Kürşadcan AKAY

HEALTHY DIET AND EXERCISE PROGRAMME

Table of Contents

1. Introduction ................................................................3

1.1 What the Problem is ..........................................................................................................3

1.2 Goals for the Project.....................................................................................................3 - 4

1.3 Stakeholders......................................................................................................................4

1.4 Motivation for the Project..............................................................................................4 - 5

1.5 Process Flow Preview........................................................................................................5

2. Analysis and Design ...................................................6

2.1 Plan for Requirements Engineering ..............................................................................6 - 8

2.2 Functional Requirements.............................................................................................9 - 11

2.3 Non Functional Requirements...................................................................................11 - 16

2.4 Use Cases.................................................................................................................16 - 18

2.5 Models......................................................................................................................19 – 22

3. Project Plan ..............................................................23

3.1 Task Description........................................................................................................23 - 25

3.2 Task Assignment ......................................................................................................25 - 26

3.3 Deliverables and Milestones.............................................................................................27

3.4 Project Schedule ......................................................................................................27 – 28

4.Testing.......................................................................29

4.1 Features to be Tested ......................................................................................................29

4.2 Test Cases ................................................................................................................29 - 30

4.3 Testing Schedule .............................................................................................................30

5.Conclusion.................................................................31

5.1 The Problem and Solution ..............................................................................................31

5.2 The Team and the SE Process ..................................................................................... 32

5.3 Engagement of Umbrella Activities ................................................................................32

5.4 The Stakeholder’s that Benefited ...................................................................................33

5.5 The Organization’s Benefits............................................................................................33

6. User Manual..............................................................34

6.1 Software Description .......................................................................................................34

6.2 How to use the Software ..........................................................................................34 - 35

6.3 Troubleshooting Common Problems.........................................................................35 - 36

7. Implemented Design Patterns……………….……….37

7.1 Design Patterns .........................................................................................................37-38

1.Introduction

1.1 What the Problem is

The problem is that there is no program which contains all these following services: the program holds your personal data about nutritional disease based eating behaves, your personal data which you must concern about  as your BMI , current weight etc., displaying screen for your progress about your weight and date based information  in a well designed form and which advices recipes depending on your own needs.

As mentioned preceding paragraph, all of these services are not included in one program. Because of that a person who concerns about his diet must collect data or save data from to many different sources separately. This manual collection and distinguishing process, wasting time of people. Even all of these data which the user get from the Internet or from another sources maybe not secure and if there is a nutritional disease this could be very dangerous.

When the users are decided to gain weight, loss weight, stay fit they need some menus because nutritional behaviors are very important for this process this nutritional contents are very very important because of users diet selection based on these values strictly. But again these previously mentioned values are not included in one program, users must get this information manually from the different sources.

1.2 Goals for the Project

The main purpose of program is collecting all of these services in one program for people who concerns about his diet. When someone has a nutritional disease it's hard to select a menu and this struggle is real to make something easier for people about selecting menus and concerning about their diet programs .This application thanks to all the following services the users problems will be fixed very smart way .Firstly advising menus which are really versatile that means there's lots of options for the users and the users are not must eat same thing for every and every day they can have much more batter taste with same concerns from different menus thanks to our service. Even when user wants to loss weight, gain weight or stay fit we makes this process much more easier thanks to our personalized data screen personalized exercises in addition personalized menus.

Mainly, our purpose is creating previously mentioned all-in-one program. Which has user-friendly interface and sustainable life cycle of processes with additional improvements like motivation massages, gif-backed personalized exercises, recipes etc. The people who concerns about diet can easily access all information easily thanks to our well planned program.

1.3 Stakeholders

Our stakeholders are program users because of our B2C model. While developing the program we noticed about people’s needs especially about their diet. We concerned about their nutritional disease and program selections both. Their request depends on one of main behavior of human which can easily testable. Because of that stakeholders feedbacks will be so direct and to avoid bad feedbacks we researched their main demands and needs together. After understanding stakeholders main requests we created well organized menus and advices contents.

1.4 Motivation for the Project

For healthy generations we have responsibilities. One of those make a better world as much as we can do. As programmers we can design a system for people who can not easily reach data with their opportunities or don’t have enough time to distinguish wrong and true data about nutritional issues. To avoid this we have a motivation about the creating a system and helping the people for healthy and green generations. For these purposes our developing team started a journey with good wishes.

Our menus and recipes contents selected very carefully by our team member Ece, she is not anyone because she struggled for lose weight for years and knows the importance of menu contents and it effects to process. With this experience she is already self-motivated to help people. And our another team member, Ümmünur, believed the importance and effects of visualization of progress and she implemented very powerful interface for displaying this progress with dependencies of date and weight. And with the last but not least works team member Kürşadcan, developed safe and reliable relations between file systems and program with high motivation to create long term application to reach much more people. And all team members discussed for better user experience.

1.5 Process Flow Preview

While developing each stages of program we took advices and demands always from the users if worth to attend. Each module of the program created by the advices of professionals of this stuff. For example, while developing exercises section we took advices and information from professional sport trainers. We took advices from a dietitian while creating menus and recipes. While taking these advices we discussed about risks as well. We noticed all of these for each step. Sometimes we rearranged our flow and plan but they were minors because of our predefinitions and plans were already well-discussed and well-planed. We always asked to user and make changes immediately if it’s really necessary. Thanks to this agile approach we fixed problems with little cost. Always checked progress step by step and took action immediately. Each team member shared their progress and problems directly which helped to control process better. This effective communication allowed us efficient usage of time and energy. We helped each other if necessary to catch schedule. In conclusion, we managed software development process well thanks to well-designed steps and modular programming.

2.Analysis and Design

2.1 Plan for Requirements Engineering

**Inception Task:**

Firstly, we analyzed our stakeholder community. This definition is very important at the beginning for us. Because understanding the customer is very close to get solution. This need definition must be well-planned and clear. All following steps derived from this starting point. And if this starting point defines wrongly all following steps wont be suitable as well. Because of these all we focused on stakeholders real demands. This definition was extremely informative. The rest of the work about just converting solutions to computer language. With all these points we designed a path. We discussed about process, time, scope, quality, constraints, user experience, organization type, resources. Thanks to all previously mentioned factors we were able to get sight for the rest of process of project, we were able to define milestones, check points, quality standards. With this iron triangle based approach we defined our purpose very well at the beginning for whole project and managed all resources very well. To understand shapes of our project that’s very smart way the asking following questions:

-What is the “Healthy Diet and Exercise Programme” ?

-What is the additional difference of “Healthy Diet and Exercise Programme” ?

-Who can use “Healthy Diet and Exercise Programme”?

-Why you should use “Healthy Diet and Exercise Programme”?

-What you can not expect from “Healthy Diet and Exercise Programme”?

-Why you should trust us?

-What could be added to “Healthy Diet and Exercise Programme” in the future?

-Is it important mine demand as a user, do you care about it?

**Elicitation Task:**

There could be many approaches for same problem but distinguishing best one is another deep problem for some cases. To define this optimization we need some information and make some decisions. These all depends on program’s primary purpose. For us, the answer is that fastly reachable, sustainable, reliable and user-friendly solution. For this primary purpose, we need to eliminate something which one is not have priority. Our priority is well-structured file system to hold user progress and data. This is very important primer for our solution because of the nature of diet process. When user decided to start a diet, it’s about months even years. This long time period tells us that most important point is reliable and sustainable system. As mentioned previous sentences our service depends on long time period and now the time for definition another priority. We are declaring our program as a personalized service that all in one. If we advise some menus to users while concerning about their daily calory needs and other nutritional parameters, we should collect and call all of this data to avoid wrong advices and harmful results. All requirements ,all steps, all approaches will associated with these declarations.

**Elaboration Task:**

Information gathered from the inception and elicitation stages are merged during this part. All interaction scenarioes between user and program thought here. And inner dynamics of interactions for the program are included into process here as well.

**Negotiation Task:**

We come up with possible errors in this stage. For fixing possible problems from the sight of user we predicted and tested lots of conditions. We focused on rising up user experience for our program and because of that we took their ideas in each step.

**Specification Task:**

In this task, we created lots of outlines and visual materials to ensure main pattern of our structure. We used diagrams, schemas, templates even hand-written logical symbols to specify programs structure. And every side of project thought from different sights. For example, use case diagram, class diagram, sequence diagrams are used. We focused on user experience so deeply. With possibility schemas we tried to create blue print of possible errors and checked them all. We thought about possible errors systematically. After all of this, specification is completed.

**Validation Task:**

In this stage, we ensured about definitions specifically. To avoid turn backs that cost to much we defined almost all details of our project in this step. To ensure some details was sometimes so costly, so we made decisions with the concerning of our priorities. These decisions were generally about visualization issues. These little decisions were not needed to ask to user for each. But rest of decisions and specifications are defined with effective communications in group which includes even users. At least three or four times in a week we checked last form and design of our project in developers team to avoid costly turn backs. And thanks to these all attendance for every side project, every specification completed and ensured.

**Requirements Management:**

Thanks to our well-structured object oriented implementation, our changing requirements reflections were so agile. Flexible and well-formed structure allowed us to implement changing demands on time. Already, we always communicated with users. This communication helped us about decreasing requirements changes in process. Although these precautions, we faced with problems. These problems generally did not change or effect all process. In conclusion, thanks to all these attentions changing requirements demands did not effect development process significantly.

2.2 Functional Requirements

**Hardware Requirements:**

Our program must be run on desktop or laptop environments. Operating system does not matter for our program. Ethernet connection no needed, program works offline. Saves data to secondary storage units of computer.

**User Interface - Primary Tasks:**

* Allow for registration
* When sign up button is selected, navigate to register screen and take data of new user.
* Allow to user log in
* If the user already registered, after controls allow him to sign in.
* View main screen
* If user signed in or signed up, main screen is displayed with selections of personalized profile, menus, recipes, exercises, progress graph, edit my weight, leader board and motivation messages.
* View personalized data and profile
* Display personalized data after all updates.
* View personalized menus
* Display personalized menus depending on user’s daily calory need, nutritional disease.
* View personalized exercises
* Depending on program selection, display exercises to user.
* View recipes search screen
* Display a screen and search bar for recipes.
* Allow to user edit date and weight
* Allow to user edit date and weight with a pop up screen.
* View personalized progress graph
* Display user’s weight and date data couples in a graph to make much more understandable of tendency.
* View leader board
* Display top-3 users depending on currently active user’s program selection.
* View motivation messages
* Display messages to motivate user.

**User Interface - Secondary Tasks:**

* Control username and password if sign in selected
* Check steps works in an order which hierarchically designed for all conditions of each step. These steps includes username and password validation, input types checks, special character existence and restrictions.
* Control register screen data inputs for several validation rules
* When a new user tries to submit, we control his inputs under two header. First one more about username and password. Firstly we control their forms, if it is okay secondly we check repetition of username because of possible collisions. Second one about program selection and date. We control logical acceptances of given inputs.
* Allow hide and show password
* For much more secure log in process, we allow to user hide and show password with a button.
* Allow upload/change profile photo
* Allowing upload and change profile photo
* Allow to save a menu
* Allow to save a menu which listed in personalized menus screen, this selected menu displayed in user’s person profile as saved menu.
* Allow to sort personalized menus by protein, carbohydrate, fat, calory
* When we display personalized menus, we also allow to user sort them by protein, carbohydrate, fat, calory values.

2.3 Non Functional Requirements

**Performance Requirements:**

With our file structure preferences there will be no performance loss. With usage of Java’s reliable libraries, we reach to files and make operations with optimize algorithms. Thanks to our distributed file system structure, we work with less amount of data per operation. And our object oriented system helped to create sequential control systems which allows us to make control step by step and make operation in an order. These all together concerned in our development process to get a good performance report. Our main requirements completed with these approaches and there is no critic latency.

* Make controls in several milliseconds before sign in or sign up.
* Make changes in progress data file immediately when user adds new data.
* Display all personal data in several milliseconds which must be reading from file.
* When one attribute of user changed we must update it in secondary memory units directly.

**Security Requirements:**

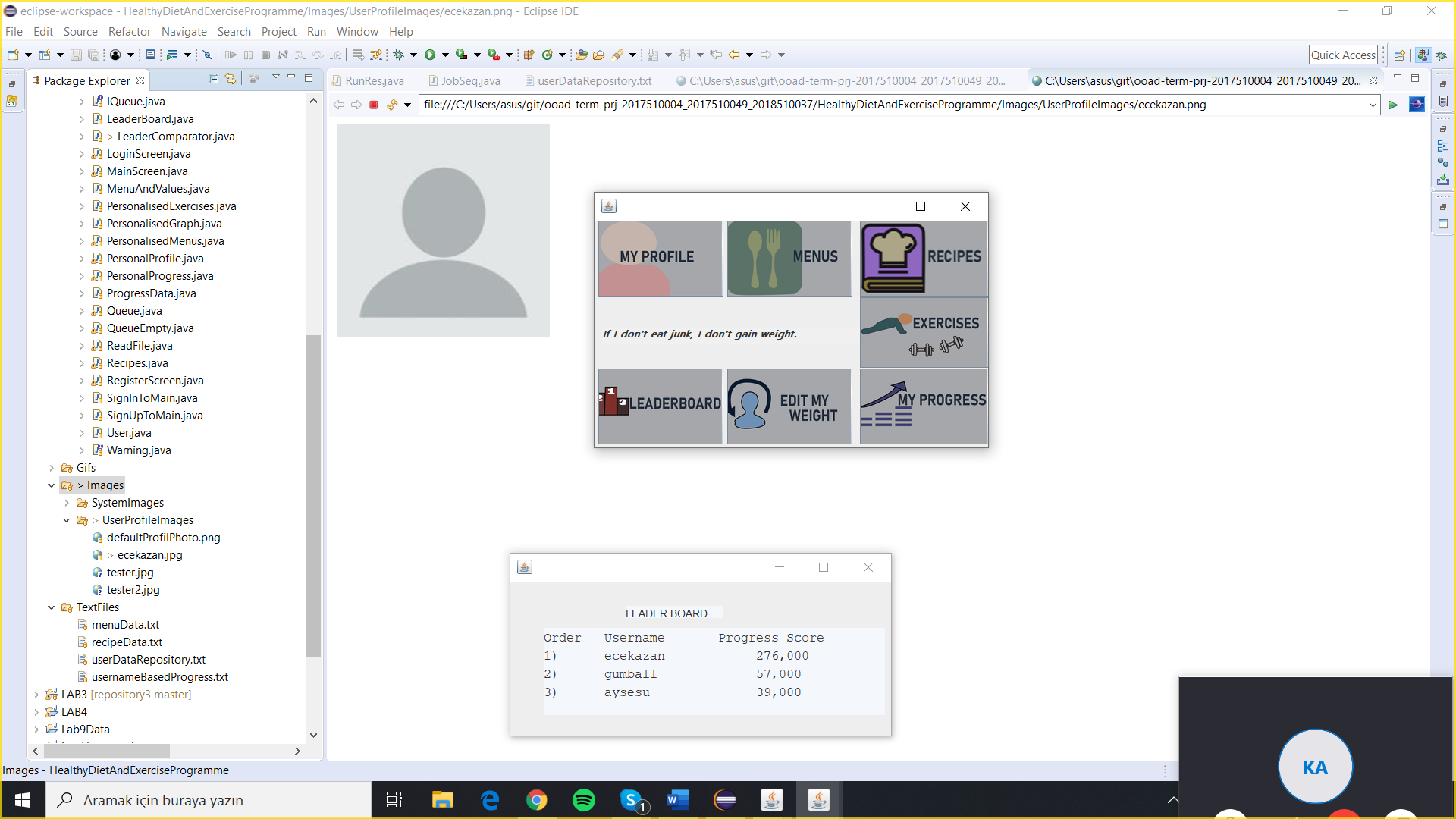
* We must allow hide and show password while user sign in.
* Only our program makes changes in the file directly rest of accesses and changes can not be directly with usage of read only format conversion by program.
* We don’t hold in an unique string object directly that created for holding password to avoid reachability from garbage collector.

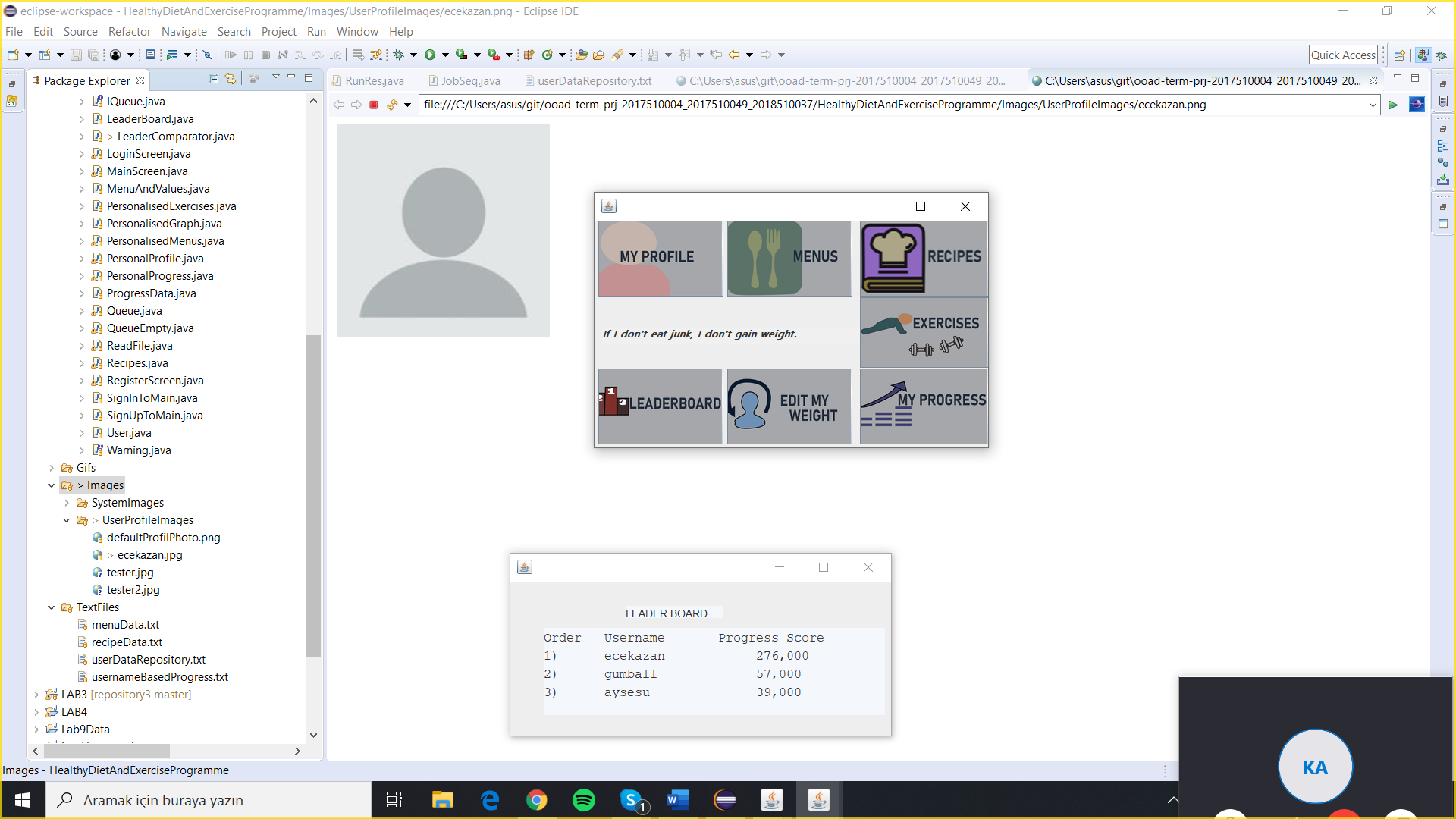
**Quality Attributes:**

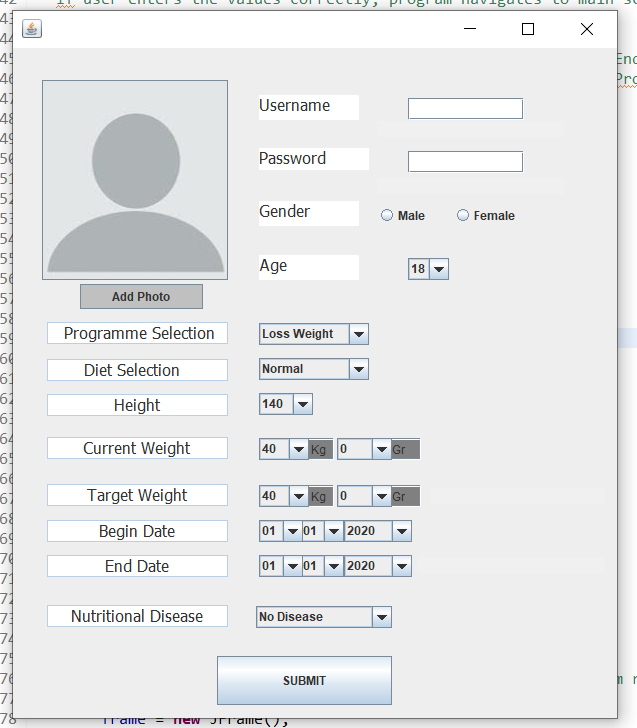
* Lots of visual materials and user-friendly interface must be implemented for better user experience.
* Lots of tests must be applied for difference conditions to fix all possible bugs.
* Care about stakeholders ideas which means users for us. And make changes with their navigations depending on real demands after realization.
* Display messages to user when error occurs within related reasons.

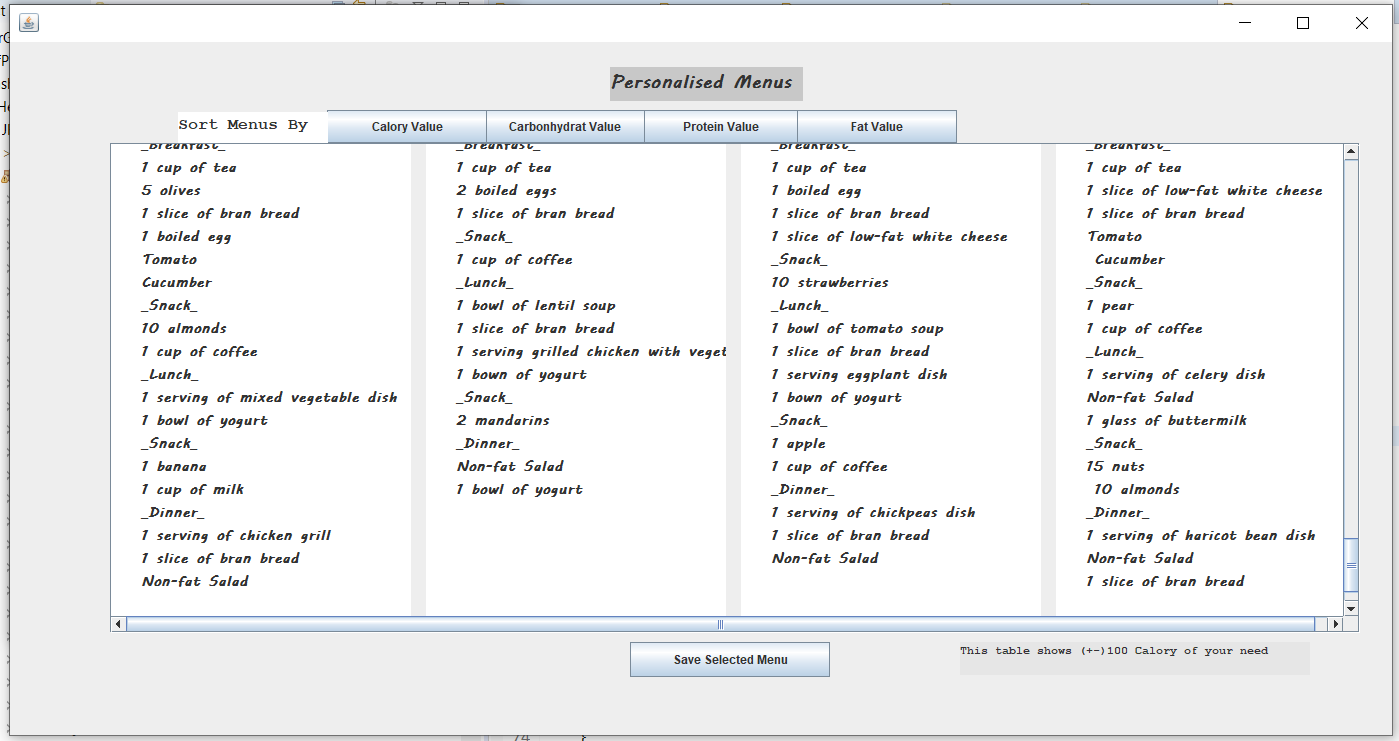
**Screenshot Mockups:**

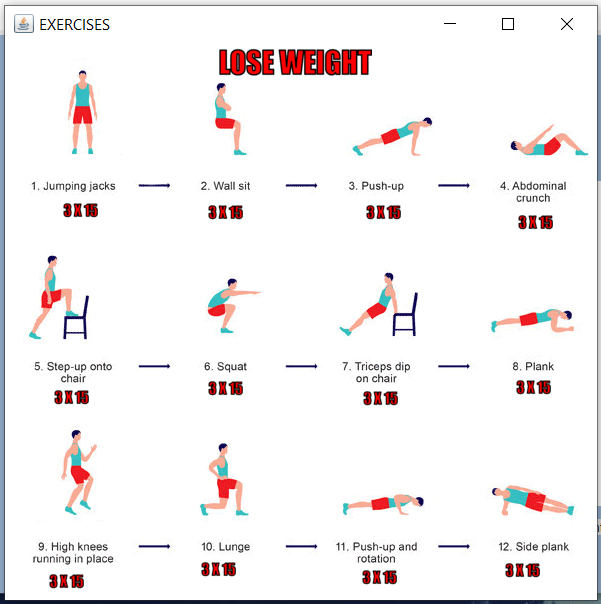


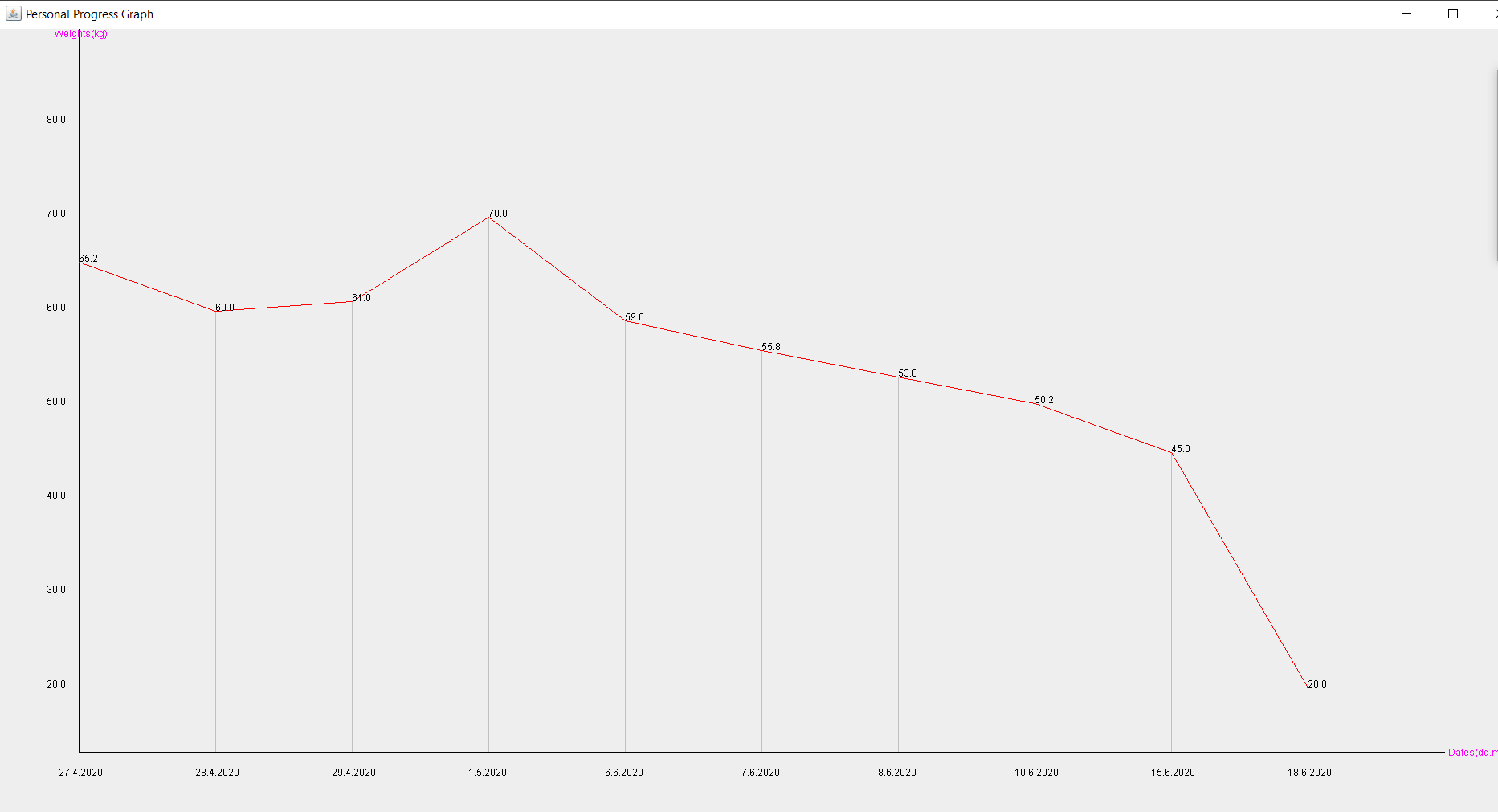


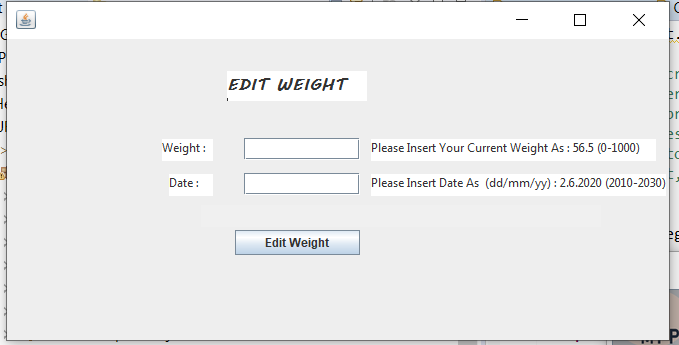


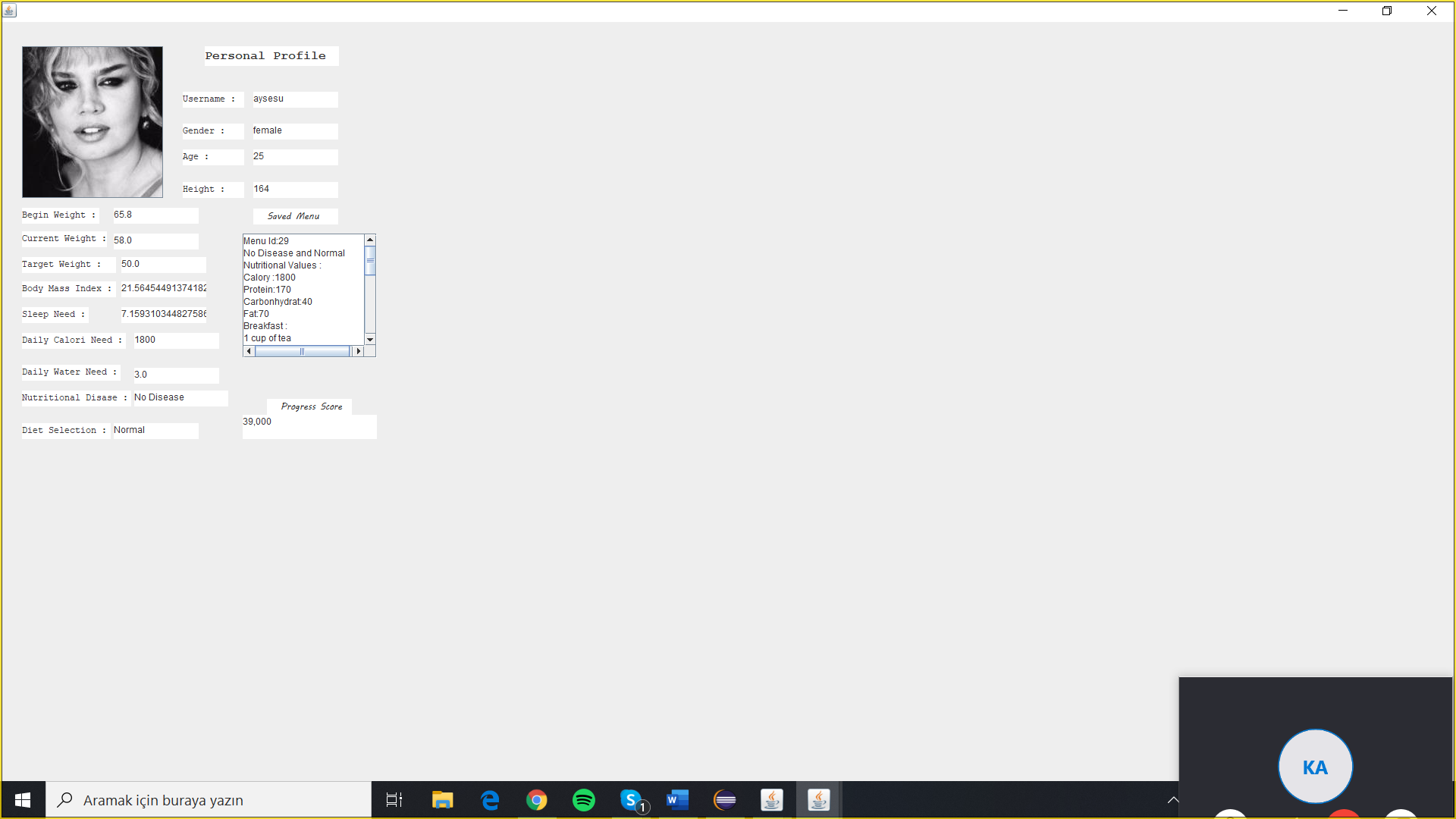














2.4 Use Cases

**Use Case #1: Add New Weight Data With Date**

**Primary Actor:** Program User

**Goal in Context:** To add new weight data and date data to user related records.

**Preconditions:** User had to be registered and his program period must not be expired.

**Trigger:** User should know date interval.

**Scenario:**

1. User enters username and password.

2. User selects “Edit My Weight” on main screen.

3. User enters weight and date information.

4. User related information are updated.

**Exceptions:**

1. Username and password types are not suitable: When given inputs length and character contents are not expectable.

2. Username and password are incorrect: When username and password are not correct, we check related possible errors as “username not found”, “wrong password”.

3. Date and weight types are incorrect: When date and weight inputs are not acceptable, we display related error messages as “wrong input type”.

4. Out of bound date: When user enters a date which is not in predefined interval. We display error messages and acceptable date interval.

**Priority:** Essential, must be implemented

**When available:** When date not expired

**Frequency of use:** Couple times per day/ week/month/year.

**Channel to actor:** Program interface

**Open Issues:**

1. The program allows to user end date change directly?

**Use Case #2: Select and Save A Menu**

**Primary Actor:** Program User

**Goal in Context:** To save user’s personalized menu.

**Preconditions:** User had to be registered and his program period must not be expired.

**Trigger:** User should clicked Menus button.

**Scenario:**

1. User enters username and password.

2. User selects “Menus” on main screen.

3. User can sort menus by protein, carbohydrate, fat, calory.

4. User selects a menu.

5. User can reach to last saved menu in his personal profile section.

**Exceptions:**

1. Username and password types are not suitable: When given inputs length and character contents are not expectable.

2. Username and password are incorrect: When username and password are not correct, we check related possible errors as “username not found”, “wrong password”.

**Priority:** Essential, must be implemented

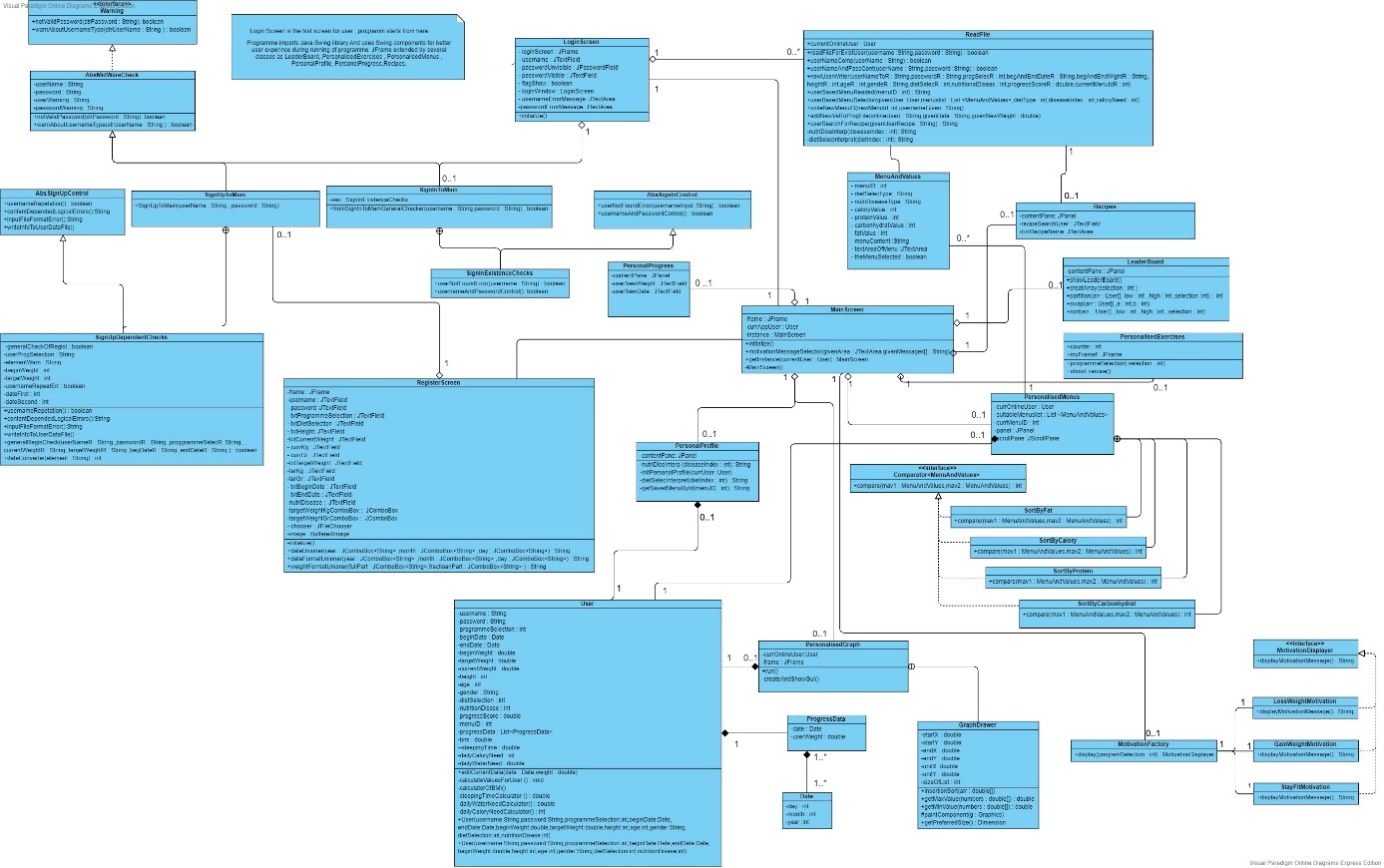
**When available:** When user signs in or signs up.

**Frequency of use:** Couple times per day/ week/month/year.

**Channel to actor:** Program interface

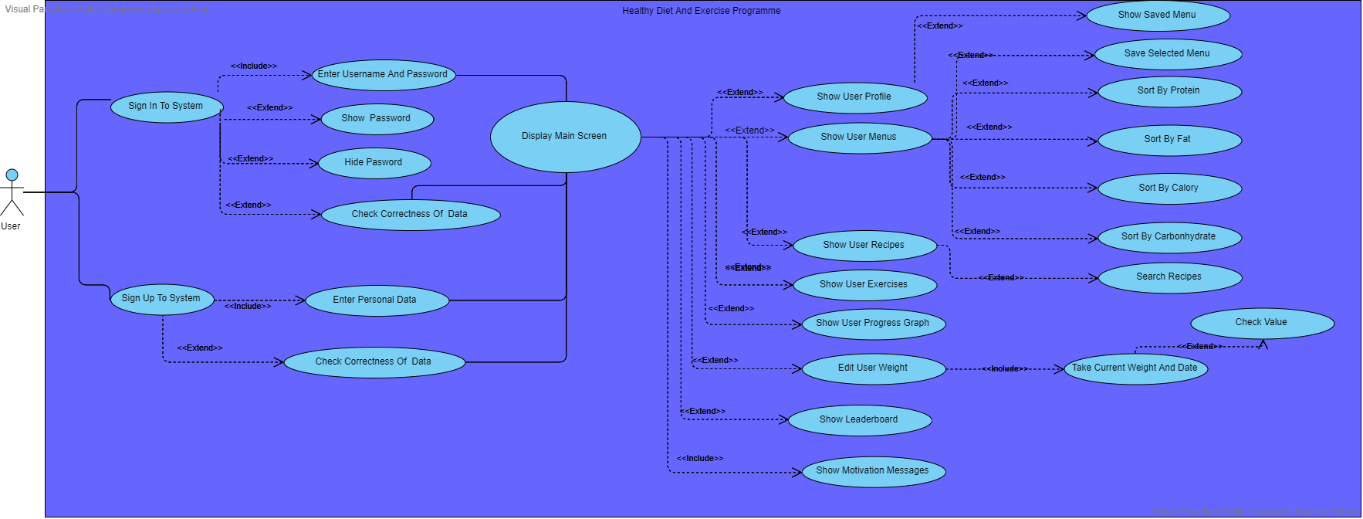
2.5 Models

**Class Diagram:**



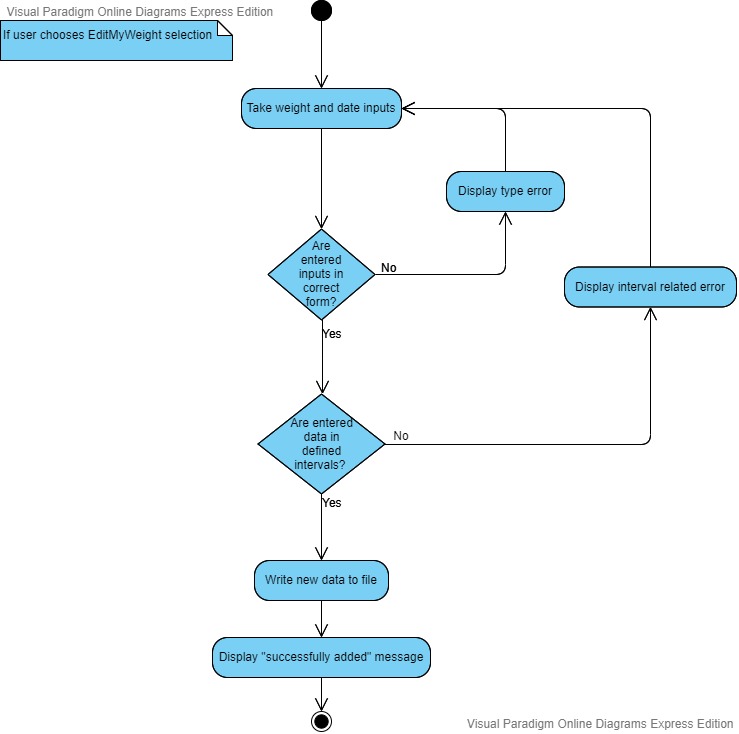
The class diagram given above refers class structure of Healthy Diet and Exercise Program that includes implemented design patterns such as singleton design pattern and factory design pattern.

**Use Case Diagram:**

****

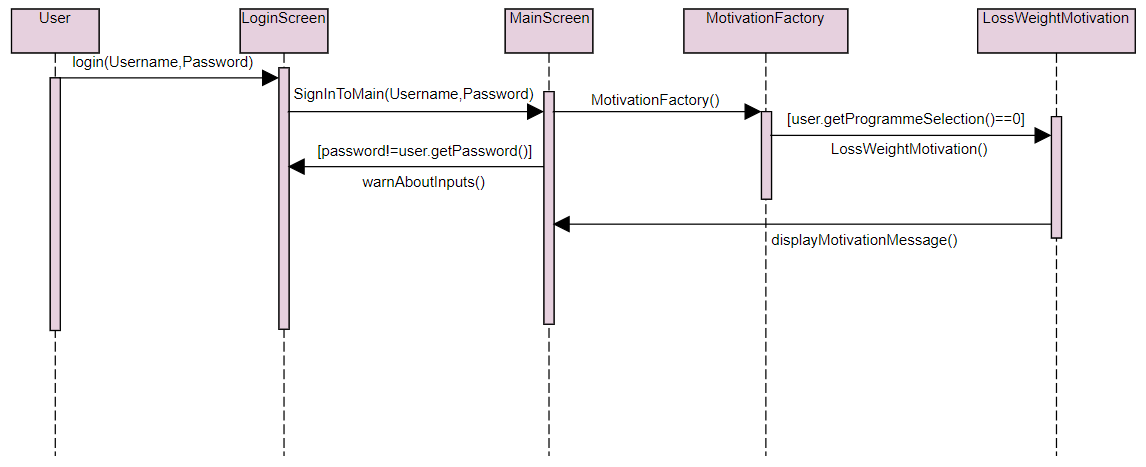
This use case diagram refers journey of user depending on possibilities.

**EditMyWeight Activity Diagram:**



The activity diagram refers “Edit My Weight” operations step by step. The operation called EditMyWeight takes user’s new weight and date inputs by the order that additionally contains control methods of given inputs depending on both type restrictions and interval restrictions.

**Sequence Diagram Of Motivation Messages:**

****

The sequence diagram given above displays user motivation messages steps in appropriate form. The users with aim of loss weight see loss weight motivation messages naturally. This message displayed by program as given sequence diagram.

3.Project Plan

3.1 Task Descriptions

**Stakeholder Meetings:**

We reach our stakeholders demands first which are predefinitions of ways to access them with links between sport community and us thanks to online platforms and face to face communication channels. After these channels created, we communicated with them periodically on different platforms and places either online and offline. We always noticed their expectations, opinions registered in which written format. We compiled all these parameters in a realistic perspective and get results, movement definitions.

**Design Models and Mockups:**

We used visualized materials so often, regardless formal definition needs of these visualized elements. This means we used lots of visual materials to communicate and design explicitly and implicitly both for all process because a picture is worth a thousand words we believe in. Sometimes we designed outlines and discussed about it. We tried to preview our user interface design thanks to design tools of development environments. And all of these usages are contributed our process significantly.

**File System Creation:**

File system design was very important issue for development process. Because in the secondary storage units we have to organize these data well. To avoid conflicts, data loss or

chaotic possibilities we focused on these details from beginning to end. We save users all data in this files. So their properties or forms must be well selected. To organize them in a efficient way we need to determine clusters of related data. This clustering partition is important to create human and machine readable format simultaneously. Because our data is also in human readable format to allow making changes in the future. After this formatting and clustering for related data together completed, we created hierarchical file system to both reach and manage easily. We added some system files and user related data under this system. In addition we defined all files as read-only for explicit accesses. Thanks to all of these we created well-organized file system.

**User Interface (Presentation Layer) Creation:** We focused presentation layer too much because of that we wanted to have positive feedbacks from users. User-friendly interfaces and easy to understand and navigate design created for much more better user experience. We designed our UI with design programs specifically. Most of our visual materials created by us. We discussed about visual elements, colors and their harmony. This brain storms contributed our presentation level design significantly. We created some demos and asked to users to understand which one is better for them. This user-friendly designed proses implemented well thanks to facilities of Java-Swing and Adobe programs and of course our designers.

**Testing:**

Testing was one of most important issue for us. From the beginning to end we always thought and discussed about testing strategies. This testing approaches were mortally important for life cycle of program. Because of nature of our program we have to save users data for a long time period. This means we have to collect and manipulate them in a well designed way. Because of snowball effect we may loss one user to many as known a stitch in time saves nine. We used visual materials to think about all possibilities and their probable results. This visualization helped us to think about scenarios systematically. We created tables and thought about their relations between error scenarios. For each level of coding we care about errors. To avoid them we used OOP designs and discussed about scope of error possibilities. For each stage of program we tested them well and reported these to team members. If there occurs user related error we display them in predefined places at UI with description of error. Thanks to all of these testing steps we decreased unpredictability of our program and raised reliability.

**Finalization and Reports:**

When all previously mentioned stages completed we started to finalize and report our program. This all conditions, steps, constraints, tasking issues etc. written in this step together. We tell described our programs properties, functionalities, development process , used materials and approaches etc. all included here as much as needed to inform. Each staged had its own explanations and report parts as milestones. These were helped us even about determine where we really are.

3.2 Task Assignment

For us , task is defined with its owner. We take care about our tasks like that. Because we think that task is not just a definition for someone. It has a mission and it must be provided to make users really satisfied. This kind of definition we prefer because of that task is kind of little explanation part of our program. When a task completed well it means we are much more closer to real achievement. While concerning all details as much as needed we don't forget to check progress of our team as well. Thanks to our well-defined task sharing and effective communication channels we always caught the calendar. This definitions formally so close to agile approaches. Because we believe in that communication is key for success. And this belief allowed us to understand what users really expecting or are they certainly sure about their demand. Why we care about that? Because big one of our task is understanding dynamically changing damans of user. This issue depends on communication. So from this point all of our team members discussed frequently while taking opinions of stakeholders. And we used almost all channels to communicate offline, online face to face etc.

As previously mentioned, communication was very important part about task definitions. Because task is not a static concept , it may change always. So after the completed one of the biggest task that called communication we were definitely sure about task sharing as following.

Our team member Ece, firstly focused on collecting suitable data form from different resources. This data was main role in our program. Because one of our main service about just collecting them instead of user properly. She researched for all details about this meaningful data sets. And advised alternative clustering methods or approaches for program. Thanks to her effort we created really optimize data sets, relations and well-distributed contents in files. And she created leaderboard screen and displayed successfully leader in the our program. Lastly she merged gifs for displaying personalized exercises.

Another member of our team is Ümmünur, as Ece she contributed all parts of project as well. But her main contributions were about following statements. She created personalized progress screen. Thanks to her effort this screen displaying users data-weight based progress in a two dimensional cartesian space. This visualization step one of main advantages of our program because this graph is very helpful to understand trend of user. She implemented lots of mathematical definitions in this screens backend. Thanks so this optimized solutions our program don't have any latency about any displaying mission.

In conclusion our teams last member Kürşadcan, contributed about following point. He firstly focused on implementation of object oriented programming principles as interfaces, abstract classes , classes and their relations like composition, aggregation etc. His first contribution was about this standard implementation into the project to create much more reliable, flexible optimize design. After implemented this polymorphism , abstraction , inference principles for main struct of project he focused on programs flow and related all other definitions. He created reading and manipulation classes for user or system related classes. He created sign up , log in ,edit weight, personalized menus, recipes screen etc. operations related controls and operations. And he designed navigations between UI frames .Lastly his one biggest task was being a part of good team with good teammates.

3.3 Deliverables and Milestones

We always checked our calendar and re-planned ourselves if needed. Thanks to well-defined flow of process we were able to understand where we are and where we should be . This reference point always helped to understand speed up or relax. We have mainly 4 stages for project:

1. Requirements Analysis And Gathering.

2. Designing and Coding stage.

3.Testing stage.

4.Demonstration Through Limited Stakeholders.

The stages given above always completed in time and planned order. Thanks to good managing of team and its natural result of predictability.

Deliverables for our processes mainly as following:

1. A main list for requirements which taken from stakeholders meets and design step.

2.The requirements related operations into the project. And design based works.

3.Get well results from users feedbacks thanks to well-designed UI and user-friendly interfaces.

4. Happy and satisfied users with comments which contributes to our program about being well-known and big-scale one.

3.4 Project Schedule

We started as a team in February to create “Healthy Diet And Exercise Programme”. We first defined our requirements and design related issues at the beginning. This period was mainly important about to get what we are gonna do with questions of why? how? where? with who? for who? etc. This kinda main questions answered and development process started after this introduction stage. The team member communications were so important as well and created channels to communicate with each other. As skype meeting, shared research drives, whatsapp groups etc. While discussing about possibilities and stakeholders demands we optimized ways to handle them. We defined our deadlines with no dependency of another formal definitions to take reference and discipline ourselves. After first several steps of this begin processes we started to coding of our previously designed steps while being in touch with stakeholders .And with stages which were gonna end in May we completed development process as following .After coding of backend and UI parts we started to test our program which is gonna be followed by demonstration step with values as following:

Requirements: 32 hours - 10.81 %

Design and code: 138 hours - 46.62%

Testing: 79 hours - 26.68%

Manual and Final Report: 35 hours - 11.82%

Demonstration and Adjustments: 12 hours - 4.05%

Total: 296 hours - 100%

4.Testing

4.1 Features to be tested

For our program, we had multiple testing conditions. Because of the versatile selection space of program we have to file related operations, methods runtimes, indexing of given data through the file system etc. Some sort of these conditions tested by our own developed strategies. We created diagrams to check all possible journey of any user. Thus we were able to understand possible errors in our program. To understand direct and indirect actions result in the system we used visual materials as well because a picture is worth thousand words. Thanks to our strategies we fixed lots of bugs. And created much more reliable program.

4.2 Test Cases

We had lots of cases because of previously mentioned wide possibility space. Combinational user defined attributes may affect file system related operations and user based displaying based operations both. We added lots of error message areas for possible cases nearby to error related input. Some examples giving as following.

Our program takes users current weight and target weight, at other hand we display three options for program selection as loss weight, gain weight and stay fit. So if you had selected loss weight program and then if you select a target weight which has higher value than current weight that logically not possible because if you want to lose weight your target weight must be lower. And similar logical errors are added into test cases.

Secondly, type dependent errors, for example your weight data cannot we a letter or string .This kind of type based cases added as well for all partition. This type base error already needed to check almost every user-interaction defined sections. This test case is needed to apply in various places even because of that we defined our type based test cases as interface.

Thirdly, we have a test case in screen functionality edit my weight data for date checking. Because we are taking your data and save then into secondary memory units. Thus we interpret users previous data and display some result for user. This relation includes definition of chronological sequence. This is what we check. Because if your editing weight and date request is not suiting with your end date of program like expire or not acceptable for your predefined interval we cannot allow use to submit new values.

4.3 Testing Schedule

Testing is an active process for us in each part of development even for requirements stage but our main testing schedule focuses on functionality of program which is really testable at the end. Because of that we implied our main tasting strategies into modular design when they start to have functionality. Thanks to these all we implemented successfully testing which made our program muck more reliable.

5.Conclusion

5.1 The Problem and Solution

The main problem definition depends on lack of service which provides all following properties. For people who wants to see versatile menus and combinations with concerns rely on nutritional diseases, eating behaves and their daily need of nutritional contents based on their program usage purpose like loss weight, gain weight and stay fit. This combinational service is rich dataset-backed. When we focused on people’s eating behaves especially when concerned about nutritional values we realized that for common they are often eating same things while just focusing on nutritional values they miss versatility of menu contents. This problem effects human health so badly. And this lack of variety has direct and indirect effects on human body.

Secondly, as mentioned in first paragraph contents of our meals are very important and we even call that’s importance with some examples like you are that what you eating etc. After that realization we focused on another part of this problem. Eating or drinking behaves are strictly connected with resting, daily routine and exercises. So we had to focus all these together because as mentioned they are effective each other. And lack of service which has no merge between these all. These properties are existing in different programs but our program collects these meaningful properties together. And making all services reachable and personalized in easy to navigate, user-friendly interface backed was main valuation of our solution.

In conclusion, previously mentioned problems and solutions implemented and thought together with conditions of user experience importance, valuation, sustainability, readability, reliability, flexibility stuff. These solution strategies are based on agile approaches, highly motivated team members with their task response, check points and calendars, strong development techniques in an useful manner for both user and developer. We created Healthy Diet and Exercise Program with high solution valuation.

5.2 The Team and the SE Process

Our team preferred agile approaches with strongly created channels and communicative behaves. Thanks to all strong communications we always catch the calendar. Team members always share their opinions obviously. This first stage of the being team completed with the last touch of task definition steps. We shared our tasks fairly and helped each other always as much as our primary tasks allowed in the project. We tried to think with approach of iron triangle principle. Task scheduling is distributed with visualized diagrams in our development team. For example, we created class diagram and each member get their own task definitions which assigned from this class diagram. We scheduled our program thanks to these all helper approaches well.

5.3 Engagement of Umbrella Activities

1. Software Project Management: With OOP principles and dynamic approaches with rely on communication. Manage process with visualized materials and diagrams. Created check points and milestones. Predefined conditions provided and their quality conditions thought by iron triangle.

2. Technique Requirement Providing Progression Management: We always controlled are we provide requirements as expected with highly contributed brain store and discussing meetings.

3. Sustainability Management: For both edges of user side and development side we focused on sustainability conditions. Because of our program’s nature time line of any user could have expands through decades.

4. Flexibility Management: Thanks to our well-designed software architecture we allow new futures. This allowance has low cost because of highly implemented OOP principle.

5.4 The Stakeholders that Benefited

Our stakeholders are benefited highly. They are already using this application for their own health. And our other stakeholders apart from users which mentioned previous sentence, personal trainers, dietitians are benefited because of that rising consciousness about exercises and nutritional behaves. This consciousness helps to being understood by community for people who attends these points. This realization benefits our stakeholders.

5.5 The Organization's Benefits

Our organization had very big valuable meetings which contributed our visions to understand how to propose a solution for a problem that must be thought by project management principles. This know how contributed our vision with attendance of much bigger works. As a kernel of organization we realized that our composition has meaning with all related branch attendance. Our organization’s members realized that their powerful and talented properties in the other way can call a self journey through our owns. This realization is very important for any organization because of that organization structures not only depends on their hierarchical system. Our partners and relations for both internal and external promise abilities for much bigger problems even thanks to all dynamics of our organization.

6.User Manual

6.1 Software Description

“Healthy Diet and Exercise Program” is a program which provides personalized needs based solutions through user. This service depends on user purpose which are previously defined as loss weight, gain weight and stay fit. When an user joined the system makes program’s selections as given previous sentence. When user makes this decision secondly we expect user’s selection about eating behaviour between vegan, vegetarian and normal. And following step, we take user’s nutritional disease between celiac, reflux and diabetes if exists. This information is very important for our solution because we concern about this issue while advising menus to avoid harmful results about advising strategies. If we declare three statements given above following attribute is interval of program usage because our service takes target and begin dates also weight as well. And other information are taken just for monitorization or interpretation which are meaningful returning values for nutritional processes like BMI. We take additionally profile photo to create much more personalized and user-friendly profile. In conclusion, we display and manipulate all user’s related meaningful data into efficient form. We display user’s profile photo, username, age, gender, height, begin weight, current weight, target weight, BMI, daily calory need, daily water need, daily sleep need, progress score, nutritional disease, eating behave and last saved menu.

6.2 How to Use the Software

Usage of “Healthy Diet and Exercise Program” is very easy thanks to user-friendly interface and easy to navigate relational contents. Our program includes lots of visual material which added for much more understandable design.

Program displays users a sign in screen which includes three functionalities as sign in, sign up and hide/show password. As understandable from their names these operations call all related stuff. When user selects sign in, we look username and password’s harmony. Else when user selects sign up, we navigate to sign up screen. When user selects hide/show password display button which settled side of password area, program just hides and shows password with special character. When user signed up with filled areas mainly focused on program selection, nutritional disease, eating behaviour our program creates personalized profile, personalized menus, personalized exercises depending on previously mentioned selections. User can save one of menu card in menus which is gonna be displayed in personalized profile. We compare all users and display top-3 of them depending on their progress score which calculated by correlation between program selection and user’s current weight. We display user’s progress in two dimensional graph with axis of date and weight. Recipes screen allows to user search for a recipe. Depending on user’s program selection, program displays personalized exercises. And user’s current weight taken in screen of edit weight with date and weight values. And lastly, program displays motivation messages.

6.3 Troubleshooting Common Problems

**Login Screen**

Problem: Username and password not valid

* If you see this error message, you should check your username and password harmony and make sure truth about them.

**Register Screen**

Problem: End date must be later

* If you see this error messages, you should check your dates about which one is late begin date or end date because of our program takes end date after from begin date.

Problem: Target weight must be lower/higher

* Because of our program selection definitions if you select one of loss weight or gain weight, you must select begin and target weight with this dependency. For example, if you select loss weight program but your target weight is higher, you get this error message and you must change end weight.

Problem: Username exists

* When your username is already taken, you see this error message and try another one.

Problem: Username/password type is wrong(length, character based error)

* If you see this error message, you should control your username/password about characters and length contents. For example, upper case usage is forbidden and if you try to use you see this error message.

Problem: File Format Wrong

* If you see this error message, you should control your file type which you are trying to set as profile photo. It must be ended with “.jpg”.

**Edit My Weight Screen**

Problem: Date/weight type error

* If you don’t enter date/weight values as expected which described at the right side of input areas, you take this message. And rearranged inputs into expected forms.

Problem: Date must be between … values

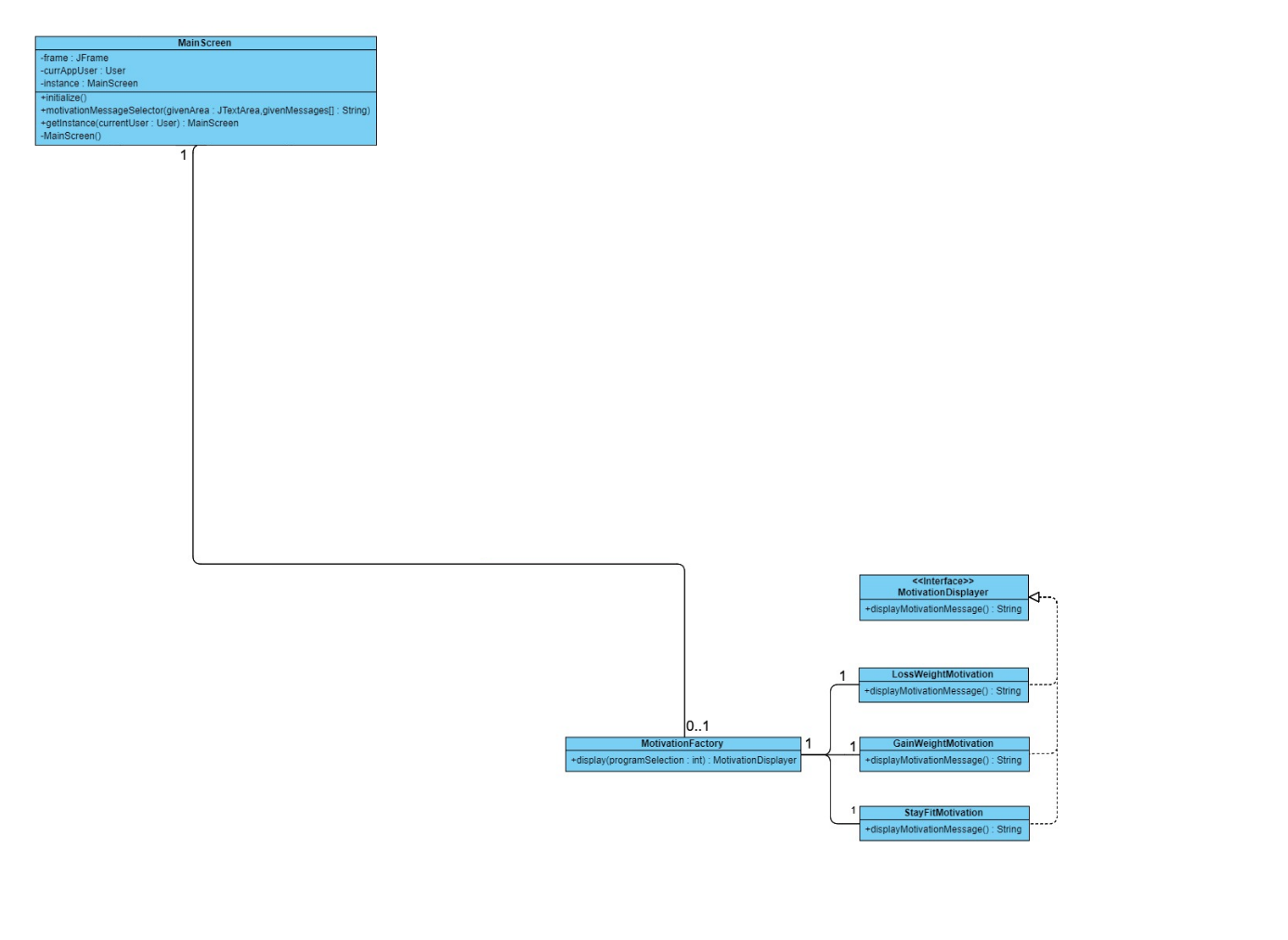
* If you see this error messages, you should check your input date. You can not input any dates beyond to end date or previous than last entered date or previous than begin date.

7.Implemented Design Patterns

7.1 Design Patterns

The following design patterns implemented:

1. Factory design pattern implemented into project as following. MotivationDisplayer interface implemented by LossWeightMotivation, GainWeightMotivation and StayFitMotivation classes. And these classes created from MotivationFactory class. Thanks to user’s program selection required motivation messages are created from MainScreen class as an MotivationFactory object. In conclusion, related motivation message is created in a appropriately implemented factory design pattern.



1. Singleton design pattern is implemented as following. Thanks to nature of our program’s life cycle singleton design pattern is very appropriate to implement in MainScreen. Our program needs MainScreen object only once statically. At the sum, singleton design pattern is implemented in appropriate form.

