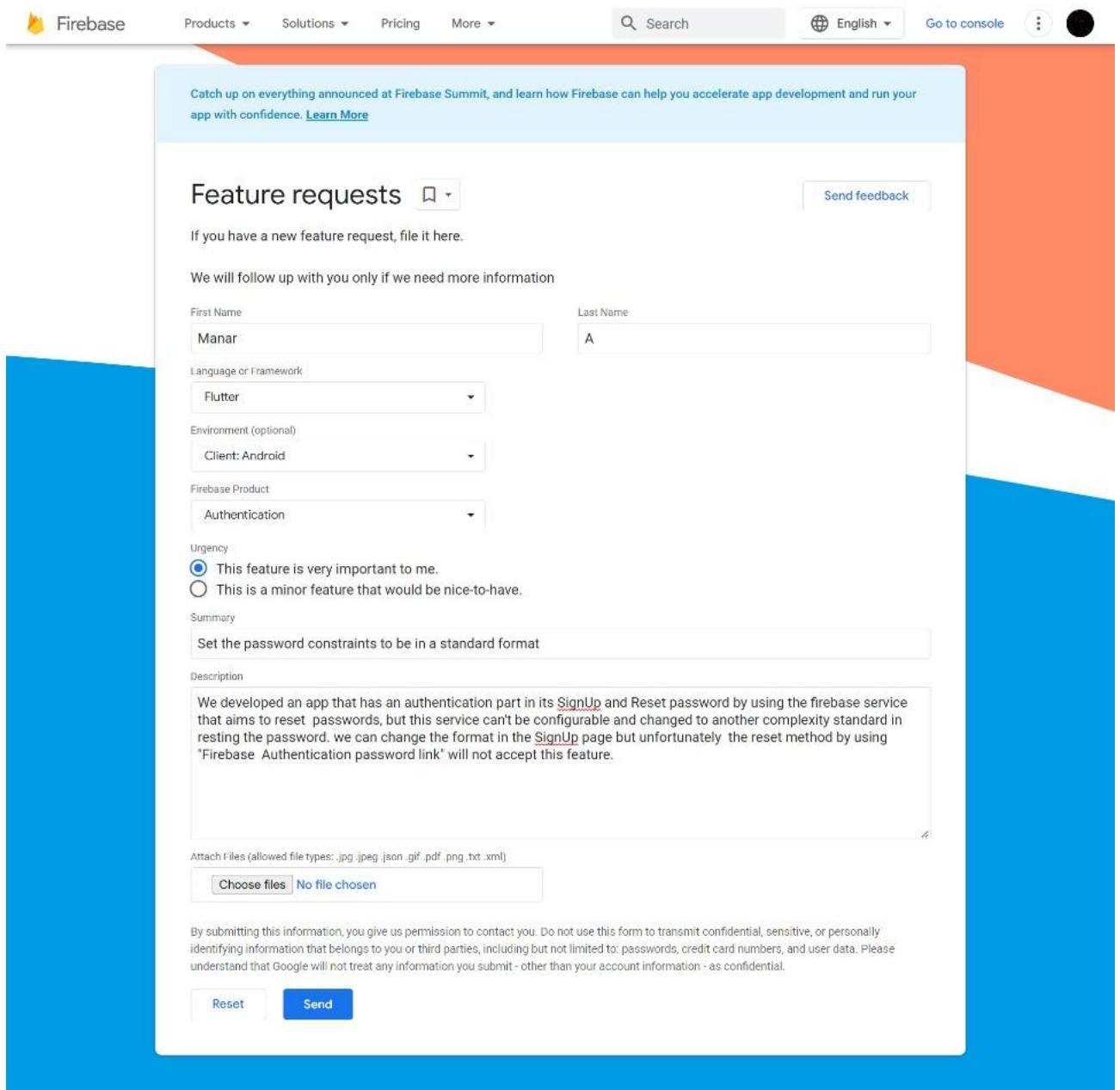
In addition to that, we faced an error with validation. Despite the validation code functioning and running without errors, it would not appear on our interface, which caused confusion since the code itself ran as expected. Our second challenge was concerning the fact that the development team was somewhat inexperienced with Flutter and Dart, so implementing the “Reset Password” method was difficult at first. We solved this issue by using a One Time Password (OTP) package, which sends the OTP to the user’s email after they enter it.



During implementation, we encountered another password challenge while using Firebase, where we couldn't manually create a standard format for the passwords. Firebase doesn't allow developers to manipulate their format, as it is part of their firebase auth with a link method. During sign up, users have the ability to create an eight-character-or-above password, since we have implemented this section without Firebase, but if said user forgets their password and requests to "Reset Password", we must conform to Firebase's six-character password. This inconsistency might confuse users, so we have sent a feature request form to Firebase to allow us to add eight or more characters. (see figure 32).

Another main challenge consisted of creating a signing report, which required using a Java terminal. This was unexpected since the team was unaware of the fact and was required to redownload Java and its IDEs.

By far, the toughest challenge was passing the data throughout different classes without making them global variables, since it is bad practice. The team fixed this issue by researching a package named "Provider" which allowed us to transfer these shared variables whilst still adhering to good coding practices. Segmenting the code and unifying our interfaces regardless of platform/device was also a major challenge that we overcame by using only one platform for the user interface, and both for implementing.



The screenshot shows the 'Feature requests' section of the Firebase website. At the top, there's a banner about the Firebase Summit. Below it, a heading says 'Feature requests' with a dropdown arrow. A sub-instruction says 'If you have a new feature request, file it here.' A 'Send feedback' button is in the top right corner.

The form fields include:

- First Name: Manar
- Last Name: A
- Language or Framework: Flutter
- Environment (optional): Client: Android
- Firebase Product: Authentication
- Urgency: This feature is very important to me.
 This is a minor feature that would be nice-to-have.
- Summary: Set the password constraints to be in a standard format
- Description: A text area containing: We developed an app that has an authentication part in its SignUp and Reset password by using the firebase service that aims to reset passwords, but this service can't be configurable and changed to another complexity standard in resetting the password. we can change the format in the SignUp page but unfortunately the reset method by using "Firebase Authentication password link" will not accept this feature.
- Attachment: A placeholder 'Choose files' with 'No file chosen'.

At the bottom, a note states: By submitting this information, you give us permission to contact you. Do not use this form to transmit confidential, sensitive, or personally identifying information that belongs to you or third parties, including but not limited to: passwords, credit card numbers, and user data. Please understand that Google will not treat any information you submit - other than your account information - as confidential.

Buttons at the bottom are 'Reset' and 'Send'.

Learn
Guides
Reference
Samples
Libraries
GitHub

Stay connected
Blog
Firebase Summit
Twitter
YouTube

Support
Contact support
Stack Overflow
Slack community
Google group
Release notes
FAQs

[Google Developers](#) [Android](#) [Chrome](#) [Firebase](#) [Google Cloud Platform](#) [All products](#)

[Terms](#) | [Privacy](#)

[English](#)

Figure 39 Feature Request in Firebase



Implementing a community feature in Motazen was one of the biggest coding challenges we faced. We had to carefully consider how to create a functioning yet user-friendly platform that allowed users to connect with each other and share their progress, knowledge, and experiences. This required a significant amount of testing and debugging to ensure that the application functioned properly.

Developing an efficient multi-criteria decision-making algorithm was a significant challenge during the implementation of Motazen. Our goal was to provide users with a powerful tool that could evaluate and prioritize their progress and goals based on various criteria such as importance, dependency, and due date. The algorithm had to be carefully tested and refined to ensure that it produced accurate and meaningful results for our users while maintaining optimal performance and speed. Despite these difficulties, we were able to successfully incorporate the algorithm into the code, giving users an effective way to achieve balance and perspective in their lives.

Chapter 7:

System Testing



7 System Testing

7.1 User Acceptance Testing

User Acceptance Testing (UAT), also known as end user testing, is the last phase of the software testing process and development life cycle. The objective of UAT is to allow the end-user to verify and confirm the requirements and decide whether the application satisfies. The main goal is to guarantee that software can complete tasks in the real world and meet performance requirements.

The testing process will be divided into two main segments. First, we will obtain general reactions to the system as we begin by asking the user to run the app how they would run any mobile application and analyze how they navigate and use the app. The second part will consist of a questionnaire taken anonymously by the users. The survey created asks a variety of questions about the program to gather feedback that helps us examine the users' reaction to the overall system, system interface, learning to operate the system, and testing our non-functional requirements. It is a test created by SMU, Singapore Management University, that has been used to conduct UAT tests since 2012 [\[43\]](#). We have modified the survey as necessary to fit our application, while still maintaining its structure and key questions, that can be found in [Appendix B](#).



7.1.1 Demographics of Participants

Provide relevant demographic information of the UAT testers in a tabular or graphic format.

Table 7 Demographics of Participants.

Variable	Value	Count (n=6)	Percentage
Age	18-30	3	50%
	30-50	2	33.33%
	50+	1	16.67%
Gender	Male	2	33.33%
	Female	4	66.67%
Technical Level	Novice	1	16.67%
	Intermediate	3	50%
	Proficient	2	33.33%

7.1.2 Questionnaire/Interview Results

We asked six participants a series of questions on the Motazen app. There were four sections to the questions. We wanted to know if the overall feel of the program was positive, so we asked general system questions in the first section. The second section was questioning the participant about the application's interface and design. The third section was asking the testers how easy or difficult it was learning to operate the system. The following figures will present our team of tester's results.

Strongly Disagree 1	Disagree 2	Neutral 3	Agree 4	Strongly Agree 5
------------------------	---------------	--------------	------------	---------------------

I found the system's interface an easy way to perform system functions.

6 responses

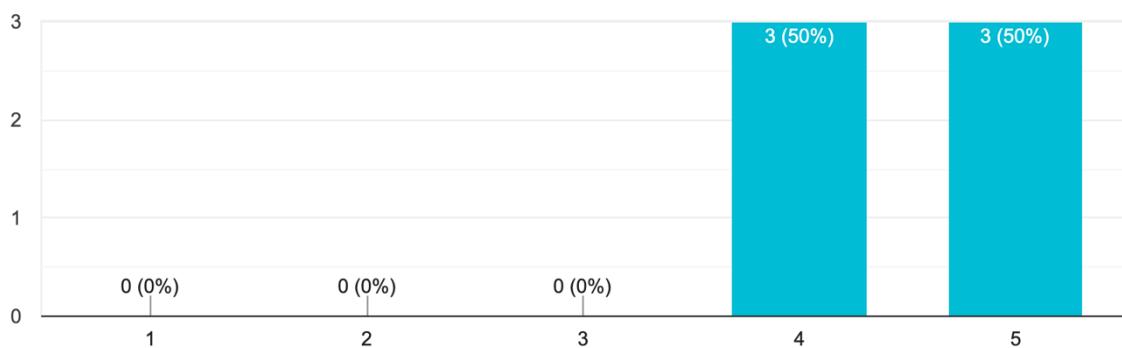


Figure 40 histogram of first question's results

I found the error messages in the application are clear and indicate the details of the error

6 responses

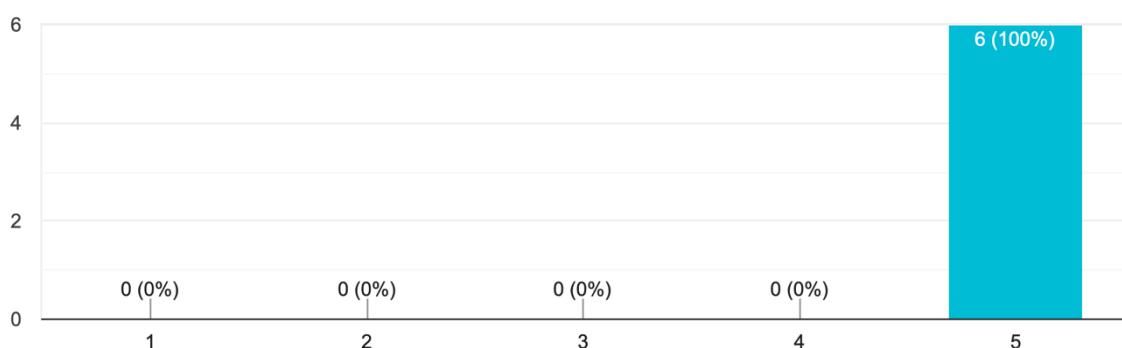


Figure 41 histogram of second question's results

I felt the system responded in a consistent and predictable way.

6 responses

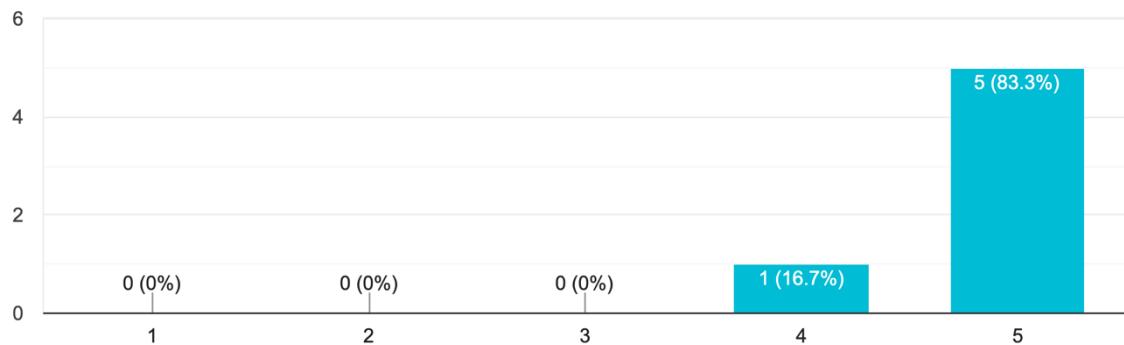


Figure 43 histogram of the third question's results

I found the terminology used to be clear and precise.

6 responses

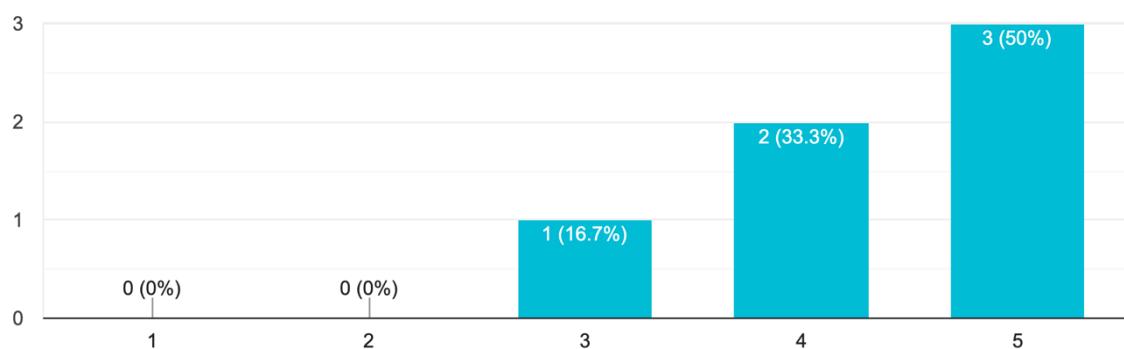


Figure 44 histogram of the fourth question's results

I always knew where to access the information I wanted

6 responses

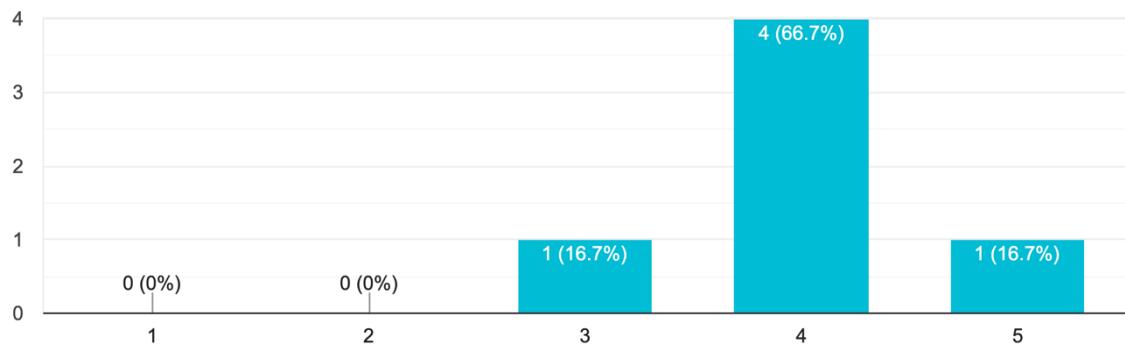


Figure 45 histogram of the fifth question's results

I get enough feedback after adding/editing/deleting a goal/habit/task

6 responses

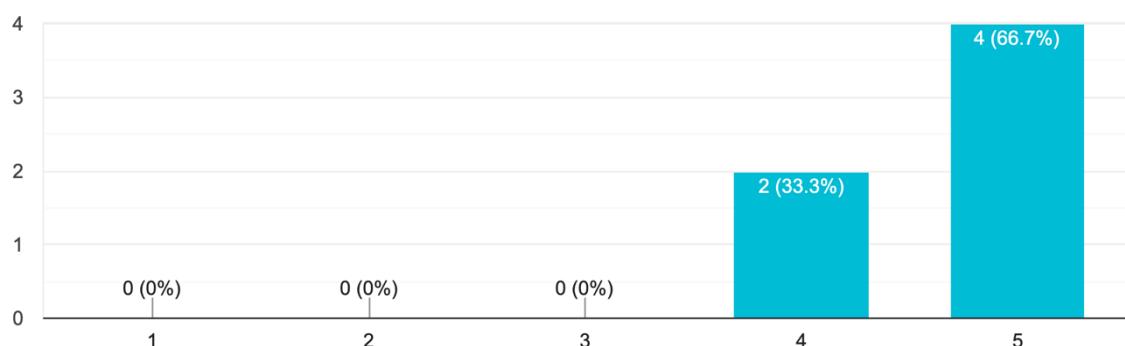


Figure 46 histogram of the sixth question's results

I journal my daily thoughts with ease.

6 responses

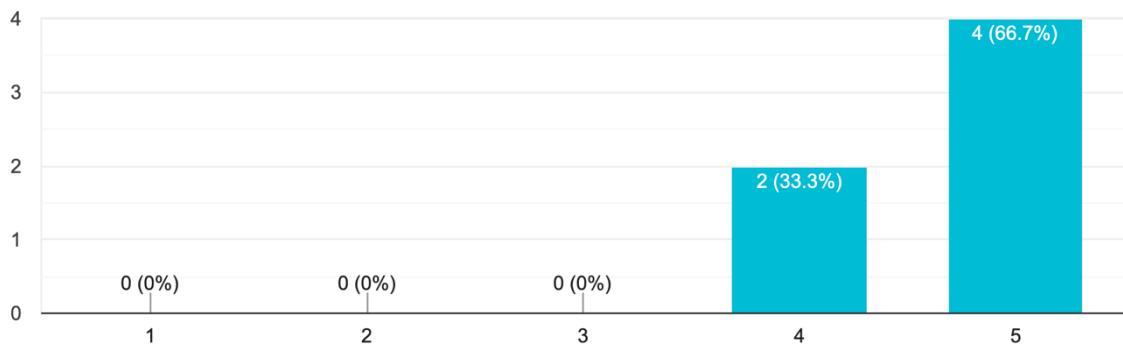


Figure 47 histogram of the seventh question's results

I can search for and join communities easily.

6 responses

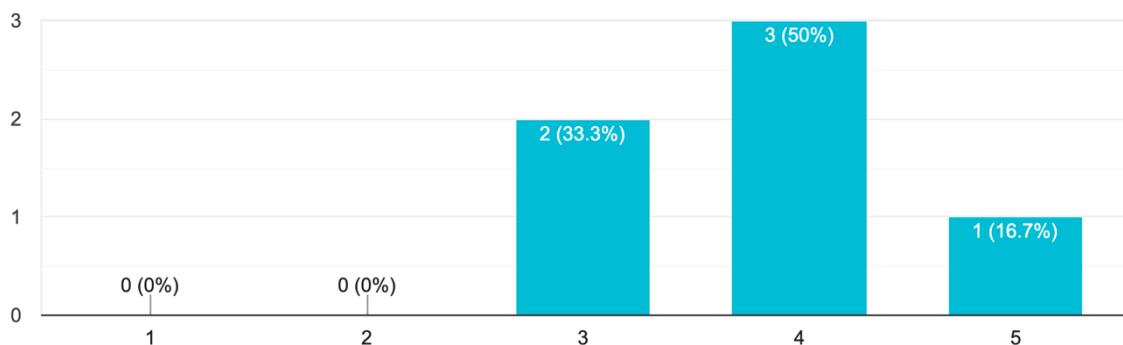


Figure 48 histogram of the eighth question's results

I exchange messages and pictures with members in communities.

6 responses

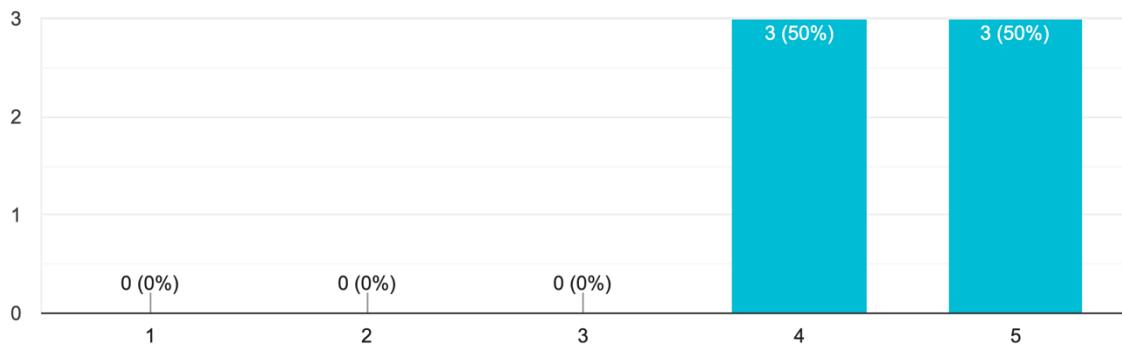


Figure 49 histogram of the nineth question's results

I can create communities.

6 responses

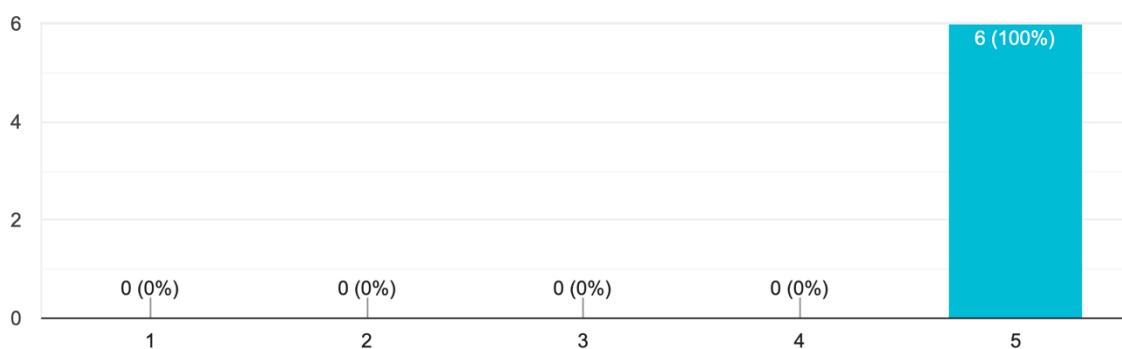


Figure 50 histogram of the tenth question's results

I can invite friends to join my communities

6 responses

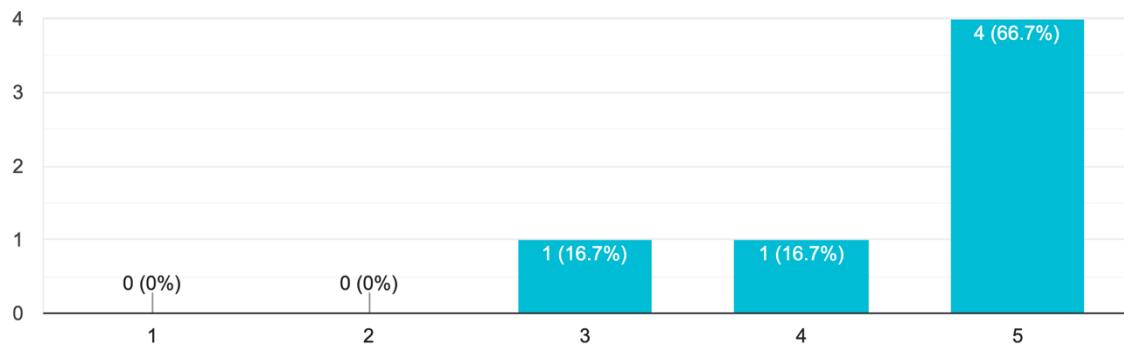


Figure 51 histogram of the eleventh question's results

The screen layout was clear and helpful

6 responses

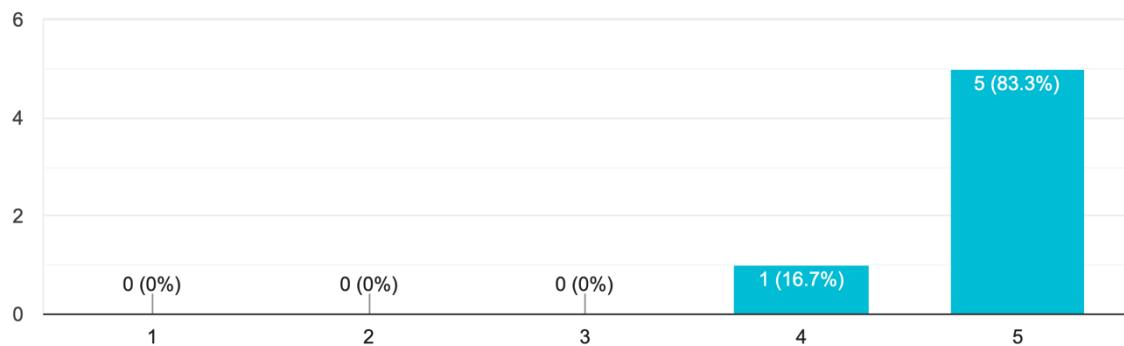


Figure 52 histogram of the twelfth question's results

The amount of information displayed on screen was sufficient.

6 responses

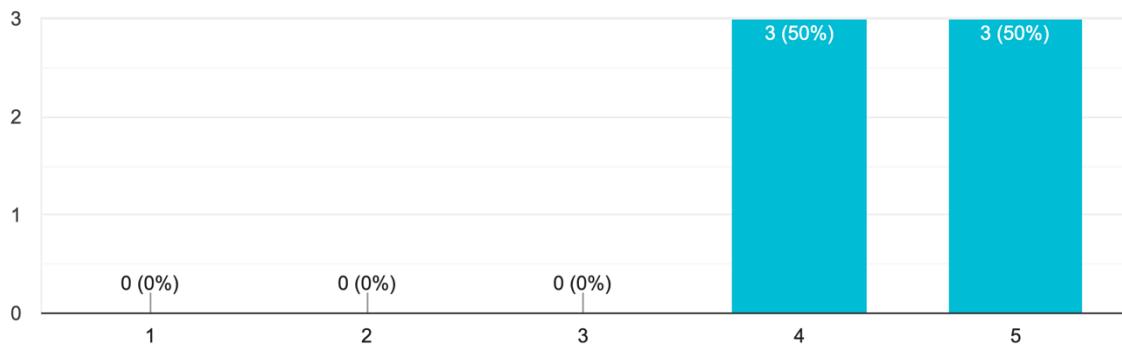


Figure 53 histogram of the thirteenth question's results

The sequence of screens was predictable and made sense.

6 responses

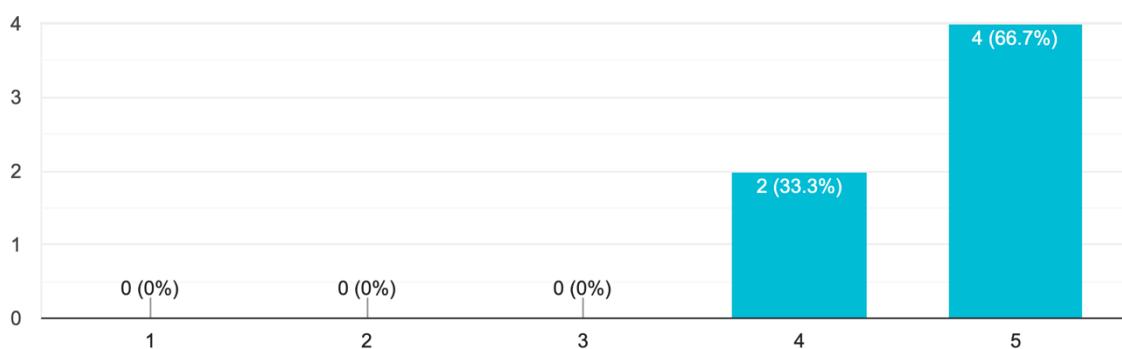


Figure 54 histogram of the fourteenth question's results

The interface design was friendly and aesthetically pleasing.

6 responses

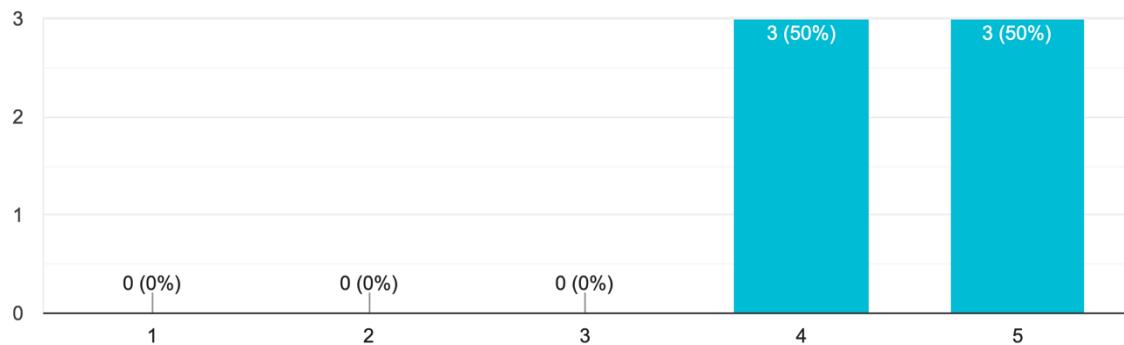


Figure 55 histogram of the fifteenth question's results

I needed help getting started with the system.

6 responses

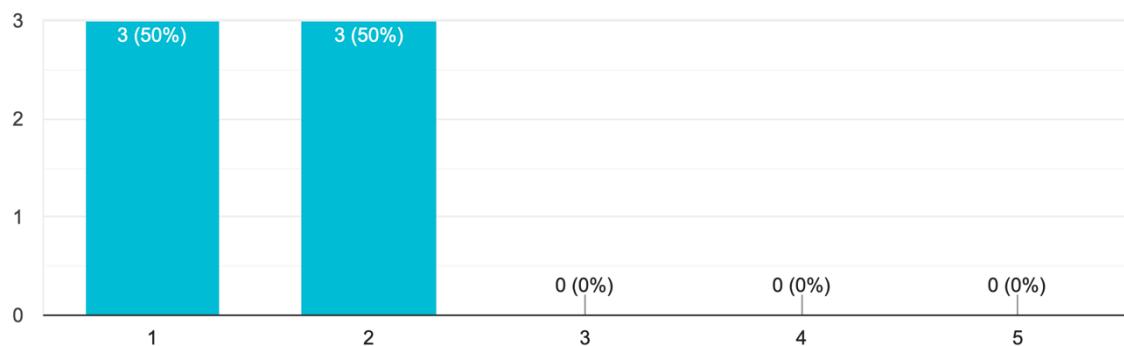


Figure 56 histogram of sixteenth question's results

I had to remember names and use of commands.

6 responses

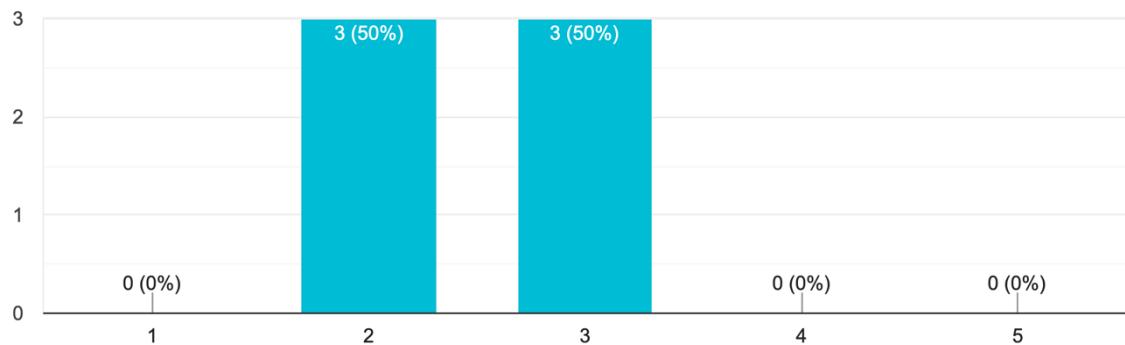


Figure 57 histogram of seventeenth question's results

I had to remember specific rules about clicking buttons.

6 responses

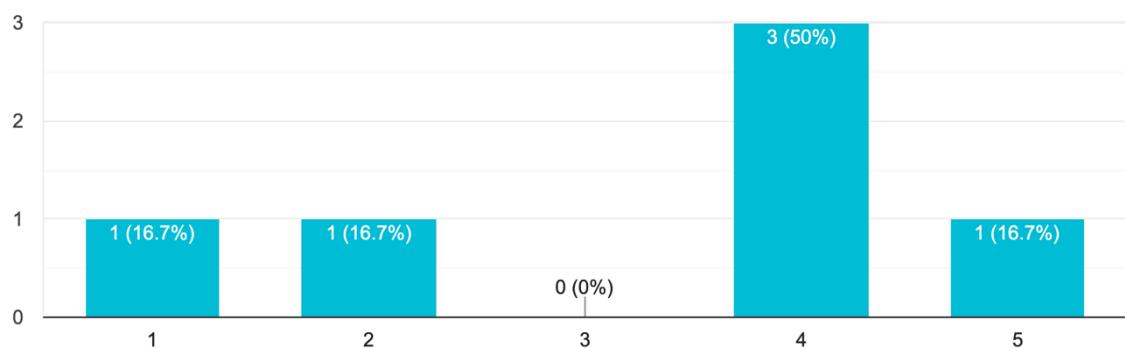


Figure 58 histogram of eighteenth question's results



The tasks were performed in a straightforward manner.

6 responses

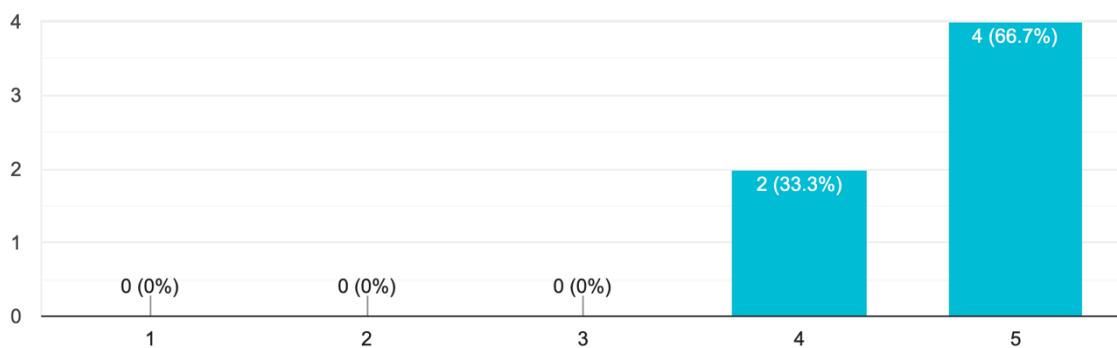


Figure 59 histogram of nineteenth question's results

The findings were based on a rating system ranging from 1 to 5, where 1 indicated strong disagreement and 5 indicated strong agreement. Six testers participated in the study, and all of them concurred that the system's interface was user-friendly and aided in the execution of its functions. They also strongly agreed that the error messages were clear and informative, and that the system responded in a predictable manner. Although the majority of users found the terminology used to be clear, one user remained neutral on the subject. It was reassuring to discover that most users could locate the desired information, with only one user remaining neutral on this issue. Furthermore, all testers agreed that they received adequate feedback after adding a goal, habit, or task. With respect to the recently added features, all users agreed that they could use the journal function with ease, but about 30% of our users had a neutral response when it came to searching for and joining communities. To address this, we reworded our approach and added feedback after joining a community. Nonetheless, all users agreed that they could create communities and exchange messages and images with other community members. Nearly all users reported that they could easily invite friends to private communities. In the interface section of the survey, we received highly positive results for all questions, as all testers found the interface to be aesthetically pleasing, and the layout was clear and helpful. No testers encountered any interface issues. In terms of learning to use the system, none of the testers required assistance getting started with the system. Half of the users found that they did not need to remember the names and commands of features, while the other half were neutral on the matter. Although the percentage of users who had to remember specific button-clicking



rules was moderate, all testers agreed that their tasks and application usage were performed in a straightforward manner.

7.2 Quality Attributes (Non-functional testing)

This section outlines the non-functional requirements (NFR) of our application, which include Usability (error), Usability (Learnability), and Availability. For assessing the usability requirements, we selected six users and provided them with the application to try. We then recorded the time it took each user to complete their assigned user story.

Table 8 NFR Testing

User Story	Quality Attribute	Measure	Results
As a user, I want to be able to successfully perform application functions, so that my average number of errors per task ⁸ is less than 0.7 ⁹ .	Usability – error: the ease with which users can navigate the application without encountering errors or bugs.	Measured using the following equation: $\frac{\text{No. of errors occurring during task}}{\text{No. of task attempts}}$	We assigned 19 tasks to each of the users. Upon analyzing the results, we found that even the user that encountered the most errors (5), is still less than our target number (0.7). $\frac{5}{19} = 0.26$
As a user, I want to be able to learn all application functions in less than 3 minutes ¹⁰ , so that I do not get irritated and exit the app.	Usability – Learnability: how quickly users can learn how to use the application and how intuitive the application's design is.	Measure the duration of time it takes for users to become familiar in using the system. Specifically, the goal is for users to acquire the necessary knowledge of the app's features within three minutes or less.	To evaluate the system's usability, we selected six users and recorded the duration it took them to become acquainted with the system. From our findings, 33.4% of the testers became familiar with the system in under three minutes, 16.7% took approximately three minutes, while 50% of the users took more than three minutes to become familiar with the system.

⁸ The equation used for measuring the number of error per task is as follows: total number of error/ total number of tasks, where the total number of errors is the total number of errors that occur during testing.

⁹ <0.7 is known to be the standard for the error per task result [34]

¹⁰ The results were benchmarked from other applications



<p>As a user, I want the system to be available 99.999%¹¹ of the time I try to access it, so that I do not get irritated and find another app to use.</p>	<p>Availability: the amount of time that the application is accessible and functioning. It includes metrics such as uptime and downtime</p>	<p>Measure the availability of the system using the following equation:</p> $\text{Availability} = \frac{\text{Uptime}}{\text{Uptime} - \text{Downtime}}$	<p>We have identified that our application relies on the Firebase database service, which promises to use commercially reasonable measures to maintain a Monthly Uptime Percentage (defined below) of at least 99.95% during each billing cycle. This commitment is referred to as the "Service Commitment." [47]</p>
--	--	---	---

7.3 Discussion

Based on the information provided, it can be deduced that the system under evaluation has a user-friendly interface that enables users to perform its functions efficiently. Additionally, the error messages are clear and informative, and the system responds in a predictable manner. Most users were able to locate the desired information easily, and the majority of users found the terminology used to be clear. The newly added features, such as the journal function, were found to be easy to use by all users, while the community section was rated positively but with some neutral responses, which prompted us to improve the wording and add more feedback, particularly for the invite function. We notified the invitee once he/she has been invited to a community, to ensure that no issues arise from the sender. Learning all features of the application within the first three minutes was a contentious issue, given the diverse demographic of the users.

¹¹ 99.999% also known as the five nines is what most vendors use for their products, it indicates a down time of 6.05 seconds per week. [35]

Chapter 8: Conclusions & Future Work



8 Conclusions and Future Work

8.1 Local Impact

The main objective of Motazen was to help people become their best selves. By incorporating the Wheel of Life technique into a ranked to-do list, we are able to first examine each person's points of imbalance in order to better understand their areas of weakness and help them aspire to improve their lives in numerous ways. By letting the user select which of the key ones to emphasize, we kept the approach to the attributes flexible. Additionally, including a community section fosters a feeling of comfort and unity because you know that all the users nearby are working towards your same objective. In accordance with one of our 2030 Saudi Visions, “Creating a Vibrant Society”, we believe that a strong society is characterized by care and a strong sense of community [44]. The significance of this community feature goes beyond the individual level and can have a broader impact on society. By promoting connections and support among users, Motazen cultivates a sense of empathy, understanding, and unity. This can lead to improved mental health, better general well-being, reduced social isolation, and increased community engagement. Overall, the community feature has the potential to bring up positive social change and create a more connected and supportive society.

8.2 Global Impact

There has been a recent increase in awareness of the value of maintaining task managers and to-do lists. Many technological management techniques have been used by people worldwide to get by, and the results have proven that they are much more productive [45]. It goes without saying that including a crucial feature of ranking daily tasks will help people everywhere, especially those who struggle with ADHD, to accomplish and stay focused on what's important throughout their day [46].



8.3 Problems and challenges

The two main challenges faced during implementation and development was passing our variables throughout different classes without using global variables, since they can be altered by any part of the code and thus has no access control. The second tough challenge was finding an effective solution to catch firebase authentication exceptions, especially since with the new and young language, Flutter, error codes change often by Flutter developers. This resulted in the team facing errors with little to no documentation and forced into testing a variety of packages to find the best fit.

8.4 Limitations of the system

Motazen is fixated on the eight main aspects of a user's life, and although users can choose amongst these eight, they cannot add nor edit the aspects. Also, the application currently only supports iOS and the Arabic language. Our application is limited in scalability where it only serves limited number of user interactions at once.

8.5 Main contribution of the project

Our project contributed to documenting the importance of regulating and coaching users based on imbalanced aspects in their lives and improve them by first selecting and acknowledging what areas need development, then visualize their "Wheel of Life" to help them better understand it and take the necessary steps of adding goals and habits to balance their life's aspects. We made it possible for people to assess their current situation in a simple, realistic way in hopes of that they benefit from the application and contribute to their general growth.

Motazen's second release includes a new feature of communities, allowing users to create and join public and private communities within the app and to connect with people who have similar interests and goals. Additionally, users can use the journaling tool to reflect their daily thoughts and insights. The introduction of ranked tasks is another significant addition to Motazen. Users can now rank and prioritize their tasks according to their importance, due date, and dependency, allowing them to concentrate on the most critical aspects of their life. By



assisting users in better time and resource management, this feature enhances their productivity and effectiveness in achieving their goals.

8.6 Future Work

As a life coaching application, Motazen has the potential to provide even more value to users through the addition of new features and functionalities. In the future, we could explore incorporating features such as personalized coaching sessions with certified life coaches, guided meditations, and a wrap-up feature at the end of each year.

Personalized coaching sessions could provide users with one-on-one support from trained professionals, allowing them to receive customized guidance and support as they work towards their goals. Guided meditations could help users reduce stress and improve their overall well-being by providing them with access to calming exercises that they can practice at any time.

Another potential feature that could be added to Motazen is an end-of-year "wrap-up" feature, similar to Spotify Wrapped or Snapchat Memories. This feature could provide users with a personalized summary of their progress and achievements over the past year, highlighting key milestones and accomplishments.

The wrap-up feature could include a detailed breakdown of how the user has improved in each area of their life, based on their input and progress tracked throughout the year. It could also offer personalized recommendations and goal-setting prompts for the coming year, based on the user's individual needs and priorities.

This feature would be a valuable tool for users to reflect on their progress and celebrate their accomplishments, while also providing them with a roadmap for continued growth and development in the coming year. It would also help to reinforce the value of using the Motazen application as a tool for self-improvement and personal growth and encourage users to continue using the application over the long term.

Chapter 9:

References



9 References

- [1] Schwantes, M. (2022, January 8). Studies show 91 percent of Us won't achieve our New Year's resolutions. how to be the 9 percent that do. Inc.com. Retrieved September 4, 2022, from <https://www.inc.com/marcel-schwantes/studies-show-91-percent-of-us-wont-achieve-our-new-years-resolutions-how-to-be-9-percent-that-do.html>.
- [2] Byrne, U. (2005). Wheel of life. Business Information Review, 22(2), 123–130. <https://doi.org/10.1177/0266382105054770>.
- [3] Brower, T. (2022, January 13). 4 reasons to make New Year's resolutions (even if you don't keep them). Forbes. Retrieved September 4, 2022, from <https://www.forbes.com/sites/tracybrower/2021/12/31/4-reasons-to-make-new-years-resolutions-even-if-you-dont-keep-them/>
- [4] Claridge, T., & About The Author Tristan Claridge Tristan Claridge has a passion for technology. (2022, July 30). Shared goals, shared purpose, Shared Vision • Institute for Social Capital. Institute for Social Capital. Retrieved September 4, 2022, from <https://www.socialcapitalresearch.com/shared-goals-shared-purpose-shared-vision/>
- [5] the Mind Tools Content Team By the Mind Tools Content Team, Team, the M. T. C., wrote, S. H., wrote, utibe_ebong, & wrote, B. T. (n.d.). The Wheel of Life®: – finding balance in your life. The Wheel of Life - Time Management Techniques from MindTools.com. Retrieved September 5, 2022, from https://www.mindtools.com/pages/article/newHTE_93.htm
- [6] A. Inc., “Xcode 14 overview,” Apple Developer. [Online]. Available: <https://developer.apple.com/xcode/>. [Accessed: 16-Sep-2022].
- [7] “Flutter architectural overview,” Flutter. [Online]. Available: <https://docs.flutter.dev/resources/architectural-overview>. [Accessed: 16-Sep-2022].
- [8] K. Cherry, “What does a life coach do?,” Verywell Mind, 22-Feb-2022. [Online]. Available: <https://www.verywellmind.com/what-is-a-life-coach-4129726>. [Accessed: 14-Sep-2022].



- [9] U. Byrne, “Wheel of life,” *Business Information Review*, vol. 22, no. 2, pp. 123–130, 2005.
- [10] B. O'Brien, “The wheel of life in Tibetan buddhism,” *Learn Religions*, 22-Dec-2018. [Online]. Available: <https://www.learnreligions.com/the-wheel-of-life-4123213>. [Accessed: 29-Sep-2022].
- [11] Minimalismco and Minimalism, “The wheel of life as a self-improvement tool: Plus blank PDF & samples,” *Minimalism*, 04-Aug-2020. [Online]. Available: <https://minimalism.co/articles/wheel-of-life>. [Accessed: 29-Sep-2022].
- [12] ‘Lean Coaching: The Wheel of Life and the Harada Method [A New Approach from the ‘as is’ to the ‘to be’],” *eirai.org*. 2017.
- [13] ‘How balanced is your life? get your free online assessment at wheeloflife.noomii.com,” *Wheel of Life*. [Online]. Available: <https://wheeloflife.noomii.com/>. [Accessed: 29-Sep-2022].
- [14] D. Boud, “Using journal writing to enhance reflective practice,” *New Directions for Adult and Continuing Education*, vol. 2001, no. 90, p. 9, 2001.
- [15] “Journaling for mental health,” *Journaling for Mental Health - Health Encyclopedia - University of Rochester Medical Center*. [Online]. Available: <https://www.urmc.rochester.edu/encyclopedia/content.aspx?ContentTypeID=1&ContentID=4552>. [Accessed: 15-Sep-2022].
- [16] C. Pastore, “Stress management in college students: why journaling is the most effective technique for this demographic,” thesis, 2020.
- [17] I. S. H. W. A. R. E. E. NABAR, S. H. R. U. T. I. BAIKAR, and P. U. S. H. P. A. MAHAPATRO, “Online journaling for self-awareness of mental health,” 09-Jul-2022. [Online]. Available: https://www.mililink.com/upload/article/166639254aams_vol_219_july_2022_a16_p50_41-5052_ishwaree_nabar,_shruti_baikar_and_pushpa_mahapatro.pdf. [Accessed: 16-Sep-2022].
- [18] R. Goluboff, "Does Prompted or Unprompted Journaling Have an Impact on Positive and Negative Mood?"
- [19] C. BasuMallick, “What is gamification? definition, software, examples, and best practices 2022,” *Spiceworks*, 26-Aug-2022. [Online]. Available: <https://www.spiceworks.com/tech/devops/articles/what-is-gamification/>. [Accessed: 18-Sep-2022].
- [20] T. Mitra and K. Ozbek, “Ranking by weighted sum,” *Economic Theory*, vol. 72, no. 2, pp. 511–532, 2020.



- [21] E. Roszkowska, “Rank ordering criteria weighting methods – a comparative overview,” *Optimum. Studia Ekonomiczne*, no. 5(65), pp. 14–33, 2013
- [22] ‘Multi-criteria Decision making (MCDM)¶,’ *pymoo*. [Online]. Available: <https://pymoo.org/mcdm/index.html>. [Accessed: 16-Sep-2022].
- [23] M. Mayank, “Ranking algorithms - know your multi-criteria decision solving techniques!,” *Medium*, 19-Mar-2022. [Online]. Available: <https://towardsdatascience.com/ranking-algorithms-know-your-multi-criteria-decision-solving-techniques-20949198f23e>. [Accessed: 20-Sep-2022].
- [24] O. Stepanov, “LifeWheel Balance—Goal Setting,” *lifewheel*. 25-Oct-2021
- [25] “Happy: Wheel of Life,” *makemehappy*. 07-Jul-2021
- [26] “The Fabulous - Build Better Habits & Achieve Your Goals,” *The Fabulous - Build better habits & achieve your goals*. [Online]. Available: <https://www.thefabulous.co/>. [Accessed: 15-Sep-2022].
- [27] K. Rees, “The 8 best features of fabulous app to build healthy habits and boost productivity,” *MUO*, 07-Jun-2021. [Online]. Available: <https://www.makeuseof.com/best-features-fabulous-app-habits-productivity/>. [Accessed: 15-Sep-2022].
- [28] Appadvice, “Ambitionist: Life goal tracker by Maxim Novoseltsev,” *AppAdvice*, 28-Sep-2022. [Online]. Available: <https://appadvice.com/app/ambitionist-life-goal-tracker/1577945954>. [Accessed: 15-Sep-2022].
- [29] I. Success Wizard, “Goal Tracker & Things Done,” App Store, 31-Jul-2014. [Online]. Available: <https://apps.apple.com/ne/app/goals-wizard-premium/id901800555>. [Accessed: 17-Sep-2022].
- [30] the Mind Tools Content Team By the Mind Tools Content Team, the M. T. C. Team, M. wrote, K. wrote, and C. K. C. wrote, “To-do lists: The key to efficiency,” *Time Management Training from MindTools.com*. [Online]. Available: https://www.mindtools.com/pages/article/newHTE_05.htm#:~:text=By%20keeping%20such%20a%20list,you%20can%20leave%20until%20later. [Accessed: 24-Sep-2022].
- [31] “The 7 most popular IOS architecture patterns explained,” *The 7 Most Popular iOS Architecture Patterns Explained - Zero To App Store*. [Online]. Available: <https://www.zerotoappstore.com/the-most-popular-ios-architecture-patterns-explained.html>. [Accessed: 26-Sep-2022].
- [32] M. Razan “MVC Architecture” *Medium*, 4-Dec-2020. [Online]. Available: <https://medium.com/linkit-intecs/mvc-architecture-14238e8b93bc>. [Accessed: 6-Nov-2022].



- [33] Y. Kumar, “Design patterns in flutter part 1,” *Medium*, 04-May-2020. [Online]. Available: <https://medium.flutterdevs.com/design-patterns-in-flutter-part-1-c32a3ddb00e2>. [Accessed: 26-Sep-2022].
- [34] J. Mifsud, “Usability metrics - a guide to quantify the usability of any system,” *Usability Geek*, 13-Sep-2019. [Online]. Available: <https://usabilitygeek.com/usability-metrics-a-guide-to-quantify-system-usability/>. [Accessed: 29-Sep-2022].
- [35] J. 8 and J. Hertvik, “Service availability: Calculations and metrics, five 9s, and best practices,” *BMC Blogs*, 08-Jul-2020. [Online]. Available: <https://www.bmc.com/blogs/service-availability-calculation-metrics>. [Accessed: 29-Sep-2022].
- [36] Vahid Dejwakh, “Understanding the model-view-controller (MVC) pattern,” *Vahid Dejwakh*, 16-Apr-2021. [Online]. Available: <https://vahid.blog/post/2021-04-16-understanding-the-model-view-controller-mvc-pattern/>. [Accessed: 05-Nov-2022].
- [37] “Isar: Dart package,” *Dart packages*, 03-Oct-2022. [Online]. Available: <https://pub.dev/packages/isar>. [Accessed: 05-Nov-2022].
- [38] “Home · Atina Diffley,” *Atina Diffley*, 09-Jul-2021. [Online]. Available: <https://atinadiffley.com/>. [Accessed: 05-Nov-2022].
- [39] “Firebase.Flutter” *Flutterfire overview: Flutterfire (2021) FlutterFire Blog RSS*. Available at: <https://firebase.flutter.dev/docs/overview/> [Accessed: 29-Oct-2022].
- [40] Pinkiewicz, M. *Is firebase the best back-end choice for a flutter app?*, *Digital Solutions for Healthcare*. Brandmed. Available at: <https://brandmed.com/blog/development/is-firebase-the-best-back-end-choice-for-a-flutter-app#:~:text=One%20of%20the%20most%20reliable,Firebase%20for%20the%20back%2Dend>. [Accessed: 28-Oct-2022].
- [41] Bayat, H.B. (2022) *A minimalist guide to isar - medium*. Available at: <https://itnext.io/a-minimalist-guide-to-isar-ee43c1e51a85> [Accessed: 01-Nov-2022].
- [42] Aela, E. (2022) *Nielsen's heuristics: 10 usability principles to improve UI design, Nielsen's Heuristics: 10 Usability Principles To Improve UI Design*. Available at: <https://aelaschool.com/en/interactiondesign/10-usability-heuristics-ui-design/> [Accessed: 01-Nov-2022].
- [43] 2012T1 fortune cookies: User testing (no date) *2012T1 Fortune Cookies: User Testing - IS480*. Available at: https://wiki.smu.edu.sg/is480/2012T1_Fortune_Cookies:_User_Testing#User_Acceptance_Test . [Accessed: 05-Nov-2022].



- [44] *Vibrant society* (2020) *Vision 2030*. Available at:
<https://www.vision2030.gov.sa/v2030/overview/vibrant-society/>. [Accessed: 05-Nov-2022].
- [45] Damaschino, E. and Redmond, W. (2008) *Survey shows increasing worldwide reliance on to-do lists, Stories*. Available at:
<https://news.microsoft.com/2008/01/14/survey-shows-increasing-worldwide-reliance-on-to-do-lists/>. [Accessed: 05-Nov-2022].
- [46] Gerry L. Watson, M.S.W. (2022) *The daily to do list system for ADHD brains, ADDitude*. ADDitude. Available at: <https://www.additudemag.com/daily-to-do-lists-that-work/>. [Accessed: 05-Nov-2022].
- [47] *Firebase Support* (2020). *Service Level Agreement for Hosting and Realtime Database*, Available at: firebase.google.com/terms/service-level-agreement. [Accessed: 13-May-2023]



10 Appendix A

10.1 Questionnaire's Questions

ما مدى اتفاقك مع العبارات التالية:

- 1- أواجه صعوبة في تحديد أهدافي بطريقة فعالة.
- 2- أواجه صعوبات في تبني عادات جديدة.
- 3- أواجه صعوبة في ترتيب أولويات مهامي اليومية.
- 4- أعاني من تأجيل المهام والتسويف.
- 5- أعتمد على أداة للتنكير بمهامي اليومية.
- 6- مشاركة الأهداف مع الآخرين يساعد على تحقيقها.
- 7- اسم تطبيق إدارة المهام إن وجد (سؤال مفتوح):
- 8- هل سبق لك تجربة التدوين اليومي؟
- 9- إذا أردت تدوين يومياتك إلى أي مدى تفضل استخدام الأسئلة والجمل المفتاحية كمحفز للتدوين؟
- 10- برأيك كم عدد المهام التي تستطيع إنجازها في اليوم دون الشعور بالإنهاك أو التوتر؟
- 11- عند وضعك لأهدافك ما هي المعلومات التي تحرض على تضمينها حول الهدف؟
- 12- هل سبق لك تجربة تطبيق لمتابعة العادات اليومية باللغة العربية (سؤال مفتوح)؟

10.2 Questionnaire's responses

Table9 Appendix Questionnaire's response

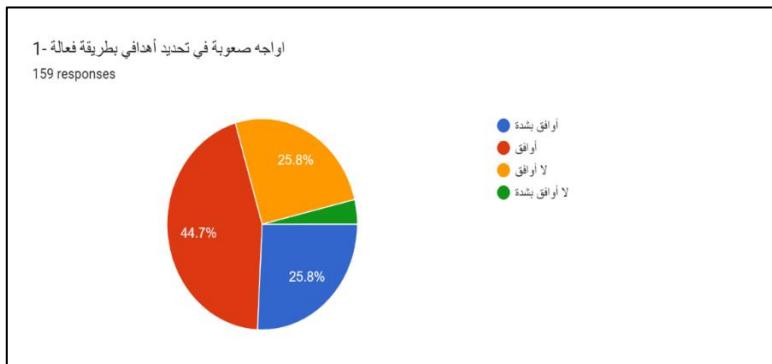


Figure 60 Questionnaire Q1

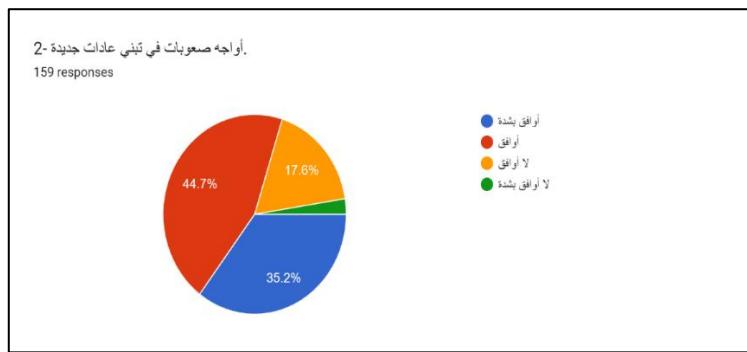


Figure 61 Questionnaire Q2

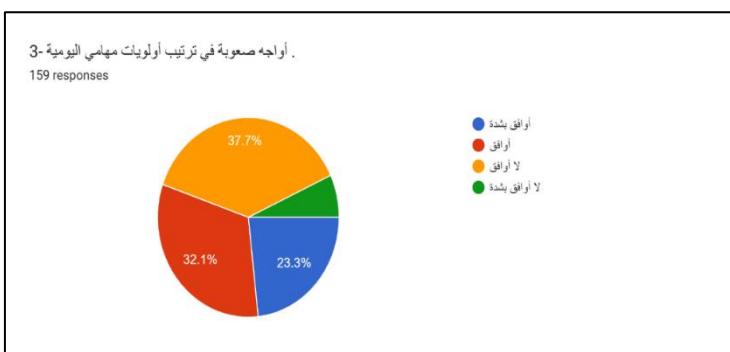


Figure 62 Questionnaire Q3

أعاني من تأجيل المهام والتسويف -4-

159 responses

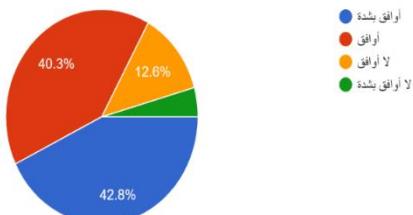


Figure 63 Questionnaire Q4

أعتمد على آدلة للتذكير بمهامي اليومية -5-

159 responses

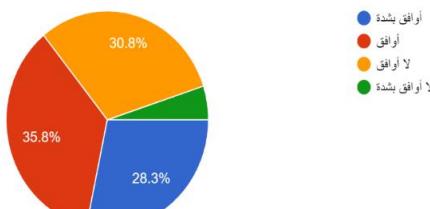


Figure 64 Questionnaire Q5

مشاركة الأهداف مع الآخرين يساعد على تحقيقها-6.

159 responses

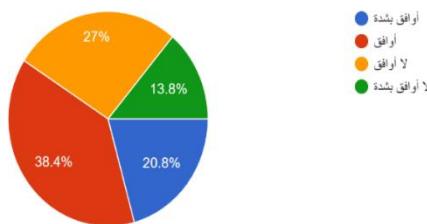


Figure 65 Questionnaire Q6

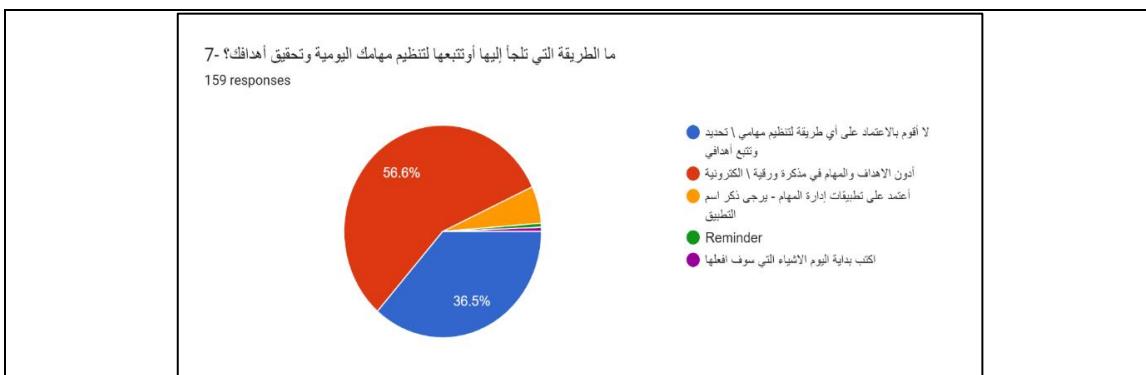


Figure 66 Questionnaire Q7

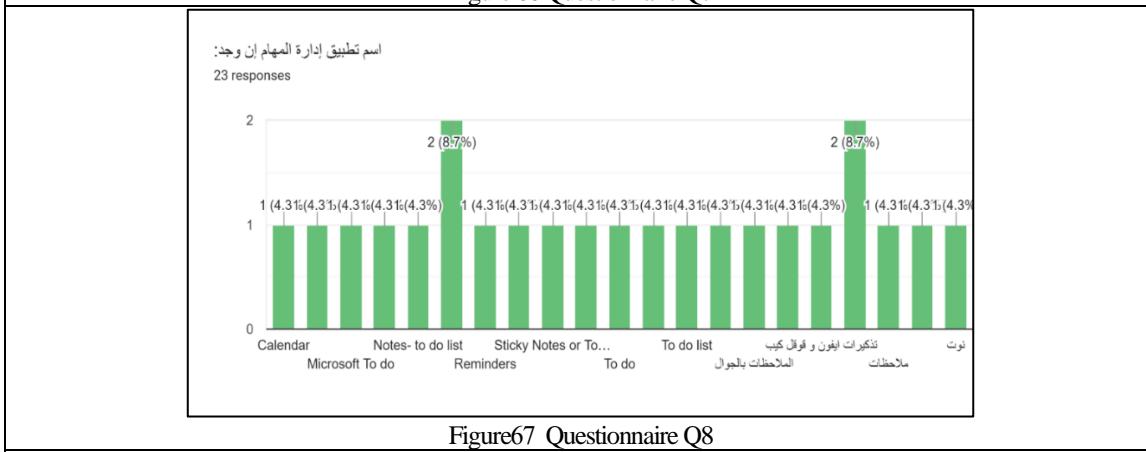


Figure 67 Questionnaire Q8

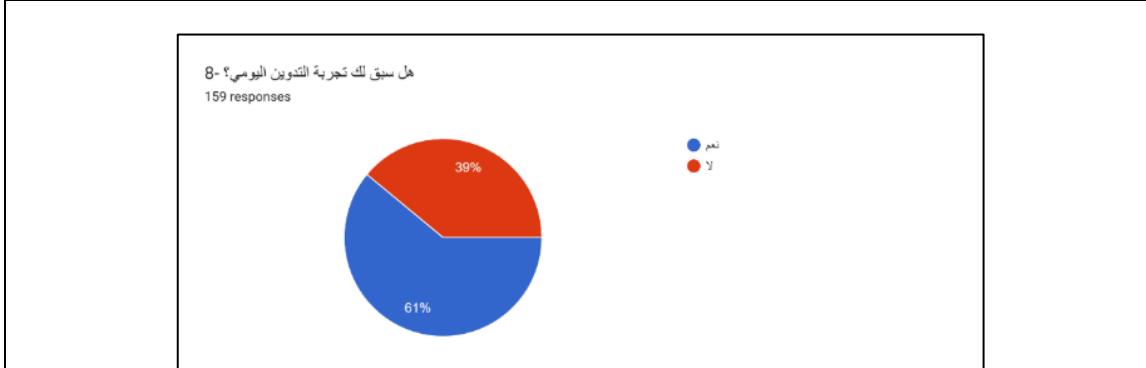


Figure Error! No text of specified style in document.9 Questionnaire Q9

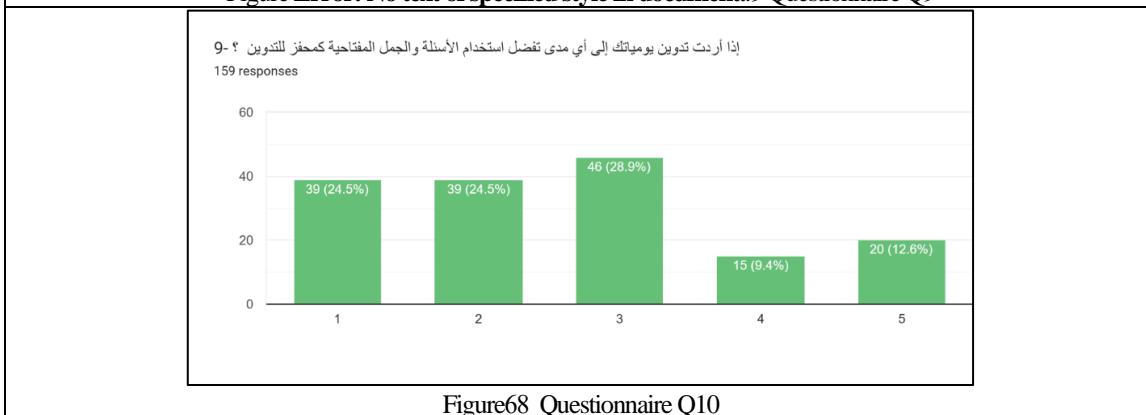


Figure 68 Questionnaire Q10

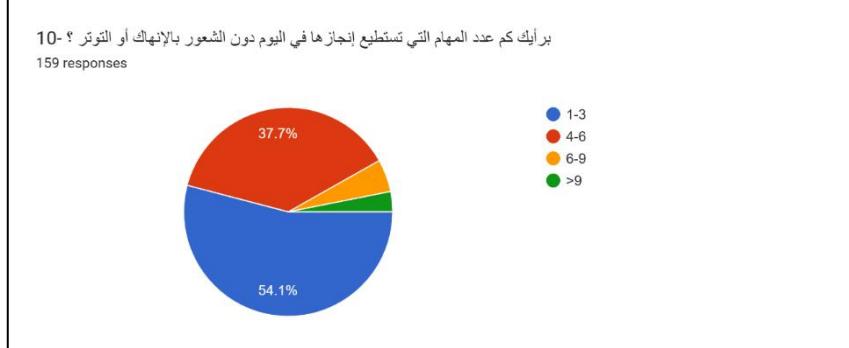


Figure69 Questionnaire Q11

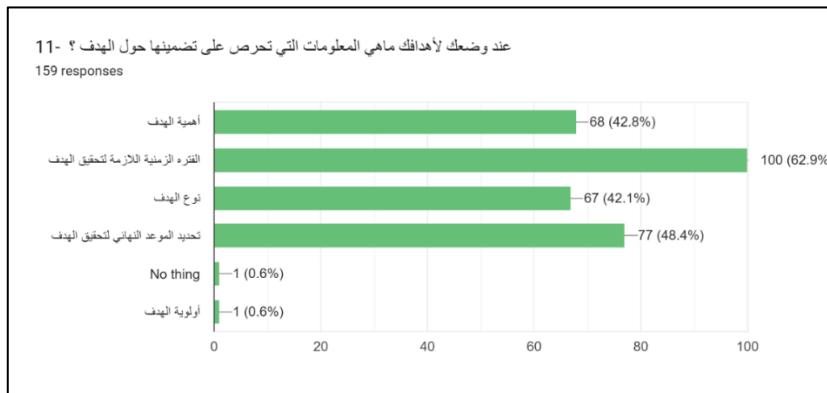


Figure70 Questionnaire Q12

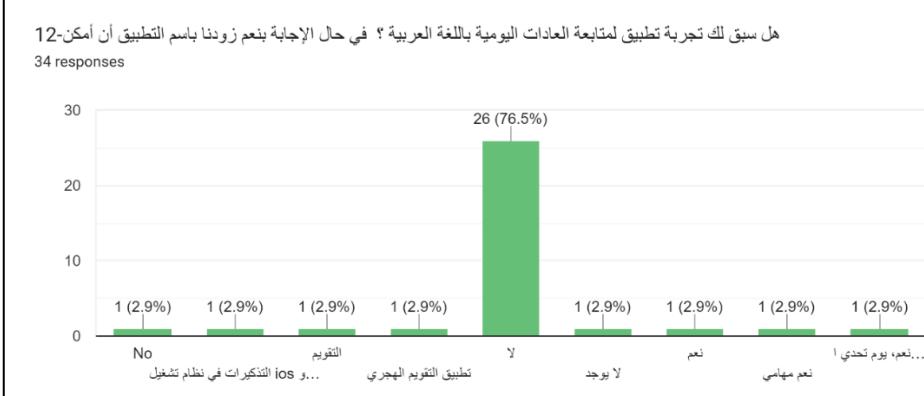


Figure71 Questionnaire Q13

11 Appendix B

11.1 Usability Testing Questions

Strongly Agree 5	Agree 4	Neutral 3	Disagree 2	Strongly Disagree 1
-----------------------------------	--------------------------	----------------------------	-----------------------------	--------------------------------------

Overall System:

1. I found the system's interface an easy way to perform system functions.
2. I found the error messages in the application are clear and indicate the details of the error.
3. I felt the system responded in a consistent and predictable way.
4. I found the terminology used to be clear and precise.
5. I always knew where to access the information I wanted.
6. I get enough feedback after adding/editing/deleting a goal/habit/task.
7. I journal my daily thoughts with ease.
8. I can search for and join communities easily.
9. I exchange messages and pictures with members in communities.
10. I can create communities.
11. I can invite friends to join my communities

System Interface:

1. The screen layout was clear and helpful.
2. The amount of information displayed on screen was sufficient.
3. The sequence of screens was predictable and made sense.
4. The interface design was friendly and aesthetically pleasing.

Learning to operate the system:

1. I needed help getting started with the system.
2. I had to remember names and use of commands.
3. I had to remember specific rules about clicking buttons.
4. The tasks were performed in a straightforward manner.

12 Appendix C

12.1 Assessment Questions

عائلتي واصدقائي

1. لدى عدد كافٍ من الأصدقاء الرائعين.
2. أثق في العلاقات التي تربطني بأصدقائي.
3. صداقاتي تدعوني وتغذيني.
4. أنا صديق جيد وأتفرق لصديقاتي
5. لا أشعر بضرورة الإخفاء أو الامتناع في علاقاتي مع أفراد الأسرة
6. لقد خلقت تجربة الأسرة في حياتي، سواء كانت مع أقارب البيولوجيين أم لا.
7. أناراض عن الدور الذي ألعبه ومستوى مساهمنتي في عائلتي
8. أشعر بالرضا عن مساهماتي تجاه عائلتي
9. أتواصل مع عائلتي بشكل مستمر

علاقاني

1. أنا منفتح لخلق علاقة حب حميمة.
2. أنا متحرر من الاستياء الماضي أو اللوم من ناحية العلاقات الحميمية.
3. أنا على استعداد للمخاطرة بنفسي من أجل العلاقة الحميمية.
4. أنا أصنع الرومانسية في حياتي.

ذاتي

1. لدى اعتقاد ذاتي يساندني ويدعمني بالرغم من التحديات التي أوجهها في حياتي.
2. أنا منخرط في حياتي وأتعامل مع كل يوم كمغامرة.
3. أحب أن أعيش حياتي بانتظام وأحب ما أنا عليه.
4. أشارك بشكل منتظم بالأنشطة التي تبني ذاتي وتصقل مهاراتي.

صحي

1. أتعامل مع صحتي بطريقة استباقية وبنظام غذائي مُخطط، بدلاً من الاعتماد على النظام الذي يدير صحتي في وقت الأزمات فقط.
2. أنا راضٍ عن مستوى نشاطي وحيويتي
3. ألتزم بأنظمة تسمح لي بالحفاظ على صحتي ورفاهيتي بسهولة.
4. أنا مدرك لمستوى لياقة جسمي ولذلك أتحمل مسؤولية صحتي البدنية.
5. لدى معرفة بالعادات التي تحافظ على صحتي وألتزم بها



اموال

1. لدى ما يكفي من المال لفعل ما أريد وإنجاز أعمالي المهمة.
2. أدير شؤوني وسجلاتي المالية بشكل جيد.
3. لا أعاني من القلق والتوتر بخصوص وضعي المالي.
4. مستقبلي المالي يبدوا بوضع جيد ومستدام.

بيئتي

- 1.أشعر بالدعم والراحة في منزلي.
2. أنا محاط بالأشياء التي أحبها وتعني لي الكثير.
3. مستوى التنظيم في محيط حياتي يناسب حاجتي (محيطي يخدمني).
4. خزانة ملابسي تعبر بوضوح شخصيتي. أنا أفتخر بما أرتدي

مهني

1. أحب عملي
2. أشعر بان مهاراتي وقدراتي يستفاد منها بشكل جيد في مهنتي
3. أنا مستمتع بيئه عملى والأشخاص الذين أعمل معهم
4. ارى فرصة للنمو والتطور في عملى
5. أشعر بأننى وجدت العمل المناسب لي

متعي

1. استقطع بانتظام وقتاً للعب والمغامرة والتجارب
2. أعرف ما الأنشطة التي تجدد نشاطي وأشارك فيها بانتظام
3. دوماً أخصص فترة كافية من وقتي للاسترخاء والاستمتاع مع الآخرين
4. أصنع المرح والسعادة لي وللآخرين



13 Appendix D

13.1 Ranking Algorithm

The tasks in the to-do list are ranked using the following algorithm: each task is assigned a number (rank) in the range [0,1] where larger number indicates a higher rank/ more important task. To do so, we consider three factors: importance, time, dependency.

Each factor has a term in the equation. Each term must have a number in the range [0,1], as follows:

Importance: the higher the importance, the higher the rank

Time: the less days left, the higher the rank

Dependency: the more tasks dependent on the task, the higher the rank

$$Rank_i = w_1 * NT_i + w_2 * importance_i + w_3 * dependency_i$$

- NT_i is calculated as follows:

For each task i for a user calculate the following

$$T_i = \frac{TaskDuration - DaysCompleted_Task}{num\ days\ left\ to\ goal\ due\ date}$$

Normalize the value by dividing over the maximum T over all tasks

$$NT_i = \frac{T_i}{maxT}$$

Where maxT is the maximum Ti over all tasks

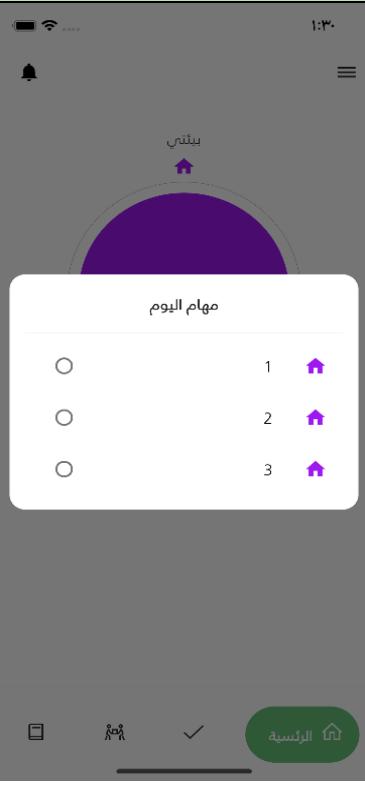
- Dependency is calculated as follows:

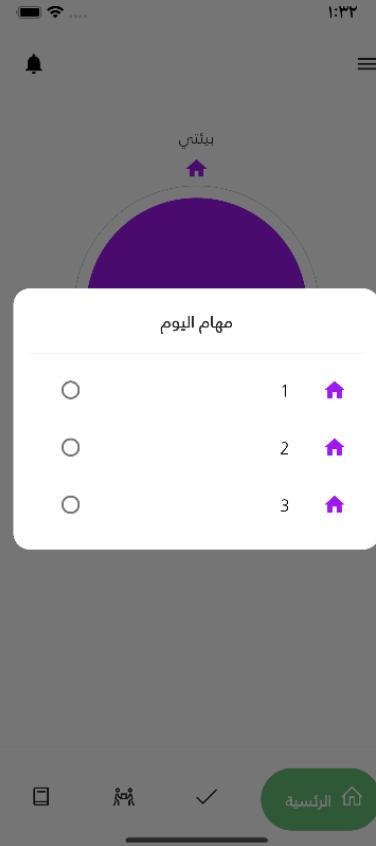
$$Dependency_i = \frac{num\ of\ tasks\ dependent\ on\ task\ i}{total\ num\ of\ tasks}$$

Num of tasks dependent on task i directly or indirectly can be calculated by creating DAG where each node represents a task and finding the number of incoming directed edges.

As for the weights, they must be in the range [0,1] and their sum must be equal to 1. The default value for all weights is 0.33, as we consider that all terms in the equation are equally important, however, users can edit the weights according to their needs.

13.2 Ranking Test Cases

Test case	Variable values	Expected result	Actual Result	Pass?	Comments
1. Adding an item with high Importance, and all other list items are of low importance	<p>Importance: Item1 = 0.75 Items2&3 = 0.25</p> <p>Due date: Items 1&2&3 = due four days from current date</p> <p>Duration: Items 1&2&3 = 4</p> <p>daysCompleted: Items 1&2&3 = 0</p> <p>Dependency: : Items 1&2&3 = 0</p>	$\text{RankItem1} = 0.33 * 1 + 0.33 * 0.75 + 0.33 * 0 = 0.5775$ $\text{RankItems2\&3} = 0.33 * 1 + 0.33 * 0.25 + 0.33 * 0 = 0.4125$ <p>Thus, the order of the tasks should be: Item1 Item2 Item3</p>		pass	Note: items with equal rank are displayed by the order of creation
2. Adding an item with high Importance, and all other list items are of medium importance.	<p>Importance: Item1 = 0.75 Items2&3 = 0.5</p> <p>Due date: Items 1&2&3 = due four days from current date</p> <p>Duration: Items 1&2&3 = 4</p> <p>daysCompleted: Items 1&2&3 = 0</p>	<ul style="list-style-type: none"> $\text{RankItem1} = 0.33 * 1 + 0.33 * 0.75 + 0.33 * 0 = 0.5775$ $\text{RankItems2\&3} = 0.33 * 1 + 0.33 * 0.5 + 0.33 * 0 = 0.4950$ <p>Thus, the order of the tasks should be: Item1 Item2 Item3</p>		pass	No comments

	<p><u>Dependency</u> : Items 1& 2&3 = 0</p>		
4. Adding an item with high Importance, and all other list items are of high importance	<p><u>Importance:</u> Items 1&2&3 = 0.75</p> <p><u>Due date:</u> Items 1& 2&3 = due four days from current date</p> <p><u>Duration:</u> Items 1&2&3 = 4</p> <p><u>daysCompleted:</u> Items 1&2&3 = 0</p> <p><u>Dependency</u> : Items 1&2&3 = 0</p>	<ul style="list-style-type: none"> Rank Items 1&2&3 = $0.33 * 1 + 0.33 * 0.75 + 0.33 * 0 = 0.5775$ <p>Thus, the order of the tasks should be: Item1 Item2 Item3</p> 	pass No comment

5. Adding an item with high Importance, and other list items have either medium and low importance	Importance: Item1 = 0.75 Item2 = 0.25 Item3 = 0.5 Due date: Items 1& 2&3 = due four days from current date Duration: Items 1& 2&3 = 4 daysCompleted: Items 1& 2&3 = 0 Dependency: : Items 1& 2&3 = 0	<ul style="list-style-type: none"> Rank_{Item1}=0.33*1+0. 33*0.75+0.33*0 = 0.5775 Rank_{Item2}=0.33*1+0. 33*0.25+0.33*0 = 0.4125 Rank_{Item3}=0.33*1+0. 33*0.5+0.33*0 = 0.4950 <p>Thus, the order of the tasks should be:</p> <p>Item1 Item3 Item2</p>		pass	No comments
6. Adding an item with high Importance, and all other list items have either high, medium, and low importance	Importance: Item1&4 = 0.75 Item2 = 0.25 Item3 = 0.5 Due date: Items 1& 2&3&4 = due four days from current date Duration: Items 1& 2&3&4 = 4 daysCompleted: Items 1& 2&3&4 = 0 Dependency: : Items 1& 2&3&4 = 0	<ul style="list-style-type: none"> Rank_{Items1&4}=0.33*1+0. 0.33*0.75+0.33*0 = 0.5775 Rank_{Item2}=0.33*1+0. 33*0.25+0.33*0 = 0.4125 Rank_{Item3}=0.33*1+0. 33*0.5+0.33*0 = 0.4950 <p>Thus, the order of the tasks should be:</p> <p>Item1 Item4 Item3 Item2</p>		pass	No comments
7. Adding an item that is due in 1 day, and all other	Importance: Items1&2&3 = 0.5 Due date:	<ul style="list-style-type: none"> Rank_{Item1}=0.33*1+0. 33*0.5+0.33*0     Item1=0.33*1+0.33*0.5+0.33*0 		pass	No comments

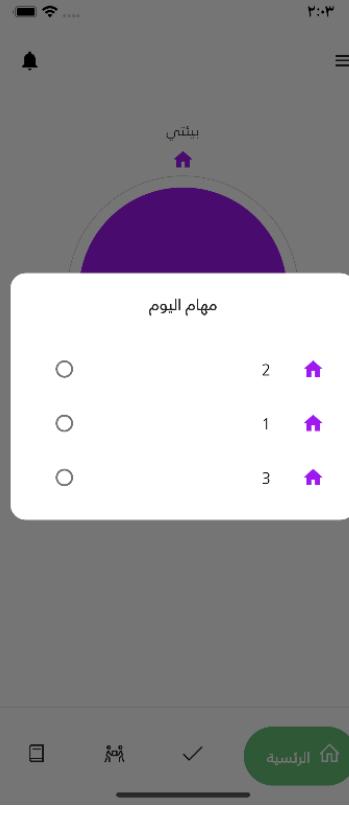
list items are due in 4 days	Item 1 = due one day from current date Items 2&3 = due four days from current date Duration: Items 1&2&3 = 1 daysCompleted: Items 1&2&3 = 0 Dependency : Items 1&2&3 = 0	$\begin{aligned} &= \mathbf{0.4950} \\ &\bullet \text{ Rank}_{\text{Items2\&3}} = 0.33 * 0.25 + 0.33 * 0.5 + 0.33 * 0 \\ &= \mathbf{0.2475} \end{aligned}$ <p>Thus, the order of the tasks should be:</p> <p>Item1 Item2 Item3</p>			
8. Adding an item that is due in 1 day, and all other list items are due in 1 day	Importance: Items 1&2&3 = 0.5 Due date: Items 1&2&3 = due one day from current date Duration: Items 1&2&3 = 1 daysCompleted: Items 1&2&3 = 0 Dependency : Items 1&2&3 = 0	$\bullet \text{ Rank}_{\text{Items1\&2\&3}} = 0.33 * 1 + 0.33 * 0.5 + 0.33 * 0 \\ = \mathbf{0.4950}$ <p>Thus, the order of the tasks should be:</p> <p>Item1 Item2 Item3</p>		pass	No comments
	Importance:				No comments

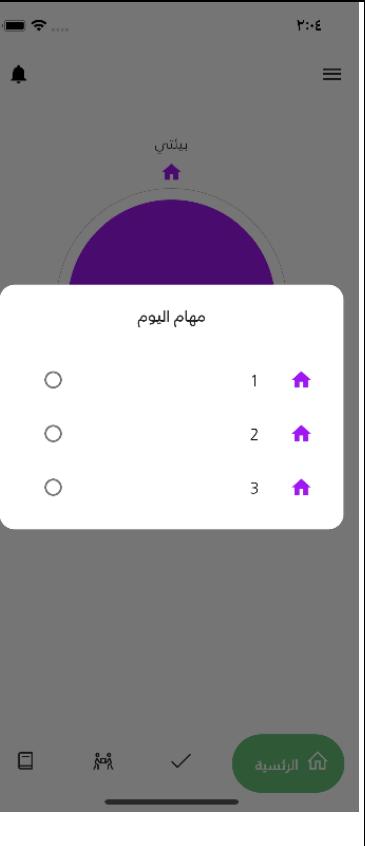
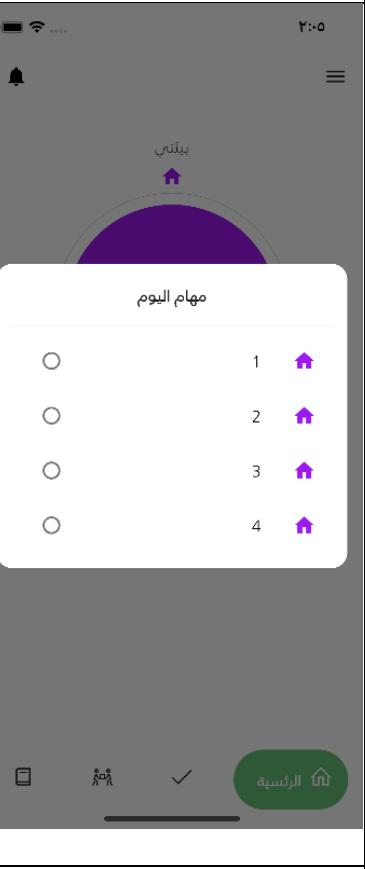
9. Adding an item that is due in 4 days, and all other list items are due in 1 day	<table border="1"> <tr> <td>Items 1&2&3 = 0.5</td></tr> <tr> <td>Due date: Item 1 = due four days from current date</td></tr> <tr> <td>Items 2&3 = due one day from current date</td></tr> <tr> <td>Duration: Items 1&2&3 = 1</td></tr> <tr> <td>daysCompleted: Items 1&2&3 = 0</td></tr> <tr> <td>Dependency : Items 1&2&3 = 0</td></tr> </table>	Items 1&2&3 = 0.5	Due date: Item 1 = due four days from current date	Items 2&3 = due one day from current date	Duration: Items 1&2&3 = 1	daysCompleted: Items 1&2&3 = 0	Dependency : Items 1&2&3 = 0	<ul style="list-style-type: none"> $Rank_{Item1} = 0.33 * 0.25 + 0.33 * 0.5 + 0.33 * 0 = 0.2475$ $Rank_{Items2&3} = 0.33 * 1 + 0.33 * 0.5 + 0.33 * 0 \quad \text{????}$ $\text{?} Items2&3 = 0.33 * 1 + 0.33 * 0.5 + 0.33 * 0 = 0.4950$ <p>Thus, the order of the tasks should be: Item2 Item3 Item1</p>		pass	
Items 1&2&3 = 0.5											
Due date: Item 1 = due four days from current date											
Items 2&3 = due one day from current date											
Duration: Items 1&2&3 = 1											
daysCompleted: Items 1&2&3 = 0											
Dependency : Items 1&2&3 = 0											
10. Adding an item that is due in 1 day, an item that is due in 2 days, and an item that is due in 4 days	<table border="1"> <tr> <td>Importance: Items 1&2&3 = 0.5</td> </tr> <tr> <td>Due date: Item 1 = due one day from current date</td> </tr> <tr> <td>Item 2 = due two days from current date</td> </tr> <tr> <td>Item 3 = due four days from current date</td> </tr> <tr> <td>Duration: Items 1&2&3 = 1</td> </tr> <tr> <td>daysCompleted: Items 1&2&3 = 0</td> </tr> <tr> <td>Dependency : Items 1&2&3 = 0</td> </tr> </table>	Importance: Items 1&2&3 = 0.5	Due date: Item 1 = due one day from current date	Item 2 = due two days from current date	Item 3 = due four days from current date	Duration: Items 1&2&3 = 1	daysCompleted: Items 1&2&3 = 0	Dependency : Items 1&2&3 = 0	<ul style="list-style-type: none"> $Rank_{Item1} = 0.33 * 1 + 0.33 * 0.5 + 0.33 * 0 = 0.4950$ $Rank_{Item2} = 0.33 * 0.5 + 0.33 * 0.5 + 0.33 * 0 = 0.3300$ $Rank_{Item3} = 0.33 * 0.25 + 0.33 * 0.5 + 0.33 * 0 = 0.2475$ <p>Thus, the order of the tasks should be: Item1 Item2 Item3</p>		pass No comments
Importance: Items 1&2&3 = 0.5											
Due date: Item 1 = due one day from current date											
Item 2 = due two days from current date											
Item 3 = due four days from current date											
Duration: Items 1&2&3 = 1											
daysCompleted: Items 1&2&3 = 0											
Dependency : Items 1&2&3 = 0											

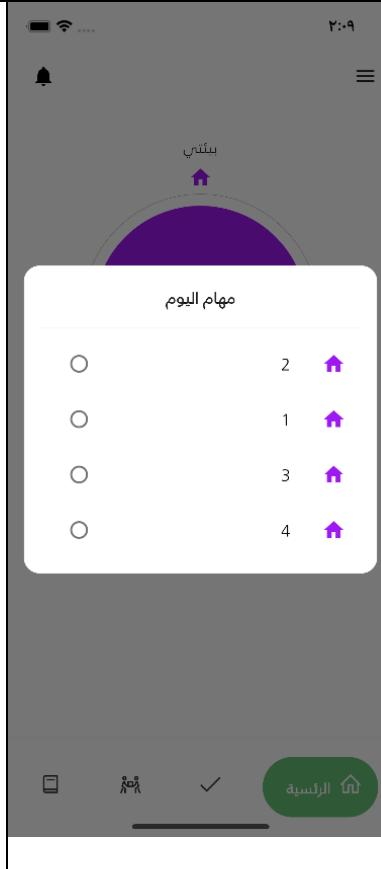
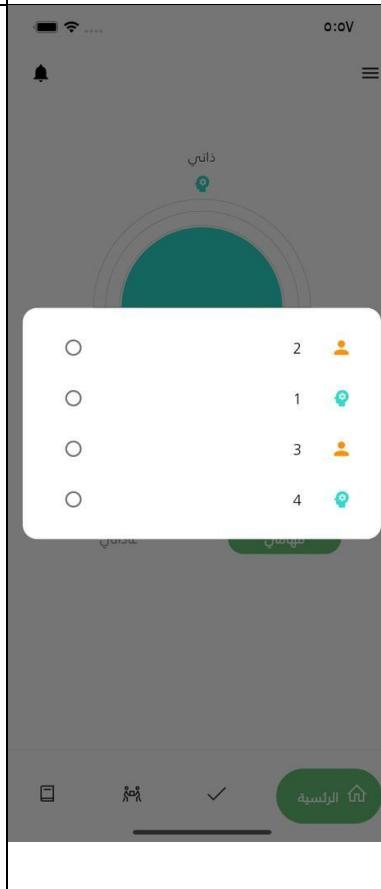
11. Adding an item with 1 day duration, and all other items have 4-day duration	Importance: Items 1&2&3 = 0.5 Due date: Item 1&2&3 = due four days from current date Duration: Item 1 = 1 Items 2&3 = 4 daysCompleted: Items 1&2&3 = 0 Dependency: : Items 1&2&3 = 0	<ul style="list-style-type: none"> $Rank_{Item1} = 0.33 * 0.25 + 0.33 * 0.5 + 0.33 * 0 = 0.2475$ $Rank_{Items2&3} = 0.33 * 1 + 0.33 * 0.5 + 0.33 * 0 = 0.4950$ <p>Thus, the order of the tasks should be:</p> <p>Item2 Item3 Item1</p>		pass	No comments
12. Adding an item with 4-day duration, and all other items have 4-day duration	Importance: Items 1&2&3 = 0.5 Due date: Item 1&2&3 = due four days from current date Duration: Items 1&2&3 = 4 daysCompleted: Items 1&2&3 = 0 Dependency: : Items 1&2&3 = 0	<ul style="list-style-type: none"> $Rank_{Item1&2&3} = 0.33 * 1 + 0.33 * 0.5 + 0.33 * 0 = 0.4950$ <p>Thus, the order of the tasks should be:</p> <p>Item1 Item2 Item3</p>		pass	No comments
13. Adding an item with 4-day duration	Importance: Items 1&2&3 = 0.5	<ul style="list-style-type: none"> $Rank_{Item1} = 0.33 * 1 + 0.33 * 0.5 + 0.33 * 0 \diamond\# \diamond\# \diamond\# It$ 		pass	No comments

duration, and all other items have 1-day duration	Due date: Item 1&2&3 = due four days from current date Duration: Item 1 = 4 Items 2&3 = 1 daysCompleted: Items 1&2&3 = 0 Dependency: : Items 1&2&3 = 0	$\begin{aligned} em1 &= 0.33 * 1 + 0.33 * 0.5 + 0.33 * 0 \\ &= \mathbf{0.4950} \\ \bullet \quad Rank_{Items2\&3} &= 0.33 * 0.25 + 0.33 * 0.5 + 0.33 * 0 \\ &= \mathbf{0.2475} \end{aligned}$ <p>Thus, the order of the tasks should be:</p> <p>Item1 Item2 Item3</p>		
13.Completing (checking) an item, and all other items are incomplete (unchecked)	Importance: Items 1&2&3 = 0.5 Due date: Item 1&2&3 = due four days from current date Duration: Items 1&2&3 = 4 daysCompleted: Item 1 = 1 Items 2&3 = 0 Dependency: : Items 1&2&3 = 0	$\begin{aligned} \bullet \quad Rank_{Item1} &= 0.33 * 0.75 + 0.33 * 0.5 + 0.33 * 0 \quad \text{◆◆◆} \\ \text{◆◆} \quad Item1 &= 0.33 * 0.75 + 0.33 * 0.5 + 0.33 * 0 \\ &= \mathbf{0.4125} \\ \bullet \quad Rank_{Items2\&3} &= 0.33 * 1 + 0.33 * 0.5 + 0.33 * 0 \\ &= \mathbf{0.4950} \end{aligned}$ <p>Thus, the order of the tasks should be:</p> <p>Item2 Item3 Item1</p>		<p>Note: For enhancing the usability and user experience, completed items are moved to the bottom of the list.</p> <p>pass</p>
	Importance:			

15. Uncheck an item, and all other items are checked	Items 1&2&3 = 0.5 Due date: Item 1&2&3 = due four days from current date Duration: Items 1&2&3 = 4 daysCompleted: Items 1&2 = 1 Item 3 = 0 Dependency : Items 1&2&3 = 0	<ul style="list-style-type: none"> Rank_{Items1&2}=$0.33*0.5+0.33*0.5=0.4125$ Rank_{Item3}=$0.33*1+0.33*0=0.33$ <p>Thus, the order of the tasks should be:</p> <p>Item3 Item1 Item2</p>		pass
15. Checking an item, and all other items are checked	Importance: Items 1&2&3 = 0.5 Due date: Item 1&2&3 = due four days from current date Duration: Items 1&2&3 = 4 daysCompleted: Items 1&2&3 = 1 Dependency : Items 1&2&3 = 0	<ul style="list-style-type: none"> Rank_{Items1&2&3}=$0.33*1+0.33*0.5+0.33*0=0.4950$ <p>Thus, the order of the tasks should be:</p> <p>Item1 Item2 Item3</p>		pass

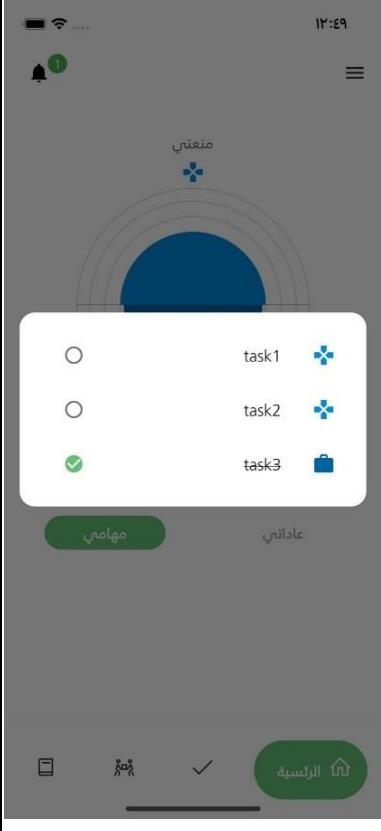
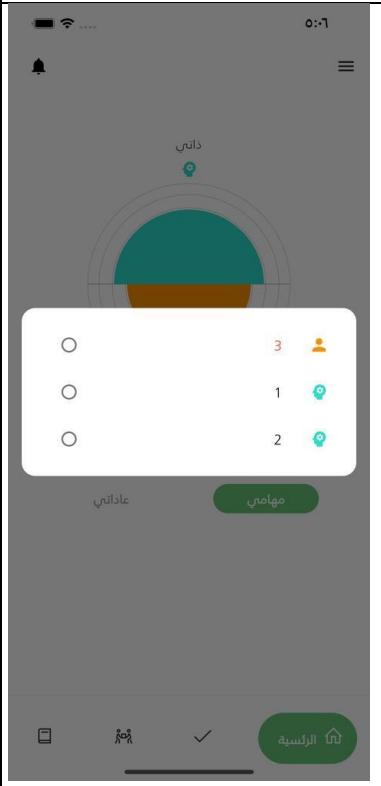
16. Uncheckin g an item, and all other items are unchecked	Importance: Items 1&2& $3 = 0.5$ Due date: Item 1&2&3 = due four days from current date Duration: Items $1 \& 2 \& 3 = 4$ daysComple ted: Items $1 \& 2 \& 3 = 0$ Dependency : Items $1 \& 2 \& 3 = 0$	<ul style="list-style-type: none"> $Rank_{Items1\&2\&3} = 0.33 * 1 + 0.33 * 0.5 + 0.33 * 0 = 0.4950$ <p>Thus, the order of the tasks should be: Item1 Item2 Item3</p>		
18. Adding an item with 1 dependency, and all other items have 0 dependencies	Importance: Items 1&2& $3 = 0.5$ Due date: Item 1&2&3 = due four days from current date Duration: Items $1 \& 2 \& 3 = 4$ daysComple ted: Items $1 \& 2 \& 3 = 0$ Dependency : Items 1&3 = 0 Item 2 = 1	<ul style="list-style-type: none"> $Rank_{Item2} = 0.33 * 1 + 0.33 * 0.5 + 0.33 * 0.33 = 0.605$ $Rank_{Items1\&3} = 0.33 * 1 + 0.33 * 0.5 + 0.33 * 0 = 0.495$ <p>Thus, the order of the tasks should be: Item2 Item1 Item3</p>		Pass No comment
	Importance:			No comment

19. Adding an item with 0 dependencies, and all other items have 1 dependency	Items 1&2&3 = 0.5 Due date: Item 1&2&3 = due four days from current date Duration: Items 1&2&3 = 4 daysCompleted: Items 1&2&3 = 0 Dependency : Items 1&2 = 1 Item 3 = 0	<ul style="list-style-type: none"> $Rank_{Item3} = 0.33 * 1 + 0.33 * 0.5 + 0.33 * 0 = 0.495$ $Rank_{Items2&1} = 0.33 * 1 + 0.33 * 0.5 + 0.33 * 0.33 = 0.605$ <p>Thus, the order of the tasks should be:</p> <p>Item1 Item2 Item3</p>		Pass	
20. Adding an item with 1 dependency, and all other items have 1 dependency	Importance: Items 1&2&3 = 0.5 Due date: Item 1&2&3 = due four days from current date Duration: Items 1&2&3 = 4 daysCompleted: Items 1&2&3 = 0 Dependency : Items 1&2&3 = 1	<ul style="list-style-type: none"> $Rank_{Item1&2&3} = 0.33 * 1 + 0.33 * 0.5 + 0.33 * 0.33 = 0.605$ <p>Thus, the order of the tasks should be:</p> <p>Item2 Item3 Item1</p>		Pass	No comment
	Importance:				No comment

20. Adding an item with 0 dependencies, an item with 1 dependency , an item with 2 dependencies, and an item with 3 dependencies	Items 1&2&3 = 0.5 Due date: Item 1&2&3 = due four days from current date Duration: Items 1&2&3 = 4 daysComPLETED: Items 1&2&3 = 0 Dependency : Item 1 = 2 Item 2 = 3 Item 3 = 1 Item 4 = 0	<ul style="list-style-type: none"> Rank_{Item1}=0.33*1+0.33*0.5+0.33*0.5 = 0.660 Rank_{Item2}=0.33*1+0.33*0.5+0.33*0.75 = 0.7425 Rank_{Item3}=0.33*1+0.33*0.5+0.33*0.25 = 0.5775 Rank_{Item4}=0.33*1+0.33*0.5+0.33*0 = 0.4950 <p>Thus, the order of the tasks should be:</p> <table style="margin-left: 100px;"> <tr><td>Item2</td></tr> <tr><td>Item1</td></tr> <tr><td>Item3</td></tr> <tr><td>Item4</td></tr> </table>	Item2	Item1	Item3	Item4		Pass
Item2								
Item1								
Item3								
Item4								
22. Adding items with different values in each term	Importance: Item 1 = 0.75 Items 2&3 = 0.5 Item 4 = 0.25 Due date: Item 1 = due nine days from current date Item 2&3 = due ten days from current date Item 4 = due five days from current date Duration: Item 1 = 4 Item 2 = 6	<ul style="list-style-type: none"> Rank_{Item1}=0.33*0.74+0.33*0.75+0.33*0 = 0.4919 Rank_{Item2}=0.33*1+0.33*0.5+0.33*0.25 = 0.5775 Rank_{Item3}=0.33*0.5+0.33*0.5+0.33*0.25 = 0.4125 Rank_{Item4}=0.33*0.67+0.33*0.25+0.33*0 = 0.3025 <p>Thus, the order of the tasks should be:</p> <table style="margin-left: 100px;"> <tr><td>Item2</td></tr> <tr><td>Item1</td></tr> <tr><td>Item4</td></tr> <tr><td>Item3</td></tr> </table>	Item2	Item1	Item4	Item3		No comment
Item2								
Item1								
Item4								
Item3								

	Item 3 = 3 Item 4 = 2 daysCompleted: Items 1&2&3&4 = 0 Dependency: : Item 1&4 = 0 Item 2&3 = 1			
23. Adding items with different values in each term	<p>Importance: Items 1&2&3 = 0.5 Item 4 = 0.75</p> <p>Due date: Item 1 = due nine days from current date Item 2&3 = due ten days from current date Item 4 = due three days from current date</p> <p>Duration: Item 1 = 1 Item 2 = 5 Item 3 = 4 Item 4 = 2</p> <p>daysCompleted: Items 1&2&3&4 = 0</p> <p>Dependency: : Item 1&4 = 0</p>	<ul style="list-style-type: none"> • $Rank_{Item1}=0.33*0.60 +0.33*0.5+0.33*0 = \mathbf{0.363}$ • $Rank_{Item2}=0.33*0.75 +0.33*0.5+0.33*0.25 = \mathbf{0.4950}$ • $Rank_{Item3}=0.33*0.60 +0.33*0.5+0.33*0.25 = \mathbf{0.4455}$ • $Rank_{Item4}=0.33*1+0.33*0.75+0.33*0 = \mathbf{0.5775}$ <p>Thus, the order of the tasks should be: Item4 Item2 Item3 Item1</p>		Pass No comment

	Item 2&3 = 1				
24. Adding items with different values in each term	<p>Importance: Items 1&2&3 = 0.5 Item 4 = 0.75</p> <p>Due date: Item 1 = due nine days from current date Item 2&3 = due ten days from current date Item 4 = due three days from current date</p> <p>Duration: Item 1 = 1 Item 2 = 5 Item 3 = 4 Item 4 = 2</p> <p>daysCompleted: Items 1&3&4 = 0 Item 2 = 1</p> <p>Dependency : Item 1&4 = 0 Item 2&3 = 1</p>	<ul style="list-style-type: none"> • $Rank_{Item1}=0.33*0.17+0.33*0.5+0.33*0 = 0.22$ • $Rank_{Item2}=0.33*0.6+0.33*0.5+0.33*0.25 = 0.4455$ • $Rank_{Item3}=0.33*0.6+0.33*0.5+0.33*0.25 = 0.4455$ • $Rank_{Item4}=0.33*1+0.33*0.75+0.33*0 = 0.5775$ <p>Thus, the order of the tasks should be: Item4 Item3 Item2 Item1</p>		Pass	No comment
25. Checking an Item that has over all higher ranking than other items	<p>Importance: Items 1&2 = 0.5 Item 3 = 0.75</p> <p>Due date: Items 1&2 = due four days</p>	<ul style="list-style-type: none"> • $Rank_{Item3}=0$ • $Rank_{Items2&1}=0.33*0.5+0.33*0.5+0.33*0 = 0.33$ 		Pass	Note: in terms of the equation, Item3 has higher overall rank even after being check; but for enhancing the usability and user experience, checked items

	<p>from current date Item 3 = due one day from current date</p> <p>Duration: Items 1&2&3 = 2 daysComplet ed: Items 1&2 = 0 Item 3 = 1</p> <p>Dependency : Item 1&2&3 = 0</p>	<p>Thus, the order of the tasks should be: Item1 Item2 Item3</p>		<p>now move to the bottom of the list</p>
26. One item is past due	<p>Importance: Items 1&2 = 0.5 Item 3 = 0.75</p> <p>Due date: Items 1&2 = due four days from current date Item 3 = due one day before the current date</p> <p>Duration: Items 1&2&3 = 2 daysComplet ed: Items 1&2&3 = 0</p> <p>Dependency :</p>	<ul style="list-style-type: none"> • $Rank_{Item3}=1$ • $Rank_{Items2\&1}=0.33*0.5+0.33*0.5+0.33*0$ $= 0.33$ <p>Thus, the order of the tasks should be: Item3 Item1 Item2</p>		<p>Pass</p> <p>Note: For enhancing the usability and user experience, due items are moved to the top of the list.</p>

	Item 1&2&3 = 0			
27. Two items are past due	<p>Importance: Items 1&2 = 0.5 Item 3 = 0.75</p> <p>Due date: Items 1 = due four days from current date Item 2&3 = due one day from current date</p> <p>Duration: Items 1&2&3 = 2</p> <p>daysCompleted: Items 1&2&3 = 0</p> <p>Dependency: Item 1&2&3 = 0</p>	<ul style="list-style-type: none"> • $Rank_{Item1} = 0.33 * 1 + 0.33 * 0.75 + 0.33 * 0 = 0.5775$ • $Rank_{Items2&3} = 1$ <p>Thus, the order of the tasks should be: Item2 Item3 Item1</p>		Pass
28. Checking an Item that is past due	<p>Importance: Items 1&2 = 0.5 Item 3 = 0.75</p> <p>Due date: Items 1&2 = due four days from current date Item 3 = due one day before the current date</p> <p>Duration: Items 1&2&3 = 2</p> <p>daysCompleted:</p>	<ul style="list-style-type: none"> • $Rank_{Item3} = 0$ • $Rank_{Items2&1} = 0.33 * 0.5 + 0.33 * 0.5 + 0.33 * 0 = 0.33$ <p>Thus, the order of the tasks should be: Item1 Item2 Item3</p>		<p>Note: For enhancing the usability and user experience, due items are moved to the top of the list.</p>
				<p>Note: For enhancing the usability and user experience, due items that are checked (completed for the day) are moved to the bottom of the list with the other checked items.</p>

<p>Items $1&2&3 = 0$</p> <p>Dependency</p> <p>:</p> <p>Item 1&2&3 $= 0$</p>			
---	--	--	--

