

**THE REPUBLIC OF TURKEY
BAHCESEHIR UNIVERSITY**

**INFORMATION SYSTEM DEVELOPMENT FOR A
COMPANY IN FITNESS BUSINESS**

Master's Thesis

YASIN TANRIVERDİ

İSTANBUL, 2018

**THE REPUBLIC OF TURKEY
BAHCESEHIR UNIVERSITY**

**GRADUATE SCHOOL OF NATURAL AND APPLIED
SCIENCES
INDUSTRIAL ENGINEERING**

**INFORMATION SYSTEM DEVELOPMENT
FOR A COMPANY IN FITNESS BUSINESS**

Master's Thesis

YASİN TANRIVERDİ

Supervisor: ASST. PROF. ADNAN ÇORUM

İSTANBUL, 2018

**THE REPUBLIC OF TURKEY
BAHCESEHIR UNIVERSITY**

**GRADUATE SCHOOL OF NATURAL AND APPLIED SCIENCES
INDUSTRIAL ENGINEERING**

Name of the thesis: Information System Development for a Company in Fitness Business

Name/Last Name of the Student: Yasin Tanrıverdi

Date of the Defense of Thesis:

The thesis has been approved by the Graduate School of Natural and Applied Sciences.

Prof. Dr., Nafiz ARICA
Graduate School Director
Signature

I certify that this thesis meets all the requirements as a thesis for the degree of Master of Arts.

Asst. Prof., Adnan ÇORUM
Program Coordinator
Signature

This is to certify that we have read this thesis and we find it fully adequate in scope, quality and content, as a thesis for the degree of Master of Arts.

Examining Committee Members

Signature

Thesis Supervisor
Asst. Prof. Adnan ÇORUM

Member
Asst. Prof. Ethem ÇANAKOĞLU

Member
Asst. Prof. Mehtap DURSUN USTA

PREFACE OF ACKNOWLEDGEMENTS

I would like to thank all those my thesis supervisor Asst. Prof. Adnan ORUM, my friends and everyone who uses the application also i would like to thank android developers who helped me to solve technical issues during project development.

İstanbul, 2018

Yasin TANRIVERDİ

ABSTRACT

INFORMATION SYSTEM DEVELOPMENT FOR A COMPANY IN FITNESS BUSINESS

Yasin Tanrıverdi

Industrial Engineering

Thesis Supervisor: Asst. Prof. Adnan Çorum

January 2018, 47 pages

In recent years when the use of mobile devices has improved, different kinds of mobile applications have made a quick entry into our life. In addition to mobile devices, the interest in the healthy living and sport has increased considerably and many people have begun regularly sport and register in fitness centers. There are some problems that people encounter in these halls. Some of these problems are: Regularly following sports programs, diet and measurements which are not possible. Making reservations only by calling sport centers in sport centers where reservation system exists. Megaformer fitness studio android application is made as a solution of the above mentioned problems by combining these two popular objects.

In this thesis, what is mobile application, the history of mobile applications and the mobile development processes are examined. The android world, history of the android application world, its development and how the application is developed has been examined. What is the megaformer and lagree method and problems experienced by the megaformer studios around the world are examined. Megaformer fitness studio android application is examined. The advantages and development process of this application are examined. A technical examination of the megaformer fitness studio android application has been done. Materials used during application development, project schedule, actor modellings, data modellings and class modellings are examined.

Keywords: Megaformer, Fitness, Sport, Android, Mobile

ÖZET

FITNESS SEKTÖRÜNDEKİ BİR FİRMA İÇİN BİLGİ SİSTEMİ GELİŞTİRİLMESİ

Yasin Tanrıverdi

Endüstri Mühendisliği

Tez Danışmanı: Yrd. Doç. Dr. Adnan Çorum

Ocak 2018, 47 sayfa

Mobil cihazların kullanımının arttığı son yıllarda, bir çok alanda mobil uygulamalar hayatımıza hızlı bir giriş yapmıştır. Mobil cihazların yanı sıra yine son yıllarda insanların sağlıklı yaşama ve spora olan ilgisi oldukça artmış ve bir çok kişi düzenli olarak spor yapmaya ve fitness salonlarına kayıt olmaya başlamıştır. Bu salonlarda insanların karşılaştığı bazı problemler meydana gelmektedir. Bu problemlerden bazıları şunlardır: Düzenli olarak spor programı takibi yapılamaması, düzenli olarak beslenme takibi yapılamaması, düzenli olarak kullanıcının yaptığı ölçümlerin takibinin yapılamaması ve rezervasyon sistemi ile işleyen salonlarda rezervasyon sisteminin sadece telefon ile spor salonun aranarak yapılmasıdır. Megaformer fitness stüdyosu android uygulaması günümüzdeki popüler bu iki objeyi bir araya getirerek yukarıda belirtilen problemlerin çözümü için yapılmıştır.

Bu tezde, mobil uygulamanın ne olduğu, mobil uygulamaların geçmişi ve geliştirme süreci incelenmiştir. Android dünyası, android uygulama dünyasının geçmişi, gelişimi ve nasıl uygulama geliştirildiği üzerine incelemeler yapılmıştır. Megaformerın ve lagree metodun ne olduğu, dünya üzerindeki megaformer stüdyolarının yaşadığı problemler, Megaformer fitness stüdyosu için geliştirilen uygulama, sağladığı avantajlar, geliştirilme süreci ve uygulamanın teknik incelemesi yapılmıştır. Uygulamanın geliştirilmesi sürecinde kullanılan materyaller, proje planlaması, aktör modellemeleri, data modellemeleri ve sınıf modellemeleri incelenmiştir.

Anahtar Kelimeler: Megaformer, Fitness, Spor, Android, Mobil

CONTENTS

TABLES.....	ix
FIGURES.....	x
ABBREVIATIONS.....	xii
1. INTRODUCTION.....	1
2. GENERAL INFORMATION.....	2
2.1 MOBILE APPLICATION.....	2
2.1.1 What is Mobile Application ?.....	2
2.1.2 Mobile Application History.....	3
2.1.3 Mobile Application Development.....	7
2.1.4 Mobile Application Development Process.....	9
2.1.5 Mobile Application Development Life Cycle.....	10
2.1.5.1 Planning.....	11
2.1.5.2 Analysis Planning.....	11
2.1.5.3 Design.....	12
2.1.5.4 Development.....	13
2.1.5.5 Testing.....	13
2.1.5.6 Maintenance.....	13
2.2 ANDROID.....	14
2.2.1 What is Android ?.....	14
2.2.2 Android History.....	15
2.2.3 Android Applications.....	18
2.2.4 Platform Usage.....	20
2.2.5 Other Usages.....	21
2.3 MEGAFORMER.....	24
2.3.1 What is Megaformer ?.....	24
2.3.2 Lagree Fitness & Method.....	24
2.3.3 Articles Written About Megaformer.....	27
3. DATA AND METHOD.....	28
3.1 SYSTEM ANALYSIS.....	28
3.1.1 Use-Case Modelling.....	30

3.3.1.1 Actor Glossary.....	30
3.3.1.2 Use-Case Glossary.....	30
3.3.1.3 Use-Case Modelling Diagram.....	31
3.3.1.4 Use-Case Narratives.....	31
3.1.2 Data Modelling.....	31
3.1.2.1 Activity Diagrams.....	32
3.1.2.2 Sequence Diagrams.....	33
3.2 MEGAFORMER FITNESS STUDIO ANDROID APP.....	32
3.2.1 Application Features.....	33
3.3 SCHEDULE OF MEGAFORMER ANDROID APP.....	43
3.3.1 GANTT Chart.....	43
3.4 MEGAFORMER ANDROID APP TECHNICAL DETAILS.....	44
3.4.1 Used Materials and Services During Project.....	44
3.4.2 Process Modelling.....	44
3.4.2.1 Context Level Data Flow Diagram.....	45
3.4.2.2 Level-0 Data Flow Diagram.....	45
3.4.3 Class Modelling.....	45
3.4.3.1 UML Class Diagram.....	45
4. FINDINGS.....	46
5. DISCUSSION AND CONCLUSION.....	47
REFERENCES.....	48
APPENDICES	
Appendix A.1 USE-CASE MODELLING.....	50
Appendix A.1.1 Use-Case Glossary.....	50
Appendix A.1.2 Use-Case Narratives.....	54
Appendix A.2 ACTIVITY DIAGRAMS.....	80
Appendix A.3 SEQUENCE DIAGRAMS.....	91
Appendix A.4 DATA FLOW DIAGRAMS.....	100
Appendix A.4.1 Context Level Data Flow Diagram.....	100
Appendix A.4.2 Level-0 Data Flow Diagram.....	100
Appendix A.4 CLASS MODELLING DIAGRAMS.....	101
Appendix A.5.1 UML Class Diagram.....	101

CURRICULUM VITAE.....102

TABLES

Table 2.1: Android version usages.....	21
Table 3.1: Actors of application.....	30
Table 3.2: GANTT chart.....	43

FIGURES

Figure 2.1: Mobile application market leaders.....	2
Figure 2.2: Google play store and Apple app store logo.....	3
Figure 2.3: Nokia smart phones.....	4
Figure 2.4: Symbian OS logo.....	4
Figure 2.5: Java logo.....	5
Figure 2.6: First Android device HTC Dream.....	6
Figure 2.7: First iPhone.....	6
Figure 2.8: Android studio logo.....	7
Figure 2.9: Xcode logo.....	7
Figure 2.10: Software development life cycle.....	10
Figure 2.11: Mobile application development life cycle.....	11
Figure 2.12: Android logo.....	14
Figure 2.13: Google logo.....	15
Figure 2.14: Android version usages.....	20
Figure 2.15: Android wear.....	21
Figure 2.16: Android TV.....	22
Figure 2.17: Megaformer machine.....	24
Figure 2.18: Sebastien Lagree.....	25
Figure 2.19: Example megaformer fitness studio.....	26
Figure 2.20: Studio members during workout.....	27
Figure 3.1: Use-case modelling diagram.....	31
Figure 3.2: Megaformer app home page.....	32
Figure 3.3: Megaformer app login page.....	33
Figure 3.4: Megaformer app lessons page.....	33
Figure 3.5: Megaformer app lesson detail page.....	34
Figure 3.6: Megaformer app my reservations page.....	34
Figure 3.7: Megaformer app personal reservations page.....	35
Figure 3.8: Megaformer app older reservations page.....	35
Figure 3.9: Megaformer app current reservations page.....	36
Figure 3.10: Megaformer app sport programmes page.....	36

Figure 3.11: Megaformer app measurements page.....	37
Figure 3.12: Megaformer app packages page.....	37
Figure 3.13: Megaformer app announcements.....	38
Figure 3.14: Megaformer app profile page.....	38
Figure 3.15: Megaformer app community fit page.....	39
Figure 3.16: Megaformer app shopping page.....	39
Figure 3.17: Megaformer app card info page.....	40
Figure 3.18: Megaformer app messages page.....	40
Figure 3.19: Megaformer app messaging page.....	41
Figure 3.20: Megaformer app contact page.....	41
Figure 3.21: Megaformer app contact page map.....	42
Figure 3.22: Megaformer app guest user home page.....	42
Figure 3.23: Megaformer app guest user message.....	43
Figure 3.24: Android studio IDE.....	44
Figure 3.25: GCM logo.....	44
Figure 3.26: Iyzico logo.....	45

ABBREVIATIONS

ADT	:	Android Development Tools
API	:	Application Programming Interface
APK	:	Android application package
APP	:	Application
CEO	:	Chief Executive Officer
GCM	:	Google Cloud Messaging
IDE	:	Integrated Development Environment
NDK	:	Native Development Kit
OS	:	Operating System
SDK	:	Software Development Kit
UML	:	Unified Modelling Language

1. INTRODUCTION

In this Master's Thesis android application which is developed for Megaformer fitness studio has been examined. Since this application is an android application, android application development history has been examined as well. Megaformer is the machine that is used in all of pilates workouts. Its patented design enables customers to take advantage of a wide variety of exercises that build strength, endurance, flexibility, balance and mental toughness. Judging solely on the desing of the machine, customers can begin to see why a Megaformer workout is not like an average Pilates class.

The Istanbul branch of Megaformer fitness is studio is recently opened. Before opening the studio, the problems of other Megaformer studios around the world has been examined in order to avoid their customers from these problems and it has been determined that mobile appliation will be a solution for all of these problems. As a result of that situation this mobile application is developed.

The main problems occured on other Megaformer fitness studios were:

- a- Customers can not make reservations easily. Reservations are only done by calling studio or via website.
- b- Lesson details are not easy to access.
- c- Customers can not access their measurements details.
- d- Customers can not Access their reservations easily (Only via website).
- e- Customers only can bug packages in studio.
- f- Studio announcements etc. are sent customer's phones and emails manually.
- g- Coaches are not able to monitor customers daily diet.

These problems are examined detailed in second part. This mobile application is implemented as a solution for all of these problems and will be improved according to customers requirements.

2. GENERAL INFORMATION

2.1 MOBILE APPLICATION

2.1.1 What is Mobile Application ?

Smartphones and tablet computers have been used by almost everyone in one of the most important developments that have taken place in the lives of computers and entered into human life. These devices are generally called mobile devices, and the software designed and developed for their operation is called "Mobile Application". Mobile applications are written in accordance with the "Mobile OS" used by mobile devices and can be downloaded from the application markets provided by each operating system.

Figure 2.1: Mobile application market leaders



Today, there are many mobile OS. However, two of them have made noise in the world in recent years and are used by many mobile device manufacturers and mobile users. Applications developed by Apple and used in the iOS operating system they use on their own mobile devices can be obtained from the Apple App Store for iPhone, iPod and iPad. Mobile apps are available from the Google Play Store which is developed by Google for devices that use android OS which is used by many mobile device manufacturers and mobile users. Mobile applications can be downloaded directly to the mobile device through these application markets. If allowed, mobile applications can be downloaded to computer first and then can be installed to mobile device.

Figure 2.2: Google play store and Apple app store logo



Mobile applications on the application markets can be offered for free or for a fee depending on the quality of the service, its content, its characteristics and the application owner's wishes. The free mobile apps can be downloaded and installed on the mobile device without charge from the application market, and if so, the application can be removed from the device and reloaded. If the mobile application is paid, it can be downloaded to device after it has been purchased with credit card which is defined in the system.

Apple CEO Tim Cook announced in June 2013 that the App Store has more than 900,000 active mobile apps. The announcement, which was held at the end of October, informed the mobile application sector that this number reached one million. In addition, close to 60 million applications have been downloaded. On the Google Play Store, the total number of apps is over 1 million. The number of downloads has reached over 50 million although it is in a short period of time. This is an indication of how fast the mobile industry is developing.

2.1.2 Mobile Application History

By the introduction of mobile phones in our lives, the first mobile applications also appeared. By the first mobile applications, nothing else was possible except to search and send messages. An important factor in this was the fact that the first mobile OS were quite primitive and did not allow full use of the hardware. The fact that mobile applications are simple and primitive because mobile phones do not have enough

memory. This low memory also greatly restricted the number of people registered on the mobile device and the amount of data to be stored on the device.

During the early period of mobile apps, mobile apps came with the phone, and they did not have the ability to uninstall or install another mobile app on top of them. These mobile applications have begun to make life easier in both business and everyday life.

Figure 2.3: Nokia smart phones



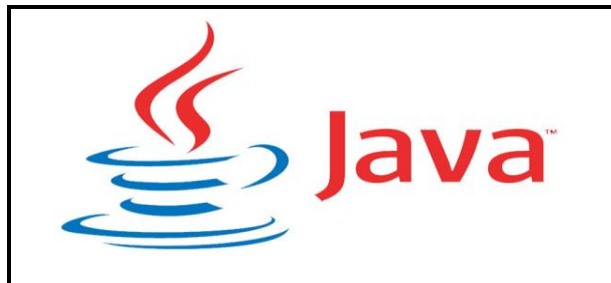
After mobile phones and mobile devices became widespread, much more flexible mobile OS based on “Java” emerged. The most important of these is the “Symbian” OS developed by “Nokia” company, which now has much more advanced versions. These operating systems were more advanced than ever and opened advanced mobile application doors. We were able to remove a mobile application that we did not want from new java based mobile OS and even download and install another mobile application which is suitable for mobile devices from markets with a limited number of mobile applications.

Figure 2.4: Symbian OS logo



By the emerging Java based operating systems, mobile devices have significantly increased their storage space, and these storage areas have become even more expandable with the memory cards. Internet connections from mobile devices are now faster, and the number of things that can be done with internet connectivity on mobile devices has increased significantly. In this regard, much more sophisticated mobile internet browsers appeared that allow users to navigate freely on web pages and send and receive files over the internet. In the 90s, mobile applications of popular messaging programs have been made and social networks have started to move to mobile. Mobile file manager applications were made to enable the mobile device to use its own internal memory and external removable memory cards effectively. One of the most interesting new mobile apps was the media player apps that came on mobile devices. Thanks to the mobile media player apps, music and video files stored in the memory of mobile device or on the internet could be played. Games, which are a separate place in the mobile application category, also started to develop and become widespread. By these mobile apps, mobile devices have become devices that can be used for entertainment purposes.

Figure 2.5: Java logo



By the proliferation and diversification of mobile applications as well as the development of mobile devices, user's expectations for mobile devices, especially mobile phones have increased considerably. As the number of prospects increased, each company had no chance to develop enough mobile apps to meet the needs of every user, so mobile app stores appeared. By the advent of these mobile app stores, companies have been able to develop mobile apps for other developers and users, in keeping with their own mobile operating systems. Mobile apps have exploded in the last 5-6 years as mobile app developers begin to develop mobile apps on their own as consumers demand them. As a result of this explosion, number of mobile application developers has

increased rapidly in the meantime, and the next generation of operating systems have caused this. Operating systems that have the most share in the development of mobile applications are the Android mobile OS, which was introduced for the first time in 2008 with the iOS mobile OS that Apple introduced in 2007 with the iPhone.

Figure 2.6: First Android device - HTC Dream



Figure 2.7: First iPhone



Mobile devices, which are growing rapidly in numbers today, have also led to the growth of the mobile application industry. Every mobile platform has been customized to hundreds of thousands of applications, with popular applications being developed on every platform to perform the same functions.

As a result, mobile applications have evolved at a tremendous pace over the last few years and have changed the lives of millions of people. We continue to live on technological boundaries with mobile applications that make every day life easier, convenient for business people, able to connect to social networking sites all the time and let us know everything immediately and have more fun in our free time.

2.1.3 Mobile Application Development

Today, there are two important factors that enable users to sell mobile devices, including mobile phone and tablet computers. The first one is which operating system you have. The second is how many types and quality mobile applications are available in the market where the OS has applications. Nowadays, two mobile operation systems are most commonly used in the mobile application market.

Mobile application development is the process of designing and creating a simple or more complex software for any mobile device operating system that it uses. This development process differs for each operating system. To speak of the most common operating systems, developers need to know objective-c or swift language for iOS operating system, while developer need good programing knowledge of java or kotlin programming to develop a mobile application suited to the android OS. The developer languages and development environments of each mobile OS are different.

Figure 2.7: Android studio logo



Figure 2.9: Xcode logo



There are 3 different ways that developers can follow in general to develop mobile application. The first one is “Native” application development. To develop a native application, a certain OS needs to be developed with a certain programming language. By this way application becomes more safer and more performant. Another way is to write a mobile app based on “Web” so developers can develop applications that can run on all platforms at once. Here application can be used directly through the browser of mobile device. It is an ideal way to avoid spending time and Money, but it’s one of the disadvantages of requiring constant internet connection and not working well. The third way is hybrid application development method indicated by “Hybrid”. In order to use this way, it is provided to Access a web-based system inside Native coding, so that two Technologies can be used at the same time.

To start developing mobile apps, it is first necessary to device which mobile platform the app will run on. It only needs to be decided whether it will work in the android environment or only in the iOS environment or both. If developer wants application to run securely and more efficiently on just one or few mobile operating systems, native application should be developed. Developing a web-based application will be much more advantageous if developer wants to develop a more current and web-focused application that runs on every platform at a low cost, but recently, reducing the “native” application support of popular operating systems can create problems in the future.

To develop apps on mobile platforms like Android and iOS , developer need to install some software to computer. One of them is software development tools known as SDK. These tools create an environment that helps and facilities app development. During developing app, library, debugger and everything else is needed.

The most commonly used development environment for Android is “Android Studio”. While developing an application for the Android Platform, developers can work on any computer with a Windows, MacOS or Linux based OS. For iOS, the situation is slightly different. The Xcode environment for iOS only Works on MacOS operating systems. To develop an iOS app developers need to have a computer with Apple MacOS.

The first thing, after a mobile app is developed is to distribute that app. Mobile app markets such as Google Play Store and Apple App Store have been created for this. Developers can offer their free app downloads through these mobile app stores, or they can monetize their downloads by offering them for a fee.

2.1.4 Mobile Application Development Process

Every mobile application project is a software project at the same time. For this reason, mobile application development processes are similar to software development processes. In order to be able to cope with the problems that may arise during mobile application development, the application has been developed in line with the mobile software development process.

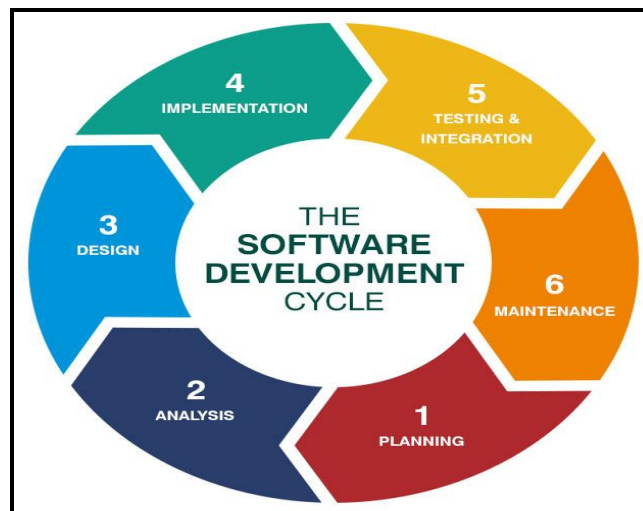
- a - In order to understand the application in detail, the needs of application to be done are taken from the owner as a document,
- b - Required analysis work is done,
- c - The requirement document is created,
- d - Scope and modules of work are identified in the requirement document,
- e - Approval is given for the scope and modules from the owner.
- f - Application development process begins,
- g - Design work is done,
- h - Development process begins after design approval,
- i - Once the coding is completed, app is presented to customer,
- j - Customer approval is taken,
- k - Implementation test cases are created,
- l - Planned tests are conducted,
- m - After the tests are done, customer approval is taken,
- n - The application is published on the market,
- o - Application maintenance and updating process begins.

2.1.5 Mobile Application Development Life Cycle

From the planning to the delivery of a mobile software project, all phases and stages of the cycle that developer have spent are called the mobile application development lifecycle.

The development of a mobile application project is not just about coding. Even when a project is being developed, the steps of planning, analysis, design, production, and testing of the project are taking place and should be taken. Once these phases are realized, the project may not be completed. These steps must be considered in a loop. This process is ongoing for issues such as future requests, error corrections, new modules to be added to the project after the project is completed. This cycle is called the life cycle of software development.

Figure 2.8: Software development life cycle

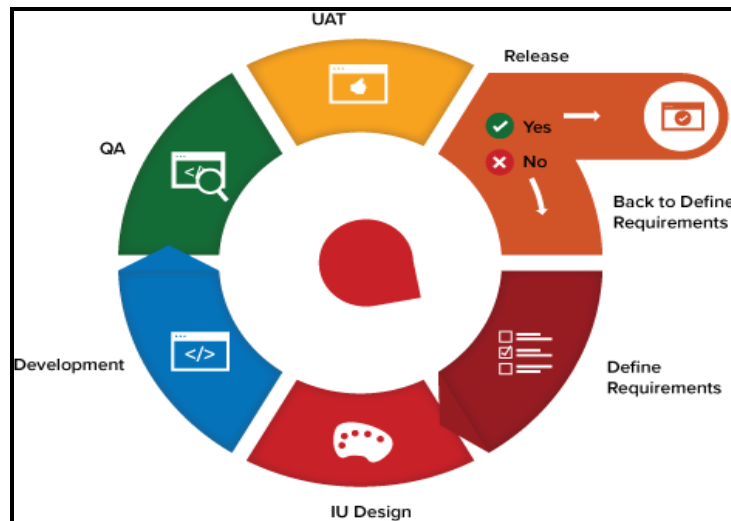


The software development process consists of timed and contented stages. In this way the software is being developed in a planned way. Because the requirements for software functions are constantly changing and expanding, the phases handled in a continuous loop. As shown in Figure 2.10, it is a matter of going in a cycle.

A mobile application project is actually considered as a software project. Assume that the software completes the projecting, analysis, design, production and testing phases, and completes the projecting process and then makes no errors and new requests. At this

point the cycle of the project is maintenance. All the software projects that have been developed will be added to the life cycle of future maintenance mistakes and maintenance software for new requests. By the end of the software life cycle is critical, the down transaction is antagonistic.

Figure 2.9: Mobile application development life cycle



2.1.5.1 Planning

Planning is the starting point of the software life cycle. It is the period when the basic needs are identified, the feasibility studies for the project are made (defining the costs and benefits of the system) and the project planning is carried out.

2.1.5.2 Analysis Planning

In the analysis process, which is one of the most important stages of the software life cycle, all functions of the project are determined in detail. The purpose of this phase is to clarify the functions and precise requirements of the system and to document them in a specific format as a result. This work includes client, software engineer, system analyst, business analyst, product manager, etc. can be done by groups in which the roles come together. In situations where the needs are unclear, communication and collaboration between the software engineer and the customer should be much more. At this stage in various software development methodologies, usage document and test

plan documents can also be created, Briefly, in the analysis process all details of the project are revealed.

2.1.5.3 Design

Based on the project details revealed in the end of the analysis work, the project is divided into its components when necessary, the operations to be carried out within the project are determined and the project plan is created. In addition to the project plan, design document must be created. The design document contains project information (purpose, scope, etc.), system design information, design details, data model, user interface designs, UML diagrams. The intent of the design document is that the software developer should have a technical documentation to ensure that new software as they develop and be included in the project during and after the project process. With the completion of the requirements, system design phase begins. Software product design is the activity of determining the features, capabilities, ad interfaces of the software product to meet the needs and desires of the customer.

Two types of design are possible: architectural design and detailed design. Architectural design deals with the general structures of software modules and their interactions within the organization. As a result, architectural design documents are created. In the course of detailed design Architectural design documents are generally revised. The stage of analysis in which the requirements are determined is what the problem is. It should be remembered that all problems in the system will not be solved by the completion of the software product. Unfortunately, most of the time what you say is a design decision How to say it may be the need of the customer. It is necessary to pay attention to this situation. One of the most important techniques used in software design is abstraction. Abstraction is the simplification of certain features of objects, events and situations to facilitate the solution of problems. It simplifies the problem and allows us to focus on the most important parts. Modeling is the basic design tool and it is possible to talk about static and dynamic models. While the static model is used to express the unchanging aspects of the program during its operation (class and object models), the

dynamic model is used to express the behavior of the program during its operation (state and sequence diagrams).

2.1.5.4 Development

Planning, analysis and design of the process to be completed and to be determined in detail the stage of development of the project. After this point, no analysis should be done and the project should proceed in the planning phase created during the design phase. The coding phase begins with the design phase reaching a certain degree. The customer is the stage of programming the product to be delivered. It is always a must to write simple code that is high quality, easy to read and easy to maintain. Whether working in a company or developing individual projects, development should always be based on a specific coding quality standard (naming standards, comment line usage, repetitive codes, extreme similar functions, long methods, etc.). It is ensured that there is no difficulty in the maintenance of the code afterwards.

2.1.5.5 Testing

Before the software development is complete and the customer is present, beta testing is being done by the test team. The other important stage during coding and after coding is the test. If we move with the early test approach and have a test perspective from the analysis stage, our error rate and costs (time, money, prestige, etc.) will decrease. There are many different categories and depth of test types that can be applied according to the process and situation, such as unit tests, smoke tests, false test, acceptance tests, usage scenario tests, load tests, user acceptance test, pass man test, test automation. Once this phase is completed and if errors are corrected, the project is taken to the public.

2.1.5.6 Maintenance

The project is the process of eliminating any errors that may occur after the release, improving the software and adding new functions. In the course of this process, these requests are realized in accordance with the information coming from the users. After

all the test steps have been completed, the software product is released to the market and a delivery phase is performed. As a delivery outlet, the product alone is not enough. The user manual and version difference document must be created for end users. With the delivery, the maintenance phase also begins. There are different maintenance activities such as debugger, preventer, infrastructure optimizer, new feature generator for the product.

2.2 ANDROID

2.2.1 What is Android ?

Android is a mobile operating system developed by Google, based on the Linux kernel and designed primarily for touchscreen mobile devices such as smartphones and tablets. Android's user interface is mainly based on direct manipulation, using touch gestures that loosely correspond to real-world actions, such as swiping, tapping and pinching, to manipulate on-screen objects, along with a virtual keyboard for text input. In addition to touchscreen devices, Google has further developed Android TV for televisions, Android Auto for cars, and Android Wear for wrist watches, each with a specialized user interface. Variants of Android are also used on game consoles, digital cameras, PCs and other electronics.

Figure 2.10: Android logo



Initially developed by Android Inc., which Google bought in 2005, Android was unveiled in 2007, along with the founding of the Open Handset Alliance – a consortium of hardware, software, and telecommunication companies devoted to advancing open standards for mobile devices. Beginning with the first commercial Android device in September 2008, the operating system has gone through multiple major releases, with the current version being 8.0 "Oreo", released in August 2017. Android applications can be downloaded from the Google Play store, which features

over 2.7 million apps as of February 2017. Android has been the best-selling OS on tablets since 2013, and runs on the vast majority of smartphones. As of May 2017, Android has two billion monthly active users, and it has the largest installed base of any operating system.

Figure 2.11: Google logo



Android's source code is released by Google under an open source license, although most Android devices ultimately ship with a combination of free and open source and proprietary software, including proprietary software required for accessing Google services. Android is popular with technology companies that require a ready-made, low-cost and customizable operating system for high-tech devices. Its open nature has encouraged a large community of developers and enthusiasts to use the open-source code as a foundation for community-driven projects, which deliver updates to older devices, add new features for advanced users or bring Android to devices originally shipped with other operating systems. The extensive variation of hardware in Android devices causes significant delays for software upgrades, with new versions of the operating system and security patches typically taking months before reaching consumers, or sometimes not at all. The success of Android has made it a target for patent and copyright litigation between technology companies.

2.2.2 Android History

Android Inc. was founded in Palo Alto, California in October 2003 by Andy Rubin, Rich Miner, Nick Sears, and Chris White (Phone Arena 2011). Rubin described the Android project as "tremendous potential in developing smarter mobile devices that are more aware of its owner's location and preferences" (Elgin, Ben 2005). The early intentions of the company were to develop an advanced operating system for digital cameras, and this was the basis of its pitch to investors in April 2004. The company

then decided that the market for cameras was not large enough for its goals, and by five months later it had diverted its efforts and was pitching Android as a handset operating system that would rival Symbian and Microsoft Windows Mobile (Alabaster, Jav 2013).

Rubin had difficulty attracting investors early on, and Android was facing eviction from its office space. Steve Perlman, a close friend of Rubin, brought him \$10,000 in cash in an envelope, and shortly thereafter wired an undisclosed amount as seed funding. Perlman refused a stake in the company, and has stated "I did it because I believed in the thing, and I wanted to help Andy" (Eadicicco, Lisa 2015).

In July 2005, Google acquired Android Inc. for at least \$50 million (Manjoo, Farhad 2015). Its key employees, including Rubin, Miner and White, joined Google as part of the acquisition. Not much was known about the secretive Android at the time, with the company having provided few details other than that it was making software for mobile phones. At Google, the team led by Rubin developed a mobile device platform powered by the Linux kernel. Google marketed the platform to handset makers and carriers on the promise of providing a flexible, upgradeable system (Block, Ryan 2007). Google had "lined up a series of hardware components and software partners and signaled to carriers that it was open to various degrees of cooperation" (Delaney, Kevin 2007).

Speculation about Google's intention to enter the mobile communications market continued to build through December 2006 (McKay, Martha 2006). An early prototype had a close resemblance to a BlackBerry phone, with no touchscreen and a physical QWERTY keyboard, but the arrival of 2007's Apple iPhone meant that Android "had to go back to the drawing board". Google later changed its Android specification documents to state that "Touchscreens will be supported", although "the Product was designed with the presence of discrete physical buttons as an assumption, therefore a touchscreen cannot completely replace physical buttons" (Ziegler, Chris 2012). In September 2007, InformationWeek covered an Evaluateserve study reporting that Google had filed several patent applications in the area of mobile telephony (Claburn, Thomas 2007).

On November 5, 2007, the Open Handset Alliance, a consortium of technology companies including Google, device manufacturers such as HTC, Motorola and Samsung, wireless carriers such as Sprint and T-Mobile, and

chipset makers such as Qualcomm and Texas Instruments, unveiled itself, with a goal to develop "the first truly open and comprehensive platform for mobile devices" (Schonfeld, Erick 2007). The first commercially available smartphone running Android was the HTC Dream, also known as T-Mobile G1, announced on September 23, 2008 (Aamoth, Doug 2008).

Since 2008, Android has seen numerous updates which have incrementally improved the operating system, adding new features and fixing bugs in previous releases. Each major release is named in alphabetical order after a dessert or sugary treat, with the first few Android versions being called "Cupcake", "Donut", "Eclair", and "Froyo", respectively. During its announcement of Android KitKat in 2013, Google explained that "Since these devices make our lives so sweet, each Android version is named after a dessert", although a Google spokesperson told CNN in an interview that "It's kind of like an internal team thing, and we prefer to be a little bit how should I say a bit inscrutable in the matter, i will say" (Menon, Murali K. 2016).

In 2010, Google launched its Nexus series of devices, a lineup in which Google partnered with different device manufacturers to produce new devices and introduce new Android versions. The series was described as having "played a pivotal role in Android's history by introducing new software iterations and hardware standards across the board", and became known for its "bloat-free" software with "timely updates". At its developer conference in May 2013, Google announced a special version of the Samsung Galaxy S4, where, instead of using Samsung's own Android customization, the phone ran "stock Android" and was promised to receive new system updates fast. The device would become the start of the Google Play edition program, and was followed by other devices, including the HTC One Google Play edition, and Moto G Google Play edition (Cunningham, Andrew 2013). In 2015, Ars Technica wrote that "Earlier this week, the last of the Google Play edition Android phones in Google's online storefront were listed as "no longer available for sale"" and that "Now they're all gone, and it looks a whole lot like the program has wrapped up" (Chavez, Chris 2015).

From 2008 to 2013, Hugo Barra served as product spokesperson, representing Android at press conferences and Google I/O, Google's annual developer-focused conference.

He left Google in August 2013 to join Chinese phone maker Xiaomi (Orion, Egan 2013). Less than six months earlier, Google's then-CEO Larry Page announced in a blog post that Andy Rubin had moved from the Android division to take on new projects at Google, and that Sundar Pichai would become the new Android lead. Pichai himself would eventually switch positions, becoming the new CEO of Google in August 2015 following the company's restructure into the Alphabet conglomerate, making Hiroshi Lockheimer the new head of Android (Bergen, Mark 2015).

In June 2014, Google announced Android One, a set of "hardware reference models" that would "allow [device makers] to easily create high-quality phones at low costs", designed for consumers in developing countries. In September, Google announced the first set of Android One phones for release in India (Pichai, Sundar 2014). However, Recode reported in June 2015 that the project was "a disappointment", citing "reluctant consumers and manufacturing partners" and "misfires from the search company that has never quite cracked hardware". Plans to relaunch Android One surfaced in August 2015, with Africa announced as the next location for the program a week later. A report from The Information in January 2017 stated that Google is expanding its low-cost Android One program into the United States, although The Verge notes that the company will presumably not produce the actual devices itself (Bohn, Dieter 2017).

Google introduced the Pixel and Pixel XL smartphones in October 2016, marketed as being the first phones made by Google, and exclusively featured certain software features, such as the Google Assistant, before wider rollout. The Pixel phones replaced the Nexus series, with a new generation of Pixel phones launched in October 2017 (D'Orazio, Dante 2015).

2.2.3 Android Applications

Applications , which extend the functionality of devices, are written using the Android SDK and, often, the Java programming language. Java may be combined with C/C++, together with a choice of non-default runtimes that allow better C++ support. The Go programming language is also supported, although with a limited set of API. In May 2017, Google announced support for Android app development in the Kotlin programming language (Miller, Paul 2017).

The SDK includes a comprehensive set of development tools, including a debugger, software libraries, a handset emulator based on QEMU, documentation, sample code, and tutorials. Initially, Google's supported IDE was Eclipse using the ADT plugin; in December 2014, Google released Android Studio, based on IntelliJ IDEA, as its primary IDE for Android application development. Other development tools are available, including a NDK for applications or extensions in C or C++, Google App Inventor, a visual environment for novice programmers, and various cross platform mobile web applications frameworks. In January 2014, Google unveiled an framework based on Apache Cordova for porting Chrome HTML 5 web applications to Android, wrapped in a native application Shell (Opam, Kwame 2014).

Android has a growing selection of third-party applications, which can be acquired by users by downloading and installing the application's APK file, or by downloading them using an application store program that allows users to install, update, and remove applications from their devices. Google Play Store is the primary application store installed on Android devices that comply with Google's compatibility requirements and license the Google Mobile Services software. Google Play Store allows users to browse, download and update applications published by Google and third-party developers; as of July 2013, there are more than one million applications available for Android in Play Store. As of July 2013, 50 billion applications have been installed. Some carriers offer direct carrier billing for Google Play application purchases, where the cost of the application is added to the user's monthly bill (Chu, Eric 2011). As of May 2017, there are over one billion active users a month for Gmail, Android, Chrome, Google Play and Maps.

Due to the open nature of Android, a number of third-party application marketplaces also exist for Android, either to provide a substitute for devices that are not allowed to ship with Google Play Store, provide applications that cannot be offered on Google Play Store due to policy violations, or for other reasons. Examples of these third-party stores have included the Amazon Appstore, GetJar, and SlideMe. F-Droid, another alternative marketplace, seeks to only provide applications that are distributed under free and open source licenses (Ganapati, Priya 2010).

2.2.4 Platform Usage

Charts in this section provide breakdowns of Android versions, based on devices accessing the Google Play Store in a seven-day period ending on October 2, 2017 (just after release of Android Oreo). Therefore, these statistics exclude devices running various Android forks that do not access the Google Play Store, such as Amazon's Fire tablets. (Android Developers Dashboard 2017)

Figure 2.12: Android version usages

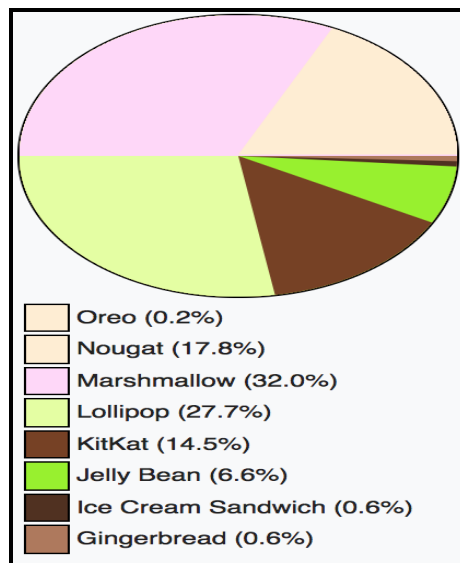


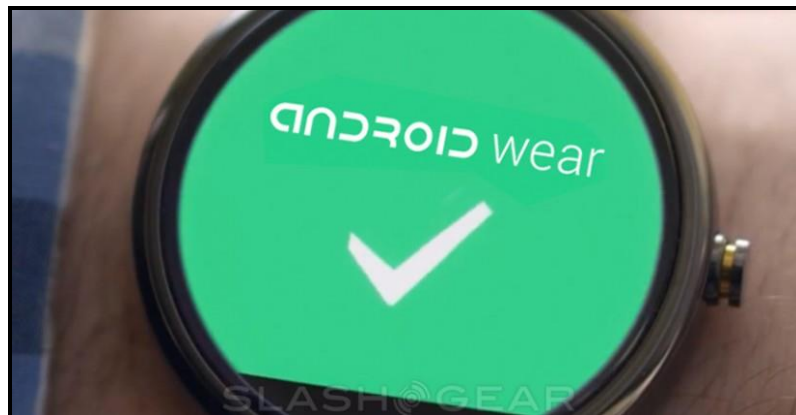
Table 2.1: Android version usages

Version ↕	Code name ↕	Release date ↕	API level ↕	DVM/ART ↕	Distribution ↕	First devices to run version ↕
8.1	Oreo	October 25, 2017	27	ART		Pixel, Pixel XL, Nexus 6P
8.0		August 21, 2017	26	ART	0.2%	N/A
7.1	Nougat	October 4, 2016	25	ART	2.0%	Pixel, Pixel XL
7.0		August 22, 2016	24	ART	15.8%	Nexus 5X, Nexus 6P
6.0	Marshmallow	October 5, 2015	23	ART	32.0%	
5.1	Lollipop	March 9, 2015	22	ART	21.0%	Android One
5.0		November 3, 2014	21	ART 2.1.0	6.7%	Nexus 6, Nexus 9
4.4	KitKat	October 31, 2013	19	DVM (and ART 1.6.0)	14.5%	Nexus 5
4.3	Jelly Bean	July 24, 2013	18	DVM	1.0%	Nexus 7 2013
4.2		November 13, 2012	17	DVM	3.3%	Nexus 4, Nexus 10
4.1		July 9, 2012	16	DVM	2.3%	Nexus 7
4.0	Ice Cream Sandwich	October 19, 2011	15	DVM	0.6%	Galaxy Nexus
2.3	Gingerbread	February 9, 2011	10	DVM 1.4.0	0.6%	Nexus S

2.2.5 Other Usages

Google has developed several variations of Android for specific use cases, including Android Wear for wearable devices such as wrist watches, Android TV for televisions, Android Auto for cars, and Brillo, later renamed Android Things, for smart devices and Internet of things (Kastrenakes, Jacob 2016).

Figure 2.13: Android wear



The open and customizable nature of Android allows device makers to use it on other electronics as well, including laptops and netbooks, smartbooks, e-books, desktop computers, cameras, headphones, home automation systems, game consoles, mirrors, media players, landline telephones, feature phones, electronic dictionaries, routers, alarm clocks, printers, refrigerators, coffee machines, payment terminals, automated teller machines, bicycles, Panel PCs, and robots. Additionally, android have been installed and run on a variety of objects, including calculators, single-board computers, and satellites (Petrovan, Bogdan 2012).

Figure 2.14: Android TV



Ouya, a video game console running Android, became one of the most successful Kickstarter campaigns, crowdfunding US\$8.5m for its development, and was later followed by other Android-based consoles, such as Nvidia's Shield Portable an Android device in a video game controller form factor (Buckley, Sean 2013).

In 2011, Google demonstrated "Android@Home", a home automation technology which uses Android to control a range of household devices including light switches, power sockets and thermostats. Prototype light bulbs were announced that could be controlled from an Android phone or tablet, but Android head Andy Rubin was cautious to note that "turning a lightbulb on and off is nothing new", pointing to numerous failed home automation services. Google, he said, was thinking more ambitiously and the intention was to use their position as a cloud services provider to bring Google products into customers' homes (Patel, Nilay 2012).

Parrot unveiled an Android-based car stereo system known as Asteroid in 2011, followed by a successor, the touchscreen-based Asteroid Smart, in 2012. In 2013, Clarion released its own Android-based car stereo, the AX1 (Low, Aloysius 2013). In January 2014, at the Consumer Electronics Show (CES), Google announced the formation of the Open Automotive Alliance, a group including several major automobile makers (Audi, General Motors, Hyundai, and Honda) and Nvidia, which aims to produce Android-based in car entertainment systems for automobiles, "[bringing] the best of Android into the automobile in a safe and seamless way" (Souppouris, Aaron 2014).

Android comes preinstalled on a few laptops (a similar functionality of running Android applications is also available in Google's Chrome OS) and can also be installed on personal computers by end users. On those platforms Android provides additional functionality for physical keyboards and mice, together with the "Alt-Tab" key combination for switching applications quickly with a keyboard. In December 2014, one reviewer commented that Android's notification system is "vastly more complete and robust than in most environments" and that Android is "absolutely usable" as one's primary desktop operating system (Lunduke, Bryan 2014).

In October 2015, The Wall Street Journal reported that Android will serve as Google's future main laptop operating system, with the plan to fold Chrome OS into it by 2017 (Tran, Sam 2015). Google's Sundar Pichai, who led the development of Android, explained that "mobile as a computing paradigm is eventually going to blend with what we think of as desktop today." and back in 2009, Google co-founder Sergey Brin himself said that Chrome OS and Android would "likely converge over time." Lockheimer, who replaced Pichai as head of Android and Chrome OS, responded to this claim with an official Google blog post stating that "While we've been working on ways to bring together the best of both operating systems, there's no plan to phase out Chrome OS [which has] guaranteed auto-updates for five years". That is unlike Android where support is shorter with "EOL dates at least 3 years [into the future] for Android tablets for education" (Lockheimer, Hiroshi 2015).

2.3 MEGAFORMER

2.3.1 What is Megaformer ?

Megaformer is the machine that is used in all of pilates workouts. Its patented design enables customers to take advantage of a wide variety of exercises that build strength, endurance, flexibility, balance and mental toughness. Judging solely on the desing of the machine, customers can begin to see why a Megaformer workout is not like an average Pilates class. During a single megaformer class people can burn up to 800 calories but during an ordinary class people can burn up to 275 calories.

Figure 2.15: Megaformer machine



2.3.2 Lagree Fitness & Method

Lagree Fitness is comprised of 2 words: Lagree and Fitness

Lagree is last name of the creator of Megaformer and Lagree Fitness and Method whose full name is Sebastien Lagree. This method and the development of it is very personal. Everything people are learning here is based on over 30 years of his research, experiences, and interactions to find uncover the most efficient, effective way of exercise.

Figure 2.16: Sebastien Lagree



The second word is Fitness. The 5 main components of Physical Fitness are Muscular Strength, Muscular Endurance, Cardiorespiratory Health, Body Composition, and Flexibility. These attributes, however very important, do not paint a complete picture of what else fitness can encompass.

He states that: “My mission has been to target every aspect of physical fitness, and complete the remaining elements to create Lagree Fitness. The Lagree Method is how you apply all of these elements in a routine. Most importantly, the Lagree Method does not only improve your body, but also your mind”.

Lagree Fitness is a challenging workout. It is an efficient, full-body sculpting, and fat-burning workout. It is intense and low-impact. Lagree Fitness is designed to test and improve customer’s Cardio, Strength, Endurance, Core, Balance, and Flexibility. Customers should expect to sweat and feel their muscles shake. It is a workout that leaves their feeling energized and stronger.

Lagree Fitness is also taught in a very specific way. The Lagree Method focuses on six key Training Elements to effectively stimulate the muscle fibers. The key term here: Effective. In order for something to be effective, it simply means it works! And when it comes to training the body, they need the precise amount of intensity to encourage and inspire change. These six key training elements are:

a- Effective Tempo

Why is speed such an important factor in Lagree? How does slow-twitch muscle fiber training benefit us?

b- Effective Duration

How long does it take for an exercise to be effective? How can we train both aerobically and anaerobically?

c- Effective Range of Motion

What is the Range of Motion for each exercise? Is there a difference between the range of motion of a muscle vs the range of motion of the movement?

d- Effective Resistance

How much resistance do you need to trigger a positive adaptive response? How does variable tension allowing us to train a wide variety of ability levels?

e- Effective Angle/Body Alignment

Body alignment/angle is the optimal placement of the hands and feet to optimize muscle fiber recruitment and minimize any excessive stress on the joints and the spine. What angle should you perform an exercise at?

f- Effective Planes of Motion

Effective Planes of Motion deal primarily with the Supra, so you will not study this in this course. To really understand about planes of motions, you have to learn how your body responds to exercise if it is up, tilted, or a combination of the two.

Figure 2.17: Example megaformer fitness studio



Figure 2.18: Studio members during workout



2.3.3 Articles that are written about Megaformer

There are some articles that are written about “Megaformer”. Some journalists visited and experienced megaformer.

“The New York Times” journalist Marikar (2014) states that “It brings to Pilates what Pilates was lacking,” “I always wanted to do Pilates, but I felt like I needed to do another workout after. I didn’t sweat”. “This workout is really intense and works so many different muscles at the same time”.

According to the traveller Kolafa (2016) “The megaformer class satisfies that blood-pumping-music-blaring-sweat-dripping gym rat feeling”.

3. DATA AND METHOD

3.1 SYSTEM ANALYSIS

Before implementing this project, problems and needs of the Megaformer studios are researched and as a results of this research it is decided that an information system which will be a solution for this problems should be implemented. During this research a comprehensive system analysis is implemented.

As system analysis method “Moder Driven Analysis” approach is used during the system analysis of this project.

As the first phase problems are determined and following problems determined for megaformer studios.

- a- Reservation system is not working stable and not easy to use
 - a. Megaformer studios works with reservation system. Each customer makes reservation before lessons but this reservation system only works by calling studio or from website (which is not working stable and not available on all studios).
- b- Monitoring customer’s diet plan is not available via a system
 - a. Trainers are not able to monitor customer’s diet plan.
- c- Customer’s measurements are not saved in a system
 - a. Each club member makes a measurement each month to see fat and muscle weight etc. but these datas are not saved on a system that’s why each customer needs to save their measurement data their selves.
- d- Customer’s sport programmes are not saved in a system
 - a. Trainers prepare sport programme for each member but these programmes are not saved on a system that’s why each customer needs to save their sport programmes their selves.
- e- Announcements are sent to customers manually
 - a. Announcements of the studio are sent to customers manually (without a system).
- f- Customers are able to buy packages in studio

- a. There is no system to buy package outside of the studio.
- g- Customer's are not able to access their trainers easily
 - a. Customers can only message with their trainers by taking their mobile phones. (Some of the members do not want to share their mobile phones with trainers).

After problems are detected, requirements of the new system are determined.

- a- There should be a stable reservation system that is easy to use and have a good performance.
- b- Customers should be able track/update their diet plans. Trainers should be able to track customer's diet plans. There should be detailed information about nutrition contents (protein, fat and carbonhydrate proportion etc.).
- c- Customers should be able to track their measurements in new system. There should be detailed informations of each measurement. Muscle rate and fat rate of each body member should be shown.
- d- Sport programmes should be accessible by customers easily. Detailed decription of each sport programme should be shown to customers via new system. Instead of writing programmes to papers (which is not safe to keep), programmes should accessible via more sheltered system.
- e- Announcements system should be binded to an automated system. Instead of sending announcement to each customer one by one, new announcement should be sent to all of the customers by a single move. Also customers (who may missed the announcement) should be able to access previous announcements.
- f- Since stable reservation system needs to be implemented. Package buying system should also be changed. Customers should be able to buy packages via new system (outside of the studio).
- g- Due to information security and personal reasons some customers do not want to share their mobile phones with their trainers. As a results of this situations, in new system customers should be able access their trainers via system without sharing their mobile phones.

After requirements of the new system is defined. Solution system (which will contain the requirements of the new system) is determined. A mobile application is a solution for all of these problems.

3.1.1 Use – Case Modelling

In use-case modelling part, actors and use-cases of the application are determined and detailed.

3.3.1.1 Actor Glossary

Table 3.1: Actors of application

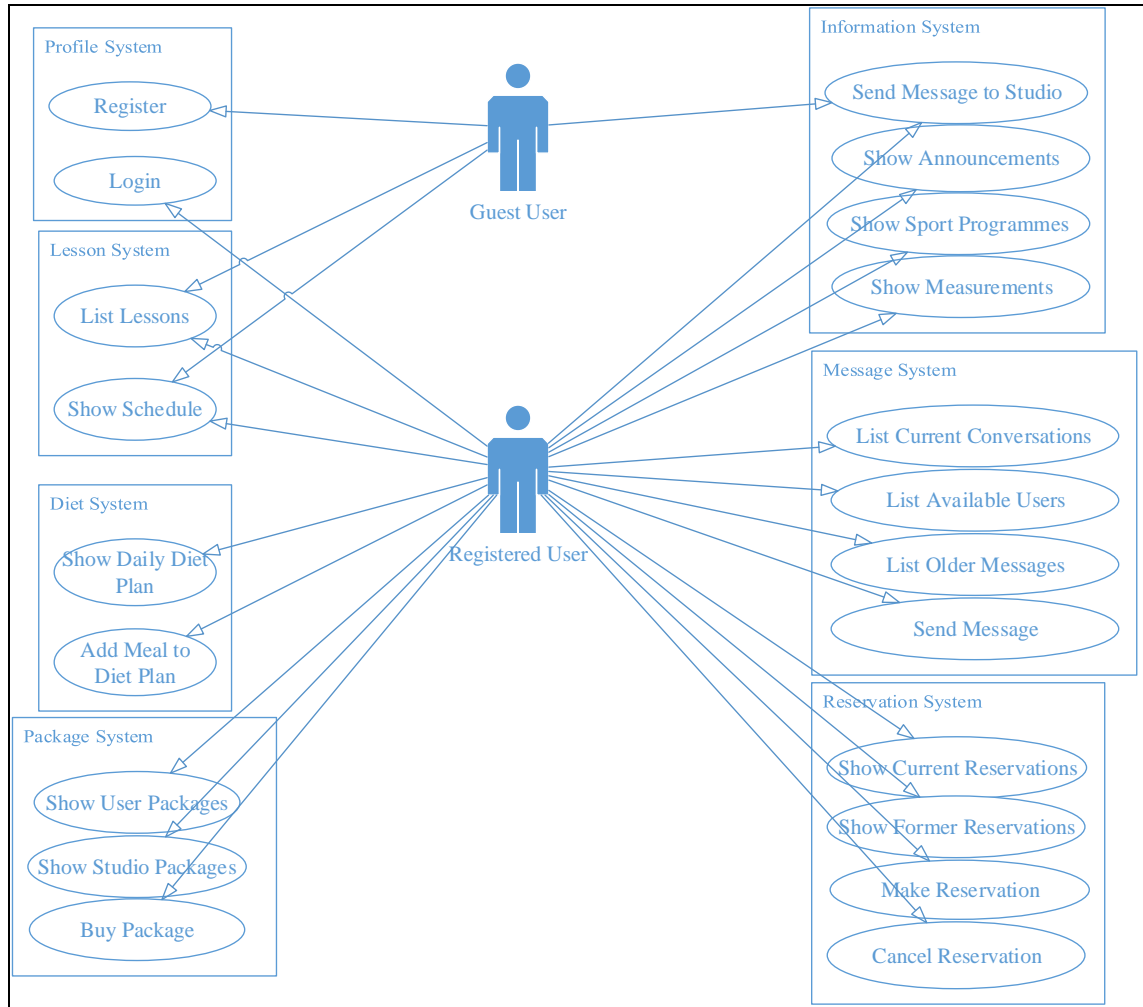
Term	Description
Registered User	A person who is registered to studio.
Guest User	A person who is not registered to studio.

3.3.1.2 Use – Case Glossary

All of the use-cases of application (use-case name, use-case description and participating actors and roles) are shown in Appendix A.1.1.

3.3.1.3 Use – Case Modelling Diagram

Figure 3.1: Use-Case modelling diagram



3.3.1.4 Use – Case Narratives

All of the use-case narratives of the application are shown in Appendix A.1.2. Each use-case is described detailly.

3.1.2 Data Modelling

In data modelling part, activity and sequence diagrams are drawn in order to detect the flow of each use-case.

3.1.2.1 Activity Diagrams

All of the activity diagrams of the application are shown in Appendix A.2. Each use-case flow is drawn detailly.

3.1.2.2 Sequence Diagrams

All of the sequence diagrams of the application are shown in Appendix A.3. Each use-case flow between user, android client and server is drawn detailly.

3.2 MEGAFORMER FITNESS STUDIO ANDROID APPLICATION

Megaformer fitness studio android application is implemented in order to provide all of the required information about the studio to their customers. This application is a solution for the problems that are occurred on other megaformer studios.

Figure 3.2: Megaformer app home page

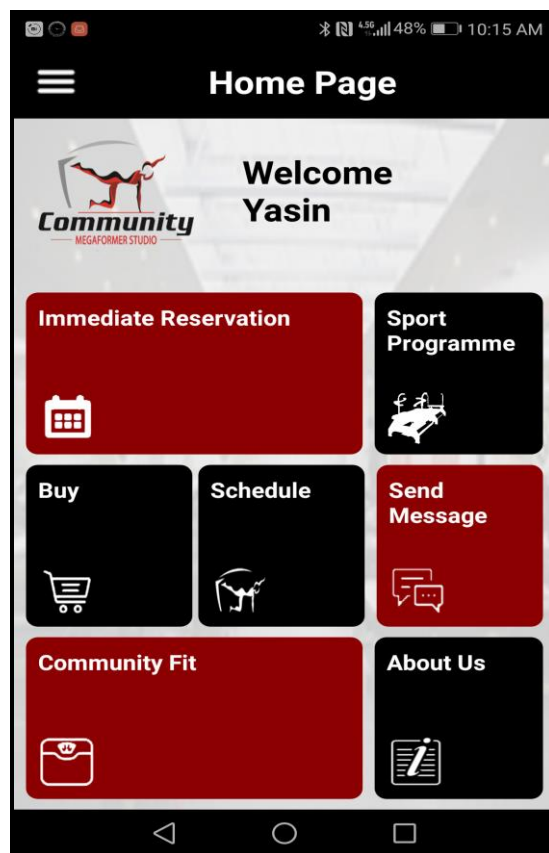
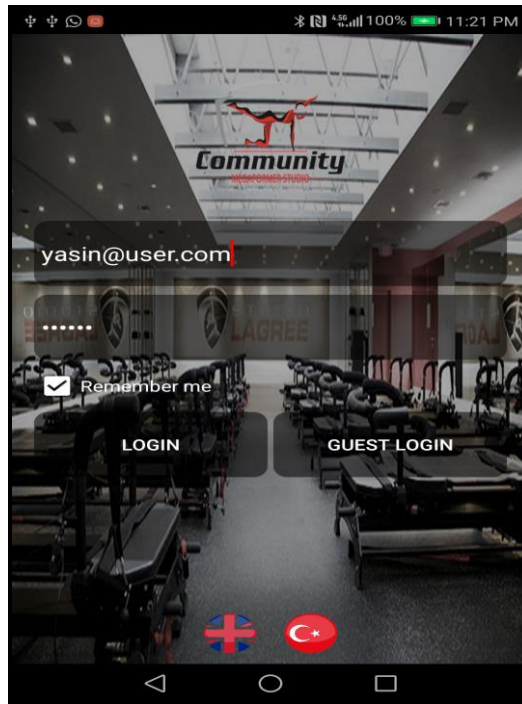


Figure 3.3: App login page



3.2.1 Megaformer Fitness Studio Android Application Features

a- Lessons and details are easy to access. Members all guest users are able to get informations about lessons and details (exercises etc.).

Figure 3.4: App lessons page

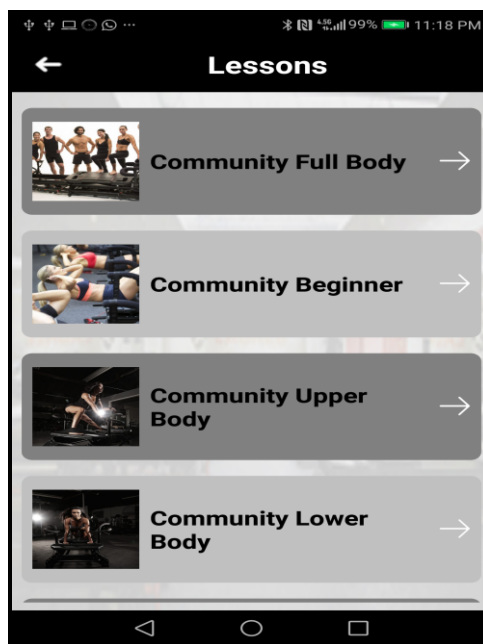


Figure 3.5: App lesson detail page



b- Members are able to make reservations via application, list their current and older reservations.

Figure 3.6: App my reservations page

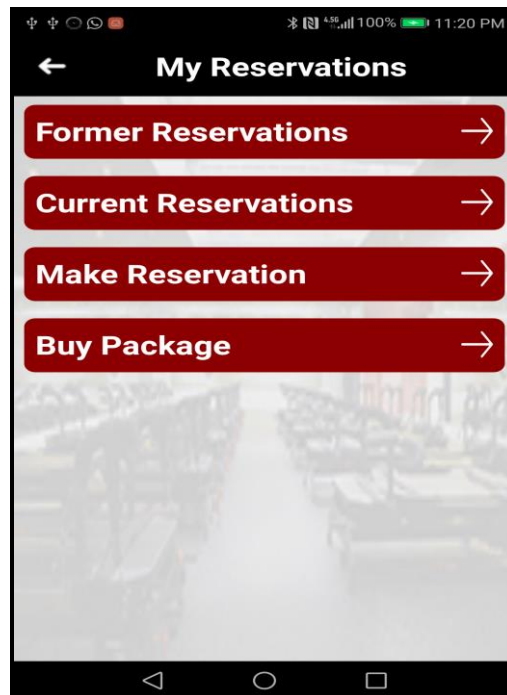


Figure 3.7: App personal reservations page

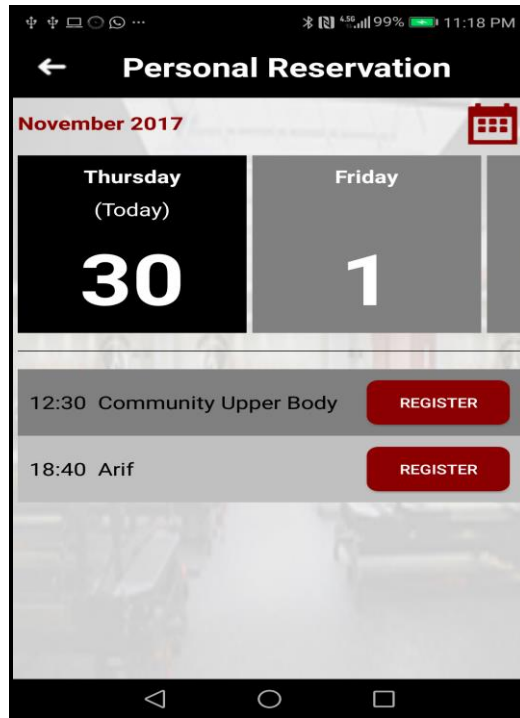


Figure 3.8: App older reservations page

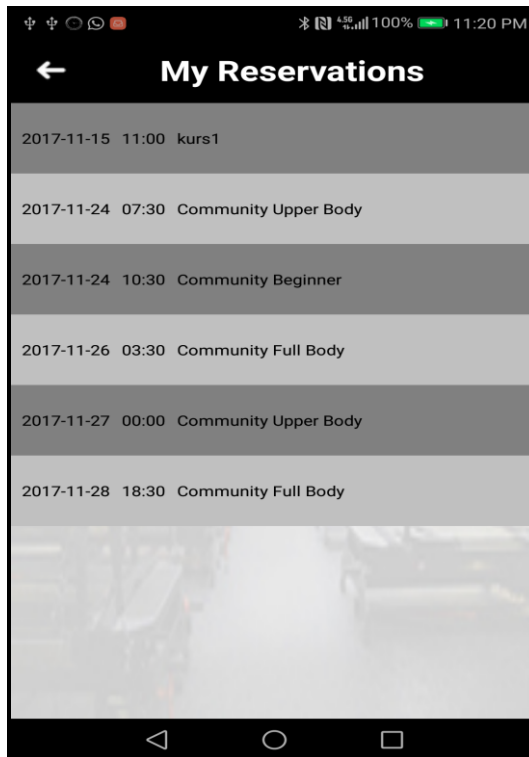
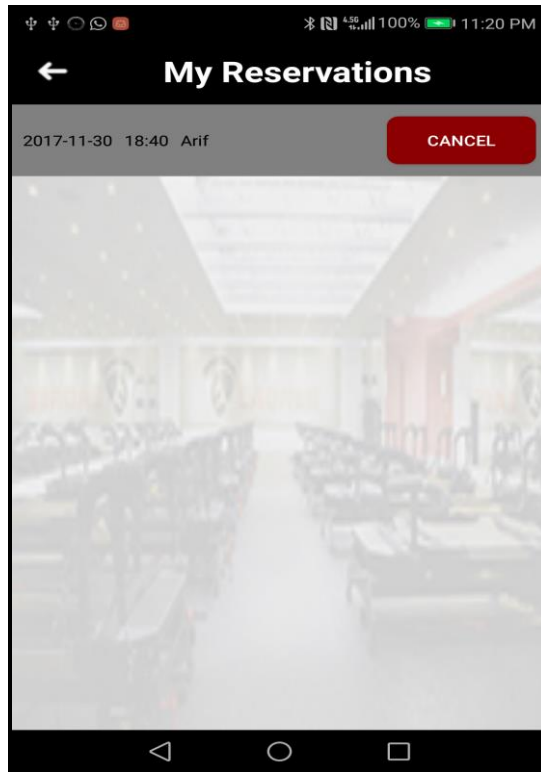
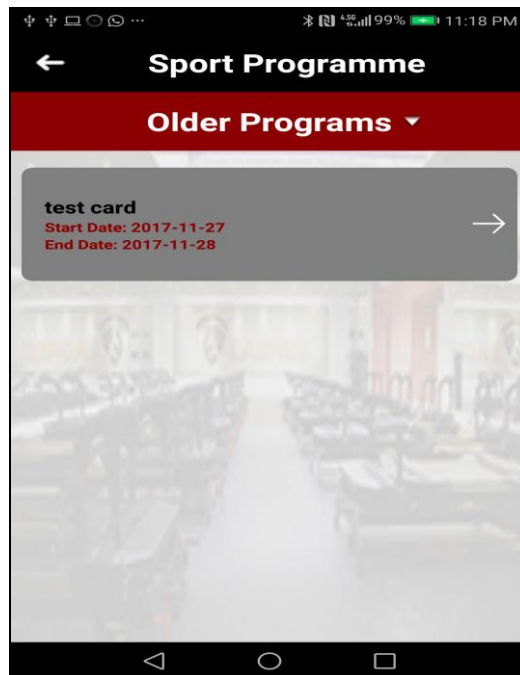


Figure 3.9: App current reservations page



c- Users are able to see their current and older sport programmes

Figure 3.10: App sport programmes page



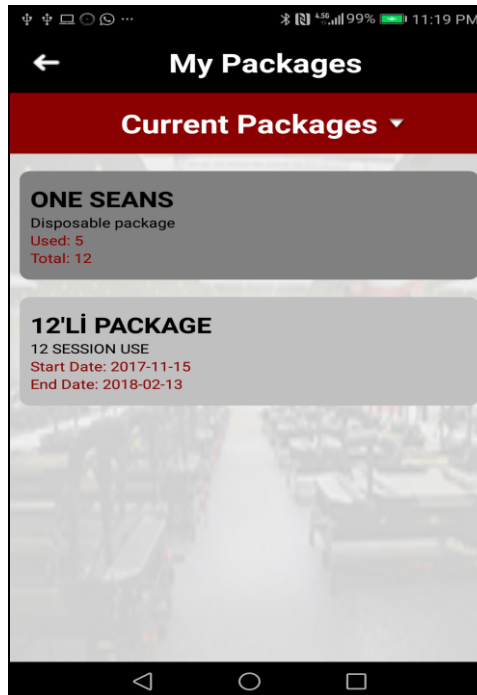
d- Users are able to see their measurements

Figure 3.11: App measurements page



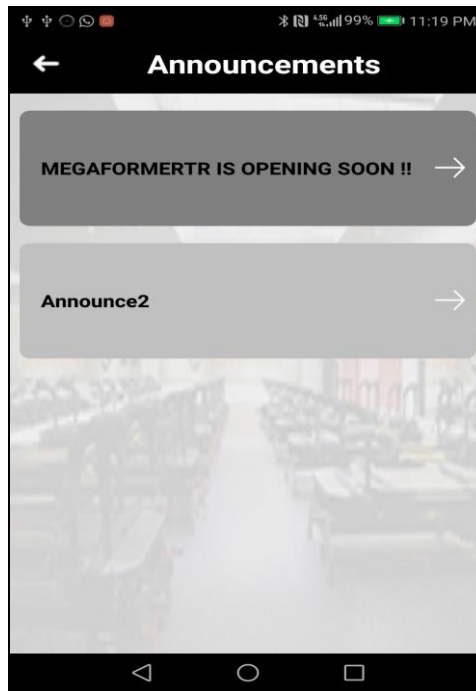
e- Users are able to see their current and older packages

Figure 3.12: App packages page



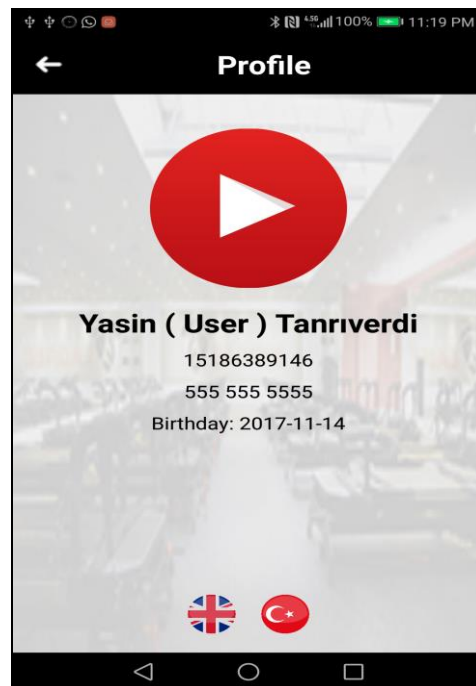
f- Users are able to see announcement that are sent from studio

Figure 3.13: App announcements page



g- Users are able to see their profile information, change their profile picture and change application language.

Figure 3.14: App profile page



h- Users are able to monitor their diet plan

Figure 3.15: App community fit page



i- Users are able to buy packages

Figure 3.16: App shopping page

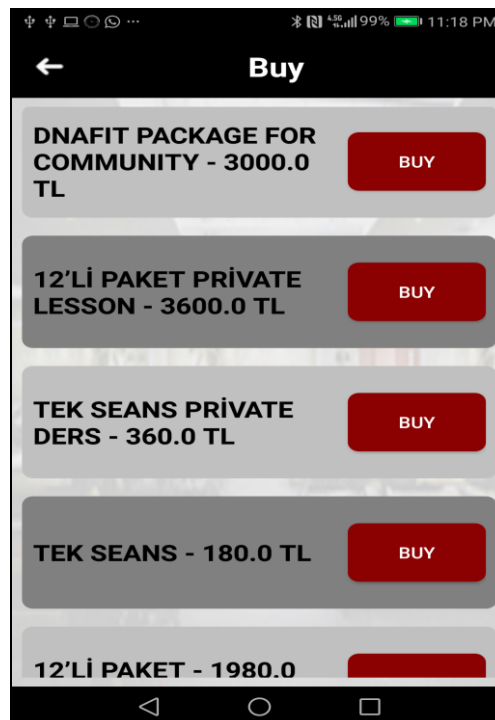
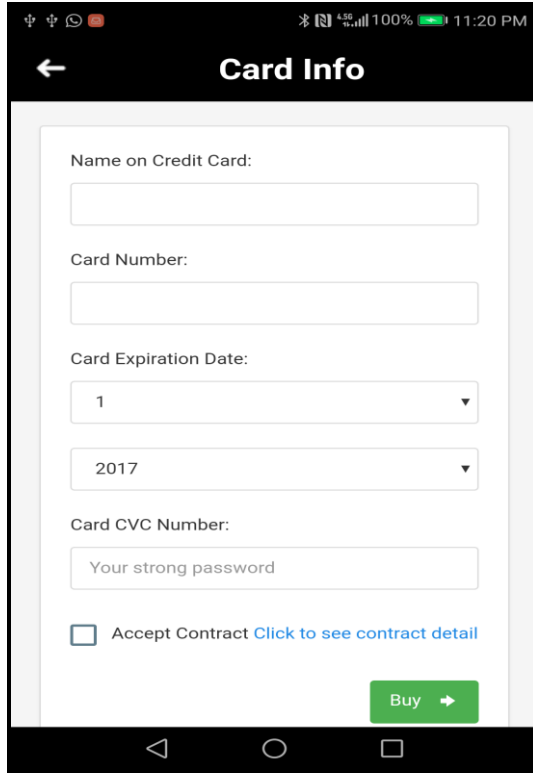


Figure 3.17: App card info page



j- Users are able to message with their trainers

Figure 3.18: App messages page

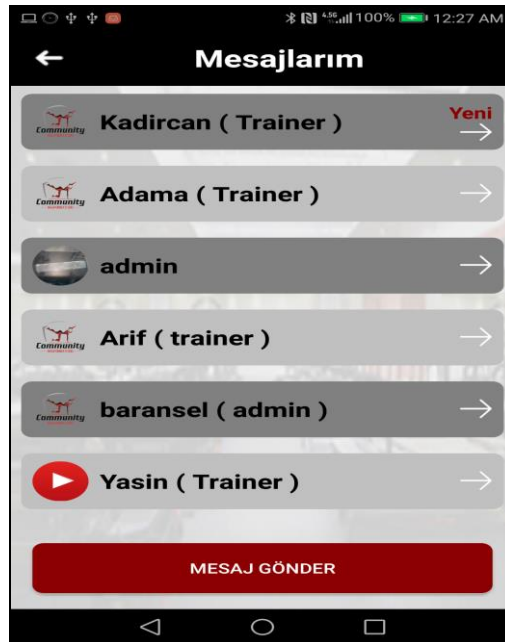
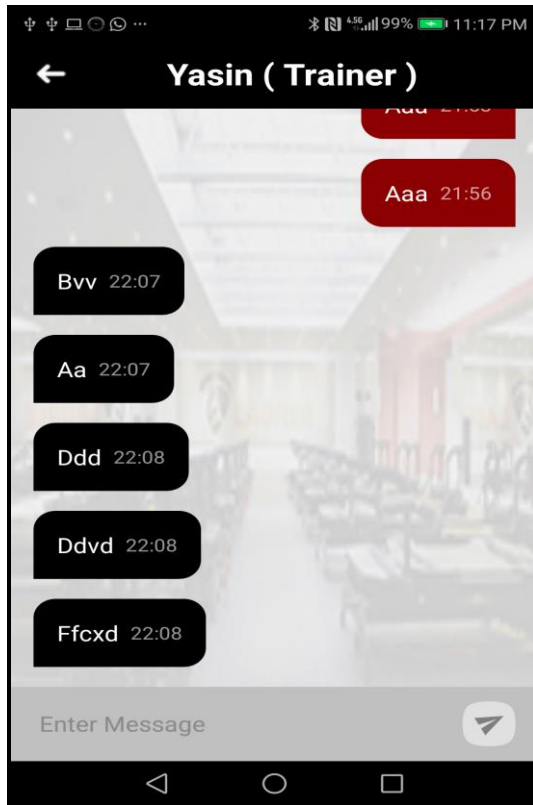


Figure 3.19: App messaging page

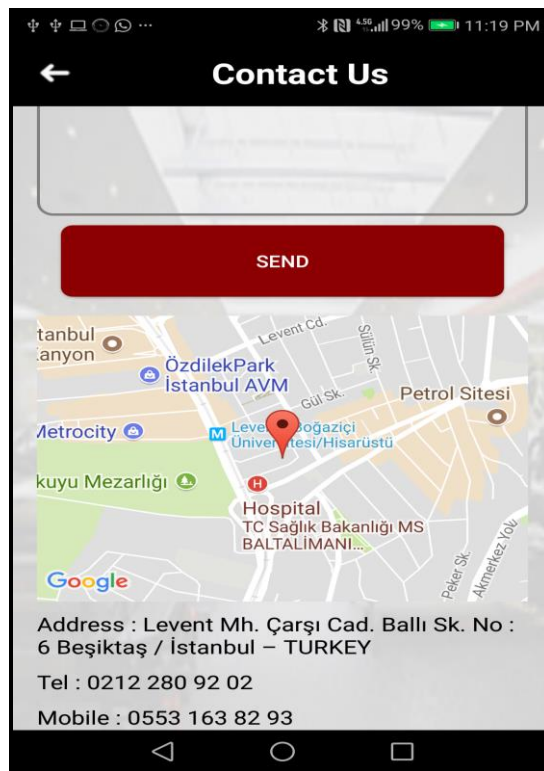


k- All users are able to send message to system

Figure 3.20: App contact page



Figure 3.21: App contact page map



1- Guest users are able to sign up to system

Figure 3.22: App guest user home page

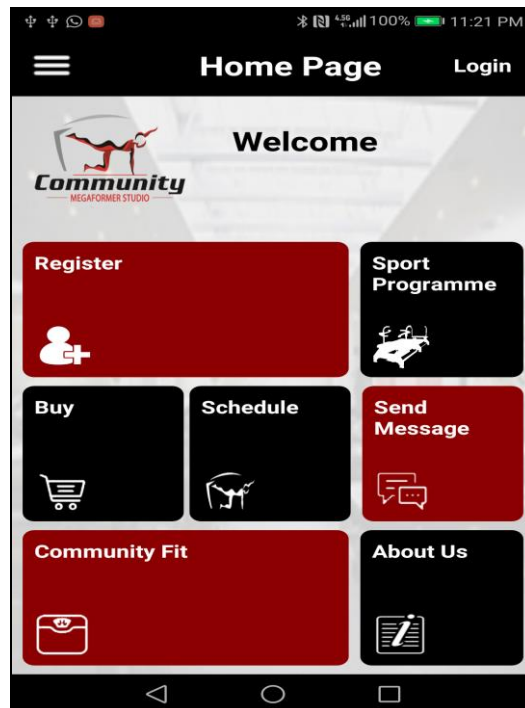
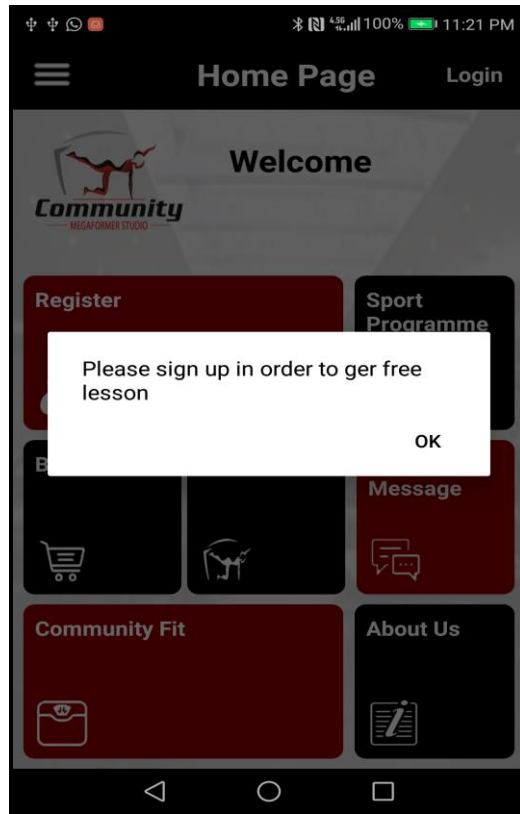


Figure 3.23: App guest user message



3.3 SCHEDULE OF MEGAFORMER FITNESS STUDIO ANDROID APP

3.3.1 GANTT Chart

Table 3.2: GANTT chart

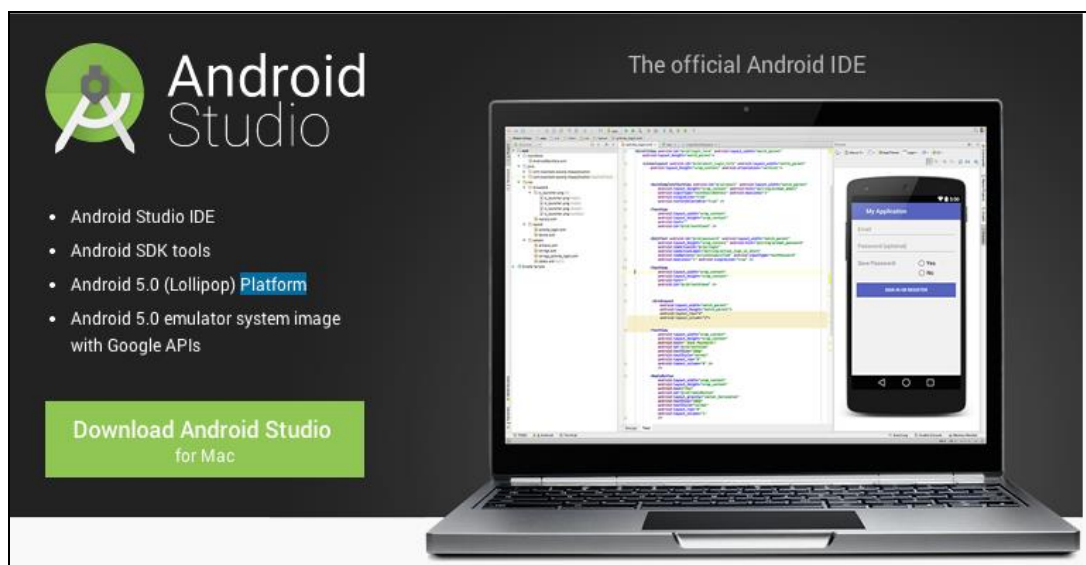
ID	Task Name	Start	Finish	Duration	Eki 2017					Kas 2017				
					24.9	1.10	8.10	15.10	22.10	29.10	5.11	12.11	19.11	26.11
1	Planning	25.09.2017	02.10.2017	6d	[Gantt bar]									
2	Data & System Analysis	03.10.2017	10.10.2017	6d	[Gantt bar]									
3	Design	11.10.2017	18.10.2017	6d	[Gantt bar]									
4	Development	18.10.2017	22.11.2017	26d	[Gantt bar]									
5	Testing	23.11.2017	01.12.2017	7d	[Gantt bar]									

3.4 MEGAFORMER FITNESS STUDIO ANDROID APP TECHNICAL DETAILS

3.4.1 Used Materials and Services During Project

During development of this android application “Android Studio” as used as IDE which is developed by “Jet Brains” company and supported by “Google”.

Figure 3.24: Android studio IDE



GCM is used in order to send notifications to customers.

Figure 3.25: GCM logo



Iyzico payment system is used during package shopping.

Figure 3.26: Iyzico logo



3.4.2 Process Modelling

3.4.2.1 Context Level Data Flow Diagram

Context level data flow diagram (between user and server) of the application is shown in Appendix A.4.1.

3.4.2.2 Level – 0 Data Flow Diagram

Level-0 data flow diagram (between user and systems) of the application is shown in Appendix A.4.2.

3.4.3 Class Modelling

3.4.3.1 UML Class Diagram

UML class diagram of the application is shown in Appendix A.5.1.

4. FINDINGS

As shown by application figures in previous section, by this application all of the information flow of the megaformer fitness studio is handled by mobile application. All of the problems that are occurred on other megaformer studios are solved thanks to mobile application.

- a- Users are able make reservations by their mobile phone
- b- Users are able to track their measurements via application
- c- Users are able to buy package via application
- d- Users are able to track their diet plan
- e- Studio can send notifications to their customer's mobile phones automatically
- f- Users are able to contact with their trainers via application.

These stated problems were the main problems that customers encountered, as addition to these problems many of the other problems are solved

5.DISCUSSION AND CONCLUSION

As stated above, encountered problems on other megaformer studios are solved by the usage of this application.

Thanks to well-designed system analysis all of the requirements (which are need to be implemented) are done via this information system. The main reason behind the success of this application is, simple usage of the system and quick customer access to the desired informations.

According to this result, before implementing an information system on any area, system analysis part should be implemented carefully. Problems and requirements should be clear. Solutions should be simple and quick for customer usage since by the increasing usage of smart phones, customers desire simple and quick systems.

If this application becomes successful for Istanbul branch of the Megaformer fitness studio, this application will be used by about all of the megaformer studios around the world.

REFERENCES

- [1] <https://www.lagreefitness.com>
- [2] <http://florafoodie.com/2016/08/megaformer-review.html>
- [3] https://www.tutorialspoint.com/sdlc/sdlc_overview.htm
- [4] <https://www.nytimes.com/2014/09/21/fashion/sebastien-lagrees-megaformer-machines-put-the-sweat-in-pilates.html>
- [5] [https://en.wikipedia.org/wiki/Android_\(operating_system\)](https://en.wikipedia.org/wiki/Android_(operating_system))
- [6] <http://www.mobillium.com/blog/tag/mobil-uygulama-gelistirme/>
- [7] Janice C. Sipior (2018) From the Editor, Information Systems Management, 35:1, 1-1
- [8] Ramakrishnan S (2012) System Analysis and Design., Inform Tech Soft Engg S8:e001

APPENDICES

Appendix A.1 USE-CASE MODELLING

Appendix A.1.1 Use-Case Glossary

Table A.1: Use cases of application

Use Case Name	Use Case Description	Participating Actors and Roles
Login	This use case describes login operation to application	Registered User (primary business actor)
Register	This use case describes register operation to application	Guest User (primary business actor)
List Lessons	This use case describes listing lessons of studio	Registered User (primary business actor) Guest User (primary business actor)
Show Lesson Detail	This use case describes showing detail of a lesson.	Registered User (primary business actor) Guest User (primary business actor)
Show Schedule	This use case describes showing schedule of the studio.	Registered User (primary business actor) Guest User (primary business actor)
Make Reservation	This use case describes making a reservation for a	Registered User (primary business actor)

	lesson.	
Cancel Reservation	This use case describes canceling a reservation for a lesson.	Registered User (primary business actor)
Show Sport Programmes	This use case describes showing sport programmes of a registered user.	Registered User (primary business actor)
Show Measurements	This use case describes showing measurements of a registered user.	Registered User (primary business actor)
Show User Packages	This use case describes showing packages of a registered user.	Registered User (primary business actor)
Show Current Reservations	This use case describes showing current reservations of a registered user.	Registered User (primary business actor)
Show Older Reservations	This use case describes showing older reservations of a registered user.	Registered User (primary business actor)
Show Announcements	This use case describes showing announcements of studio.	Registered User (primary business actor)
Show Profile	This use case describes showing profile of studio.	Registered User (primary business actor)
Change Language	This use case describes changing language of application and profile.	Registered User (primary business actor)

Show Studio Packages	This use case describes showing available packages of studio to buy.	Registered User (primary business actor)
Show Diet Plan	This use case describes showing diet plan of registered user.	Registered User (primary business actor)
Show Studio Informations	This use case describes showing studio informations.	Registered User (primary business actor) Guest User (primary business actor)
Buy Package	This use case describes buying a package.	Registered User (primary business actor)
Update Profile Image	This use case describes updating profile image of registered user.	Registered User (primary business actor)
Add Meal	This use case describes adding a meal to daily diet plan.	Registered User (primary business actor)
List Current Conversations	This use case describes listing current conversations of a registered user.	Registered User (primary business actor)
List Available Users	This use case describes listing available users message can be sent.	Registered User (primary business actor)
List Older Messages	This use case describes listing older messages between registered user and another user.	Registered User (primary business actor)

Send Message	This use case describes sending message from registered user to another user.	Registered User (primary business actor)
Send Message to Studio	This use case describes sending message to studio.	Registered User (primary business actor) Guest User (primary business actor)

Appendix A.1.2 Use-Case Narratives

Table A.2: Login use-case narrative

Use-Case Name:	Login	Use-Case Type Business Requirements:
Use-Case ID:	MGF-UC001.00	
Priority:	High	
Source:	Requirement-MGF-R1.00	
Primary Business Actor:	Registered User (primary business actor)	
Other Participating Actors:		
Other Interested Stakeholder:	System Owner	
Description:	This use case describes login operation to application.	
Precondition:	The user must be registered to application.	
Trigger:	This use case initiated when user clicks login button on login page.	
Typical Course of Events:	Actor Action	System Response
	Step 1: User enters username and password Step 2: If validation is successful user enters to application.	Alt-Step 1: If validation is not successful error message is shown to user.
Conclusion:	This use case concludes when user logs in to application or error message shown to user.	
Postcondition:	Users enters to application if validation is successful.	

Table A.3: Register use-case narrative

Use-Case Name:	Register	Use-Case Type Business Requirements:
Use-Case ID:	MGF-UC002.00	
Priority:	High	
Source:	Requirement-MGF-R2.00	
Primary Business Actor:	Guest User (primary business actor)	
Other Participating Actors:		
Other Interested Stakeholder:	System Owner	
Description:	This use case describes register operation to application.	
Precondition:	The user must be installed the application.	
Trigger:	This use case initiated when guest user clicks submit button on register page.	
Typical Course of Events:	Actor Action	System Response
	Step 1: Guest user enters required information Step 2: If validation is successful user enters the application as registered user.	Alt-Step 1: If validation is not successful error message is shown to user.
Conclusion:	This use case concludes when user logs in to application as registered user or by showing error message to user.	
Postcondition:	Users enters to application as registered user if validation is successful.	

Table A.4: List lessons use-case narrative

Use-Case Name:	List Lessons	Use-Case Type Business Requirements:
Use-Case ID:	MGF-UC003.00	
Priority:	High	
Source:	Requirement-MGF-R3.00	
Primary Business Actor:	Registered User (primary business actor) Guest User (primary business actor)	
Other Participating Actors:		
Other Interested Stakeholder:	System Owner	
Description:	This use case describes listing lessons of studio.	
Precondition:	The user must be installed the application.	
Trigger:	This use case initiated when users click lessons button on navigation menu.	
Typical Course of Events:	Actor Action	System Response
	Step 1: Click lessons button Step 2: Lessons are listed to users.	Alt-Step 1: If error is occurred, no results message is shown.
Conclusion:	This use case concludes when response is gathered from server.	
Postcondition:	Lessons are listed to user or no results message is shown.	

Table A.5: Show lesson detail use-case narrative

Use-Case Name:	Show Lesson Detail	Use-Case Type Business Requirements:
Use-Case ID:	MGF-UC004.00	
Priority:	High	
Source:	Requirement-MGF-R4.00	
Primary Business Actor:	Registered User (primary business actor) Guest User (primary business actor)	
Other Participating Actors:		
Other Interested Stakeholder:	System Owner	
Description:	This use case describes showing detail of a lesson.	
Precondition:	The user must be opened lessons page.	
Trigger:	This use case initiated when users click on any lesson in lessons page.	
Typical Course of Events:	Actor Action	System Response
	Step 1: Click one of the lessons. Step 2: Lesson detail is shown.	
Conclusion:	This use case concludes by showing lesson detail.	
Postcondition:	Lesson detail is shown to user.	

Table A.6: Show schedule use-case narrative

Use-Case Name:	Show Schedule	Use-Case Type Business Requirements:
Use-Case ID:	MGF-UC005.00	
Priority:	High	
Source:	Requirement-MGF-R5.00	
Primary Business Actor:	Registered User (primary business actor) Guest User (primary business actor)	
Other Participating Actors:		
Other Interested Stakeholder:	System Owner	
Description:	This use case describes showing schedule of the studio.	
Precondition:	The user must be installed the application.	
Trigger:	This use case initiated when users click schedule button on mainpage or on navigation menü.	
Typical Course of Events:	Actor Action	System Response
	Step 1: Click schedule button Step 2: Schedules are listed	Alt-Step 1: If error is occurred, no results message is shown.
Conclusion:	This use case concludes when response is gathered from server.	
Postcondition:	Schedules are listed to user or no results message is shown.	

Table A.7: Make reservation use-case narrative

Use-Case Name:	Make Reservation	Use-Case Type Business Requirements:
Use-Case ID:	MGF-UC006.00	
Priority:	High	
Source:	Requirement-MGF-R6.00	
Primary Business Actor:	Registered User (primary business actor)	
Other Participating Actors:		
Other Interested Stakeholder:	System Owner	
Description:	This use case describes making a reservation for a lesson.	
Precondition:	The registered user must have available package.	
Trigger:	This use case initiated when users clicks register button schedule page or reservation page.	
Typical Course of Events:	Actor Action	System Response
	Step 1: Click register button Step 2: Register is successful if package validation is successful.	Alt-Step 1: If error is occurred, error message is shown.
Conclusion:	This use case concludes when response is gathered from server.	
Postcondition:	Reservation is done or error message is shown to user after validation operation.	

Table A.8: Cancel reservation use-case narrative

Use-Case Name:	Cancel Reservation	Use-Case Type Business Requirements:
Use-Case ID:	MGF-UC007.00	
Priority:	High	
Source:	Requirement-MGF-R7.00	
Primary Business Actor:	Registered User (primary business actor)	
Other Participating Actors:		
Other Interested Stakeholder:	System Owner	
Description:	This use case describes canceling a reservation for a lesson.	
Precondition:	The registered user must have at least one current reservation.	
Trigger:	This use case initiated when users clicks cancel button on current reservation.	
Typical Course of Events:	Actor Action	System Response
	Step 1: Click cancel button Step 2: Cancellation is successful if validation is successful.	Alt-Step 1: If error is occurred, error message is shown.
Conclusion:	This use case concludes when response is gathered from server.	
Postcondition:	Reservation is canceled or error message is shown to user after validation operation.	

Table A.9: Show sport programmes use-case narrative

Use-Case Name:	Show Sport Programmes	Use-Case Type Business Requirements:
Use-Case ID:	MGF-UC008.00	
Priority:	High	
Source:	Requirement-MGF-R8.00	
Primary Business Actor:	Registered User (primary business actor)	
Other Participating Actors:		
Other Interested Stakeholder:	System Owner	
Description:	This use case describes showing sport programmes of a registered user.	
Precondition:	The registered user must have logged in to application.	
Trigger:	This use case initiated when users click sport programme button on home page or on navigation menu.	
Typical Course of Events:	Actor Action	System Response
	Step 1: Click sport programme button Step 2: Sport programmes are listed.	Alt-Step 1: If error is occurred, no results message is shown.
Conclusion:	This use case concludes when response is gathered from server.	
Postcondition:	Sports programmes are listed or no results message is shown.	

Table A.10: Show measurements use-case narrative

Use-Case Name:	Show Measurements	Use-Case Type Business Requirements:
Use-Case ID:	MGF-UC009.00	
Priority:	High	
Source:	Requirement-MGF-R9.00	
Primary Business Actor:	Registered User (primary business actor)	
Other Participating Actors:		
Other Interested Stakeholder:	System Owner	
Description:	This use case describes showing measurements of a registered user.	
Precondition:	The registered user must have logged in to application.	
Trigger:	This use case initiated when users click measurements button on navigation menu.	
Typical Course of Events:	Actor Action	System Response
	Step 1: Click measurements button Step 2: Measurements are listed.	Alt-Step 1: If error is occurred, no results message is shown.
Conclusion:	This use case concludes when response is gathered from server.	
Postcondition:	Measurements are listed or no results message is shown.	

Table A.11: Show user packages use-case narrative

Use-Case Name:	Show User Packages	Use-Case Type Business Requirements:
Use-Case ID:	MGF-UC010.00	
Priority:	High	
Source:	Requirement-MGF-R10.00	
Primary Business Actor:	Registered User (primary business actor)	
Other Participating Actors:		
Other Interested Stakeholder:	System Owner	
Description:	This use case describes showing packages of a registered user.	
Precondition:	The registered user must have logged in to application.	
Trigger:	This use case initiated when users click my packages button on navigation menu.	
Typical Course of Events:	Actor Action	System Response
	Step 1: Click my packages button Step 2: Packages are listed.	Alt-Step 1: If error is occurred, no results message is shown.
Conclusion:	This use case concludes when response is gathered from server.	
Postcondition:	Packages are listed or no results message is shown.	

Table A.12: Show current reservations use-case narrative

Use-Case Name:	Show Current Reservations	Use-Case Type Business Requirements:
Use-Case ID:	MGF-UC011.00	
Priority:	High	
Source:	Requirement-MGF-R11.00	
Primary Business Actor:	Registered User (primary business actor)	
Other Participating Actors:		
Other Interested Stakeholder:	System Owner	
Description:	This use case describes showing current reservations of a registered user.	
Precondition:	The registered user must have logged in to application.	
Trigger:	This use case initiated when users click current reservations on my reservations page.	
Typical Course of Events:	Actor Action	System Response
	Step 1: Click current reservations button Step 2: Reservations are listed.	Alt-Step 1: If error is occurred, no results message is shown.
Conclusion:	This use case concludes when response is gathered from server.	
Postcondition:	Reservations are listed or no results message is shown.	

Table A.13: Show former reservations use-case narrative

Use-Case Name:	Show Former Reservations	Use-Case Type Business Requirements:
Use-Case ID:	MGF-UC012.00	
Priority:	High	
Source:	Requirement-MGF-R12.00	
Primary Business Actor:	Registered User (primary business actor)	
Other Participating Actors:		
Other Interested Stakeholder:	System Owner	
Description:	This use case describes showing former reservations of a registered user.	
Precondition:	The registered user must have logged in to application.	
Trigger:	This use case initiated when users click former reservations on my reservations page.	
Typical Course of Events:	Actor Action	System Response
	Step 1: Click former reservations button Step 2: Reservations are listed.	Alt-Step 1: If error is occurred, no results message is shown.
Conclusion:	This use case concludes when response is gathered from server.	
Postcondition:	Reservation are listed or no results message is shown.	

Table A.14: Show announcements use-case narrative

Use-Case Name:	Show Announcements	Use-Case Type Business Requirements:
Use-Case ID:	MGF-UC013.00	
Priority:	High	
Source:	Requirement-MGF-R13.00	
Primary Business Actor:	Registered User (primary business actor)	
Other Participating Actors:		
Other Interested Stakeholder:	System Owner	
Description:	This use case describes showing announcements of studio.	
Precondition:	The registered user must have logged in to application.	
Trigger:	This use case initiated when users click announcements button on navigation menu.	
Typical Course of Events:	Actor Action	System Response
	Step 1: Click announcements button Step 2: Announcements are listed.	Alt-Step 1: If error is occurred, no results message is shown.
Conclusion:	This use case concludes when response is gathered from server.	
Postcondition:	Announcements are listed or no results message is shown.	

Table A.15: Show profile use-case narrative

Use-Case Name:	Show Profile	Use-Case Type Business Requirements:
Use-Case ID:	MGF-UC014.00	
Priority:	High	
Source:	Requirement-MGF-R14.00	
Primary Business Actor:	Registered User (primary business actor)	
Other Participating Actors:		
Other Interested Stakeholder:	System Owner	
Description:	This use case describes showing profile of studio.	
Precondition:	The registered user must have logged in to application.	
Trigger:	This use case initiated when users click profile button on navigation menu.	
Typical Course of Events:	Actor Action System Response	
	Step 1: Click profile button Step 2: Profile details are shown.	
Conclusion:	This use case concludes when profile details are shown.	
Postcondition:	Profile details are shown.	

Table A.16: Change language use-case narrative

Use-Case Name:	Change Language	Use-Case Type Business Requirements:
Use-Case ID:	MGF-UC015.00	
Priority:	High	
Source:	Requirement-MGF-R15.00	
Primary Business Actor:	Registered User (primary business actor)	
Other Participating Actors:		
Other Interested Stakeholder:	System Owner	
Description:	This use case describes changing language of application and profile.	
Precondition:	The registered user must have logged in to application.	
Trigger:	This use case initiated when users click on any flag on profile page	
Typical Course of Events:	Actor Action	System Response
	Step 1: Click on flag Step 2: Profile and app languages are changed.	Alt-Step 1: If error is occurred, no results message is shown and language is not changed.
Conclusion:	This use case concludes when response is gathered from server.	
Postcondition:	Language is changed if operation is successful.	

Table A.17: Show studio packages use-case narrative

Use-Case Name:	Show Studio Packages	Use-Case Type Business Requirements:
Use-Case ID:	MGF-UC016.00	
Priority:	High	
Source:	Requirement-MGF-R16.00	
Primary Business Actor:	Registered User (primary business actor)	
Other Participating Actors:		
Other Interested Stakeholder:	System Owner	
Description:	This use case describes showing available packages of studio to buy.	
Precondition:	The registered user must have logged in to application.	
Trigger:	This use case initiated when users click buy button on home page.	
Typical Course of Events:	Actor Action	System Response
	Step 1: Click buy button Step 2: Packages are listed.	Alt-Step 1: If error is occurred, no results message is shown.
Conclusion:	This use case concludes when response is gathered from server.	
Postcondition:	Packages are listed or no results message is shown.	

Table A.18: Show diet plan use-case narrative

Use-Case Name:	Show Diet Plan	Use-Case Type Business Requirements:
Use-Case ID:	MGF-UC017.00	
Priority:	High	
Source:	Requirement-MGF-R17.00	
Primary Business Actor:	Registered User (primary business actor)	
Other Participating Actors:		
Other Interested Stakeholder:	System Owner	
Description:	This use case describes showing diet plan of registered user.	
Precondition:	The registered user must have logged in to application.	
Trigger:	This use case initiated when users click community fit button on main page.	
Typical Course of Events:	Actor Action	System Response
	Step 1: Click community fit button Step 2: Meals are listed.	Alt-Step 1: If error is occurred, no results message is shown.
Conclusion:	This use case concludes when response is gathered from server.	
Postcondition:	Meals are listed or no results message is shown.	

Table A.19: Show studio informations use-case narrative

Use-Case Name:	Show Studio Informations	Use-Case Type Business Requirements:
Use-Case ID:	MGF-UC018.00	
Priority:	High	
Source:	Requirement-MGF-R18.00	
Primary Business Actor:	Registered User (primary business actor) Guest User (primary business actor)	
Other Participating Actors:		
Other Interested Stakeholder:	System Owner	
Description:	This use case describes showing studio informations.	
Precondition:	The application must be installed.	
Trigger:	This use case initiated when users clicks about us button on home page.	
Typical Course of Events:	Actor Action	System Response
	Step 1: Click about us button Step 2: Studio informations are listed.	Alt-Step 1: If error is occurred, no results message is shown.
Conclusion:	This use case concludes when response is gathered from server.	
Postcondition:	Studio informations are listed or no results message is shown.	

Table A.20: Buy package use-case narrative

Use-Case Name:	Buy Package	Use-Case Type Business Requirements:
Use-Case ID:	MGF-UC019.00	
Priority:	High	
Source:	Requirement-MGF-R19.00	
Primary Business Actor:	Registered User (primary business actor)	
Other Participating Actors:		
Other Interested Stakeholder:	System Owner	
Description:	This use case describes buying a package.	
Precondition:	The registered user must have logged in to application and must have selected a package to buy.	
Trigger:	This use case initiated when users click buy button on any package.	
Typical Course of Events:	Actor Action	System Response
	Step 1: Enter card information and buy button Step 2: Payment is done.	Alt-Step 1: If error is occurred, error message is shown.
Conclusion:	This use case concludes when response is gathered from server.	
Postcondition:	Package is bought by user or error message is shown to user.	

Table A.21: Update profile image use-case narrative

Use-Case Name:	Update Profile Image	Use-Case Type Business Requirements:
Use-Case ID:	MGF-UC020.00	
Priority:	High	
Source:	Requirement-MGF-R20.00	
Primary Business Actor:	Registered User (primary business actor)	
Other Participating Actors:		
Other Interested Stakeholder:	System Owner	
Description:	This use case describes updating profile image of registered user.	
Precondition:	The registered user must have logged in to application.	
Trigger:	This use case initiated when users select different image for profile image.	
Typical Course of Events:	Actor Action	System Response
	Step 1: Select image from gallery. Step 2: Profile image is updated.	Alt-Step 1: If error is occurred, error message is shown.
Conclusion:	This use case concludes when response is gathered from server.	
Postcondition:	Profile image is updated or error message is shown to user.	

Table A.22: Add meal use-case narrative

Use-Case Name:	Add Meal	Use-Case Type Business Requirements:
Use-Case ID:	MGF-UC021.00	
Priority:	High	
Source:	Requirement-MGF-R21.00	
Primary Business Actor:	Registered User (primary business actor)	
Other Participating Actors:		
Other Interested Stakeholder:	System Owner	
Description:	This use case describes adding a meal to daily diet plan.	
Precondition:	The registered user must have logged in to application.	
Trigger:	This use case initiated when users select meal to add diet plan.	
Typical Course of Events:	Actor Action	System Response
	Step 1: Select meal from list. Step 2: Meal is added to diet plan.	Alt-Step 1: If error is occurred, error message is shown.
Conclusion:	This use case concludes when response is gathered from server.	
Postcondition:	Meal is added to diet plan or error message is shown to user.	

Table A.23: List current conversations use-case narrative

Use-Case Name:	List Current Conversations	Use-Case Type Business Requirements:
Use-Case ID:	MGF-UC022.00	
Priority:	High	
Source:	Requirement-MGF-R22.00	
Primary Business Actor:	Registered User (primary business actor)	
Other Participating Actors:		
Other Interested Stakeholder:	System Owner	
Description:	This use case describes listing current conversations of a registered user.	
Precondition:	The registered user must have logged in to application.	
Trigger:	This use case initiated when users click my messages button on home page.	
Typical Course of Events:	Actor Action	System Response
	Step 1: Click my messages button. Step 2: Current conversations are listed.	Alt-Step 1: If error is occurred, no results message is shown.
Conclusion:	This use case concludes when response is gathered from server.	
Postcondition:	Current conversations are listed.	

Table A.24: List available users use-case narrative

Use-Case Name:	List Available Users	Use-Case Type Business Requirements:
Use-Case ID:	MGF-UC023.00	
Priority:	High	
Source:	Requirement-MGF-R23.00	
Primary Business Actor:	Registered User (primary business actor)	
Other Participating Actors:		
Other Interested Stakeholder:	System Owner	
Description:	This use case describes listing available users message can be sent.	
Precondition:	The registered user must have logged in to application.	
Trigger:	This use case initiated when users clicks send message button on my messages page.	
Typical Course of Events:	Actor Action	System Response
	Step 1: Click send message button. Step 2: Available users are listed.	Alt-Step 1: If error is occurred, no results message is shown.
Conclusion:	This use case concludes when response is gathered from server.	
Postcondition:	Available users are listed.	

Table A.25: List older messages use-case narrative

Use-Case Name:	List Older Messages	Use-Case Type Business Requirements:
Use-Case ID:	MGF-UC024.00	
Priority:	High	
Source:	Requirement-MGF-R24.00	
Primary Business Actor:	Registered User (primary business actor)	
Other Participating Actors:		
Other Interested Stakeholder:	System Owner	
Description:	This use case describes listing older messages between registered user and another user.	
Precondition:	The registered user must have logged in to application.	
Trigger:	This use case initiated when users clicks on any user that current user has messaged before.	
Typical Course of Events:	Actor Action	System Response
	Step 1: Click on any user. Step 2: Older messages are listed.	Alt-Step 1: If error is occurred, error message is shown.
Conclusion:	This use case concludes when response is gathered from server.	
Postcondition:	Older messages are listed.	

Table A.26: Send message use-case narrative

Use-Case Name:	Send Message	Use-Case Type Business Requirements:
Use-Case ID:	MGF-UC025.00	
Priority:	High	
Source:	Requirement-MGF-R25.00	
Primary Business Actor:	Registered User (primary business actor)	
Other Participating Actors:		
Other Interested Stakeholder:	System Owner	
Description:	This use case describes sending message from registered user to another user.	
Precondition:	The registered user must have logged in to application and there should be available user to send message.	
Trigger:	This use case initiated when users click send message button during conversation.	
Typical Course of Events:	Actor Action	System Response
	Step 1: Click on send message button. Step 2: Message is sent.	Alt-Step 1: If error is occurred, error message is shown.
Conclusion:	This use case concludes when response is gathered from server.	
Postcondition:	Message is sent.	

Table A.27: Send message to studio use-case narrative

Use-Case Name:	Send Message to Studio	Use-Case Type Business Requirements:
Use-Case ID:	MGF-UC026.00	
Priority:	High	
Source:	Requirement-MGF-R26.00	
Primary Business Actor:	Registered User (primary business actor) Guest User (primary business actor)	
Other Participating Actors:		
Other Interested Stakeholder:	System Owner	
Description:	This use case describes sending message to studio.	
Precondition:	The application must be installed.	
Trigger:	This use case initiated when users clicks contact us button in about us page, and submit button after message data is entered.	
Typical Course of Events:	Actor Action	System Response
	Step 1: Click submit button. Step 2: Message is sent.	Alt-Step 1: If error is occurred, error message is shown.
Conclusion:	This use case concludes when response is gathered from server.	
Postcondition:	Message is sent.	

Appendix A.2 ACTIVITY DIAGRAMS

Figure A. 1: Login use-case activity diagram

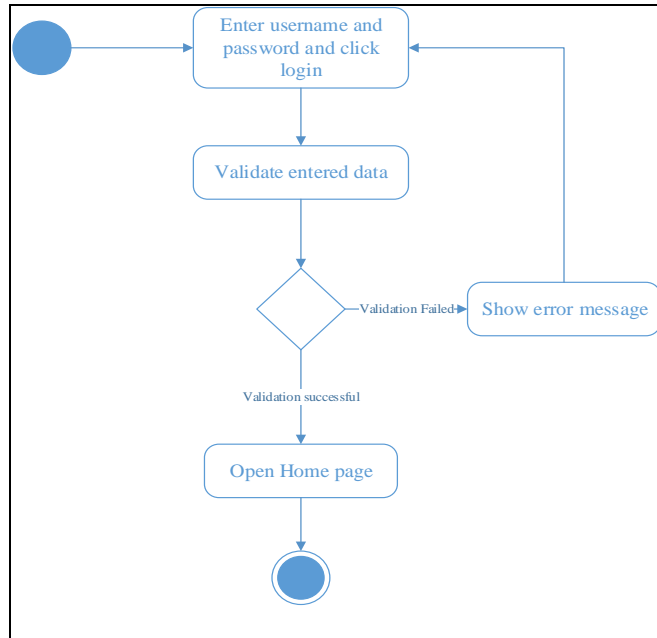


Figure A. 2: Register use-case activity diagram

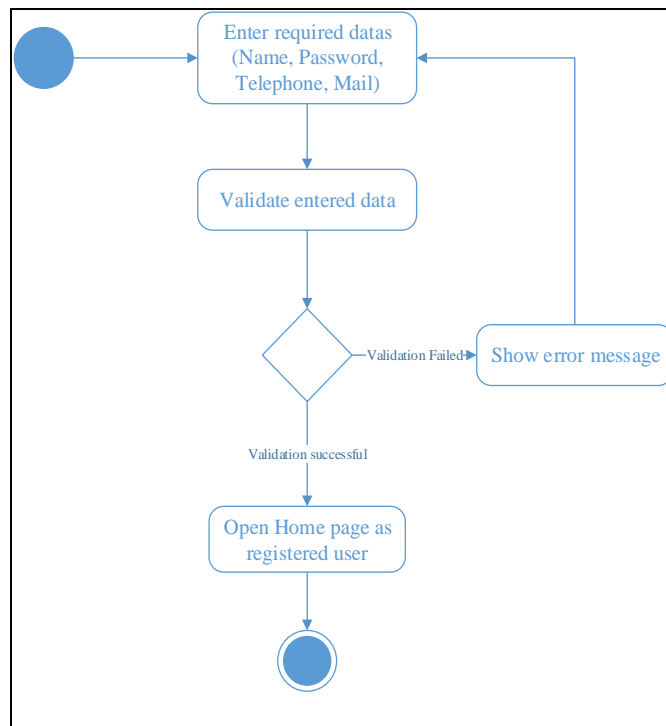


Figure A. 3: List lessons use-case activity diagram

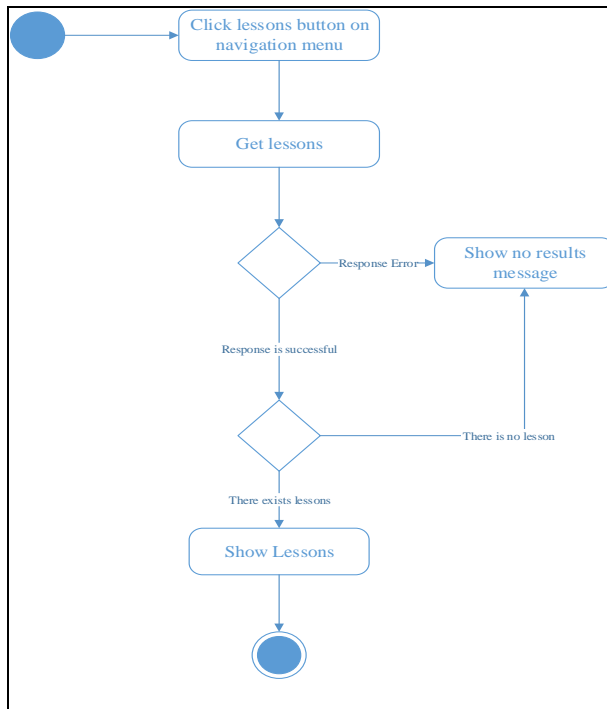


Figure A. 4: Show schedules use-case activity diagram

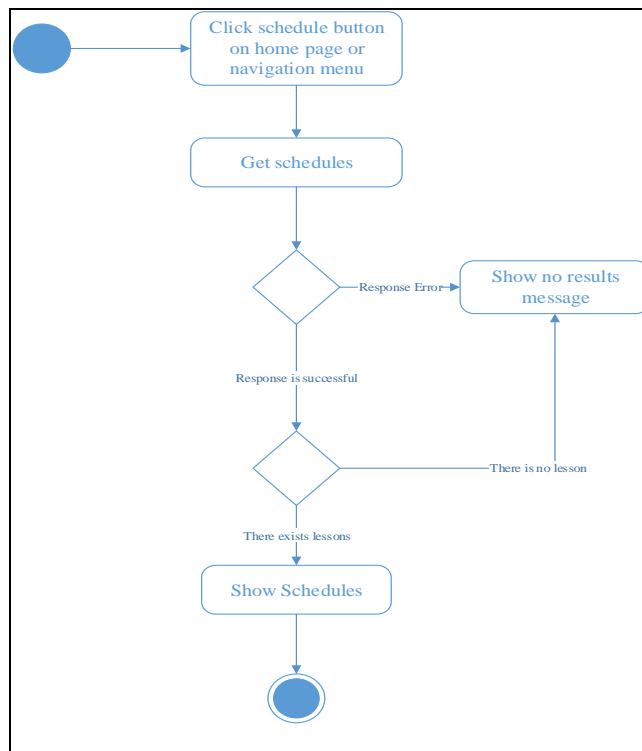


Figure A. 5: Make reservation use-case activity diagram

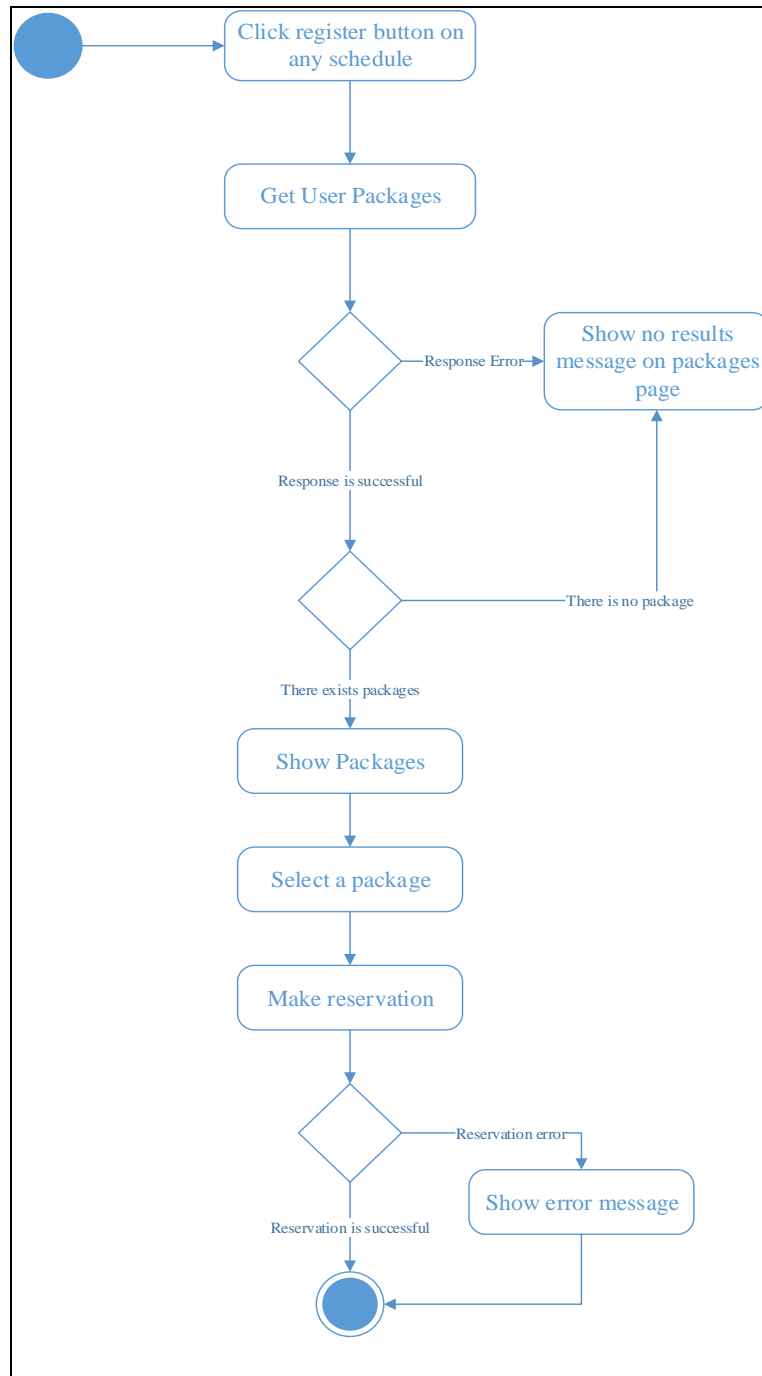


Figure A. 6: Cancel reservation use-case activity diagram

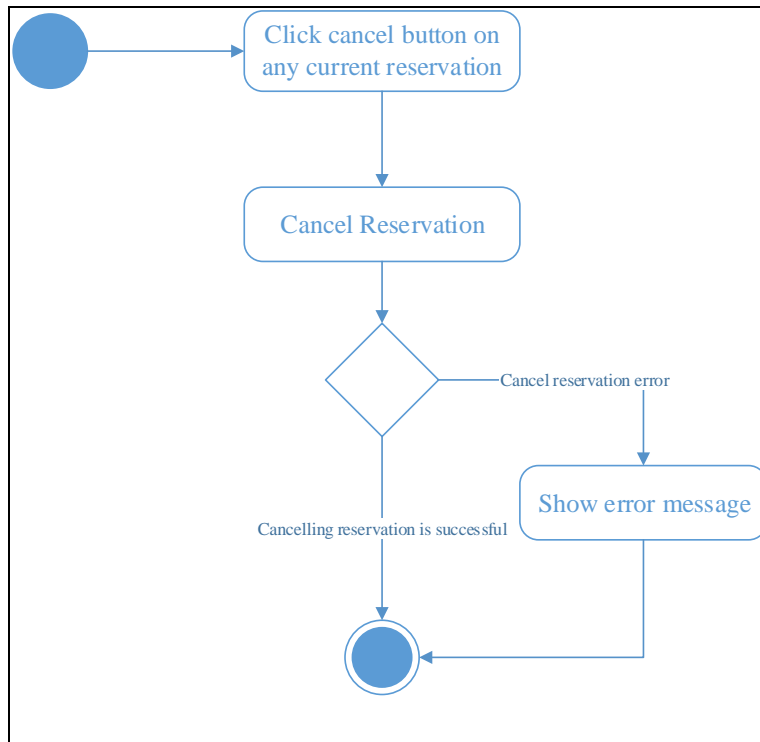


Figure A. 7: Show sport programmes use-case activity diagram

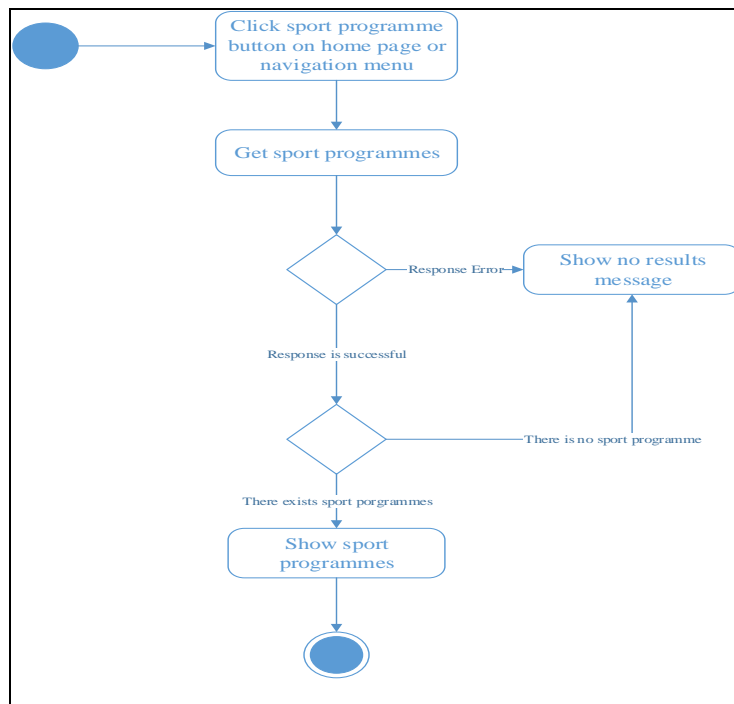


Figure A. 8: Show measurements use-case activity diagram

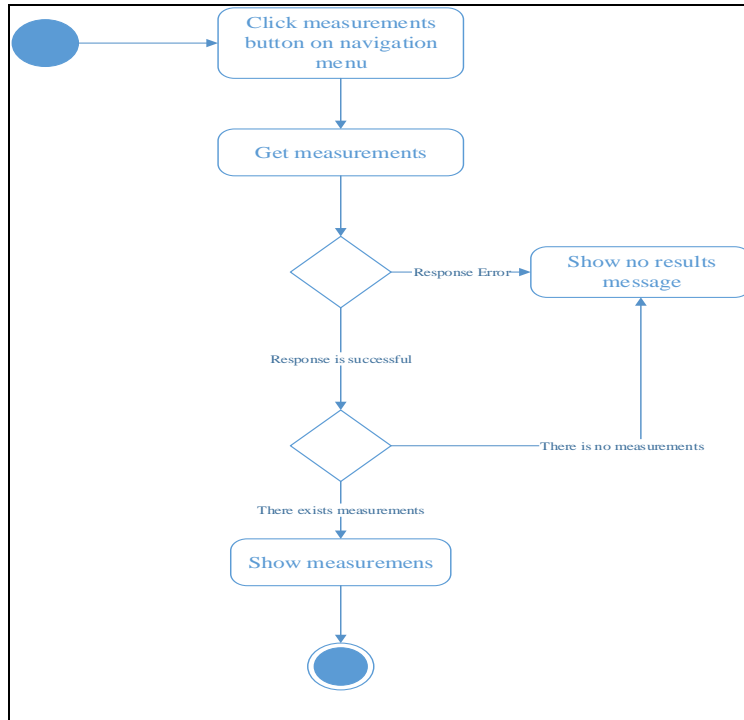


Figure A. 9: Show user packages use-case activity diagram

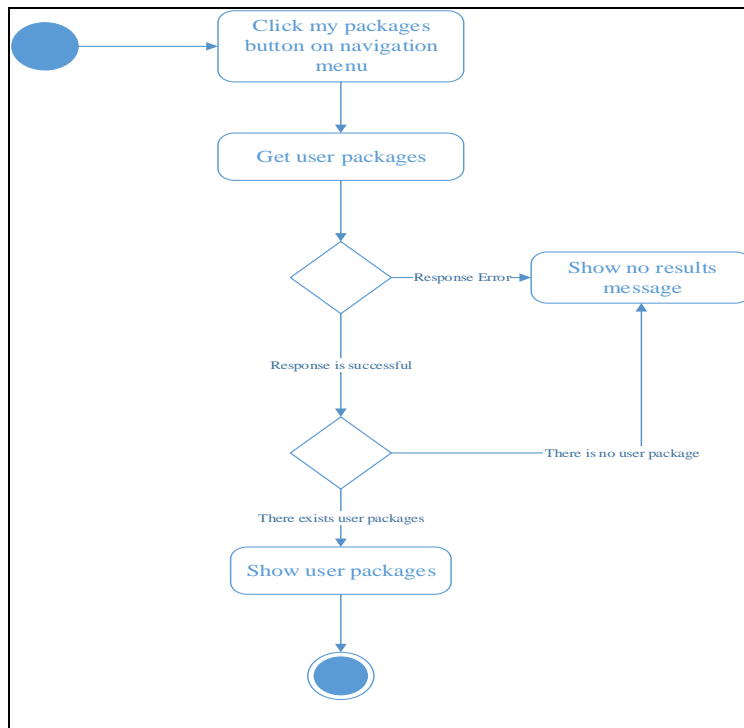


Figure A. 10: Show current reservations use-case activity diagram

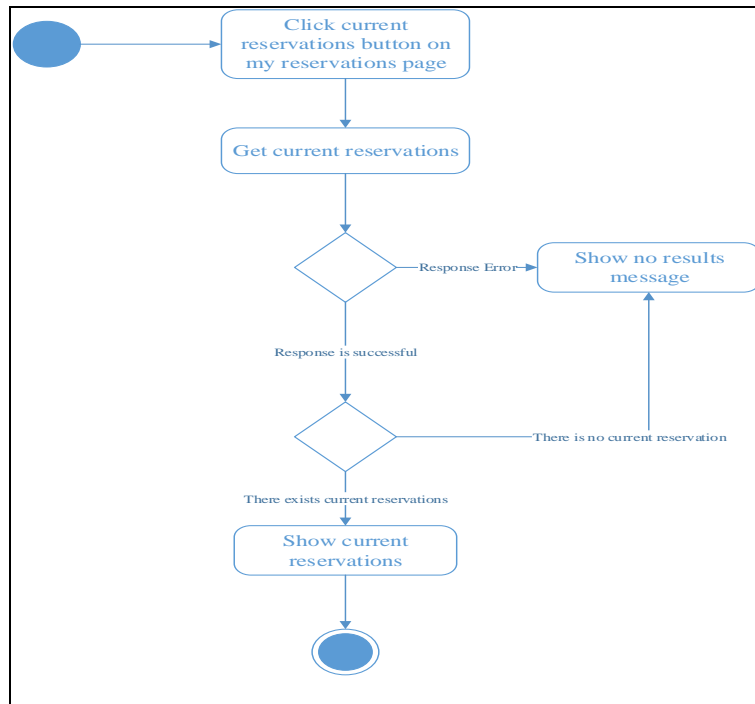


Figure A. 11: Show former reservations use-case activity diagram

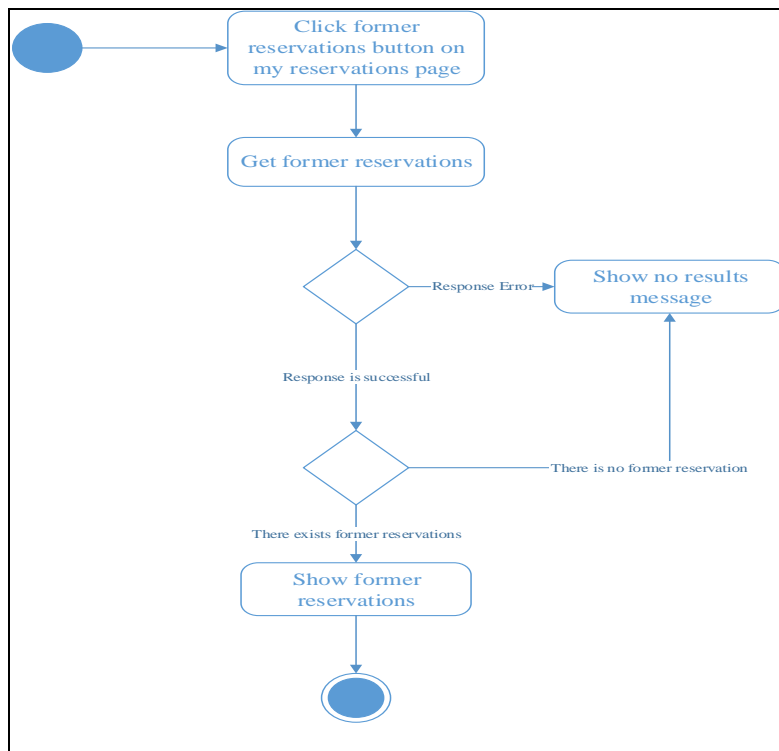


Figure A. 12: Show announcements use-case activity diagram

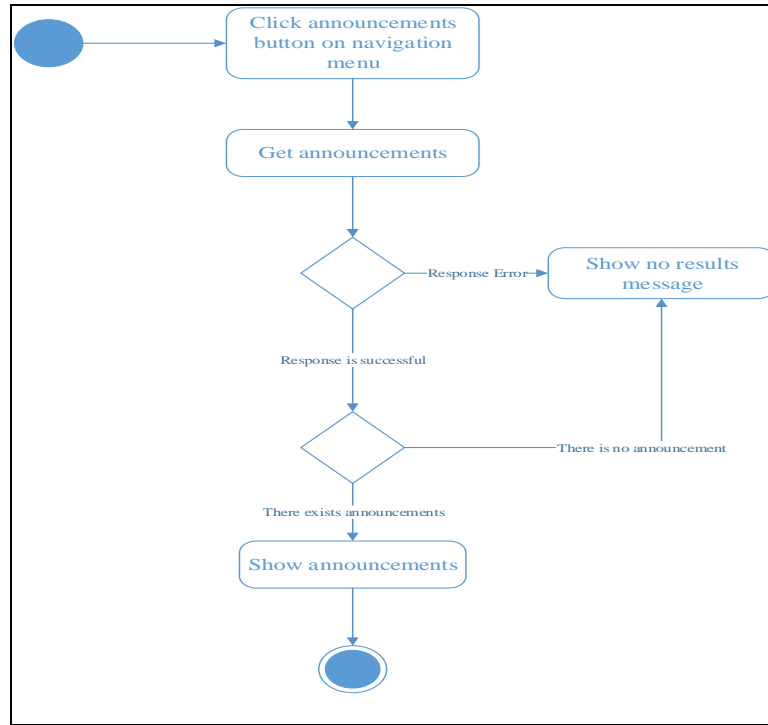


Figure A. 13: Show studio packages use-case activity diagram

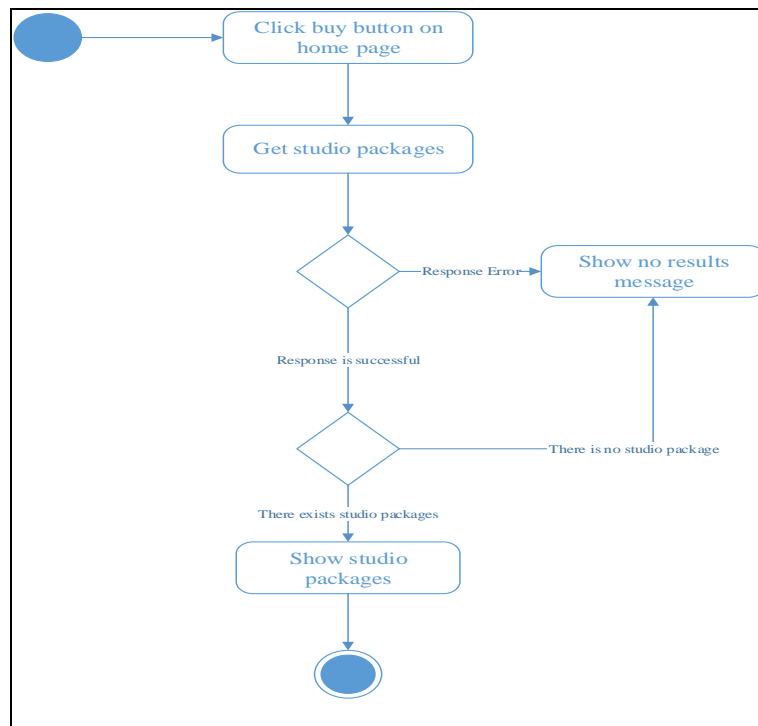


Figure A. 14: Show diet plan use-case activity diagram

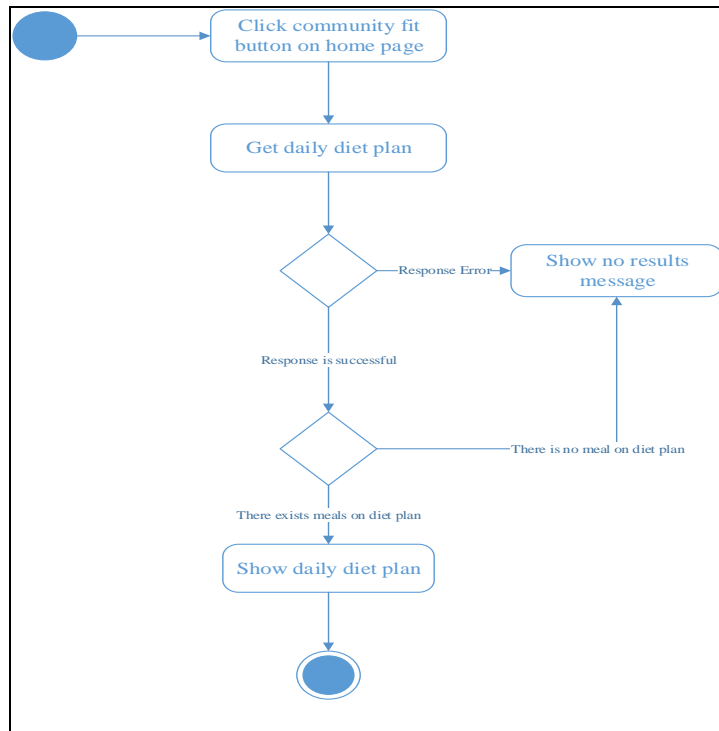


Figure A. 15: Buy package use-case activity diagram

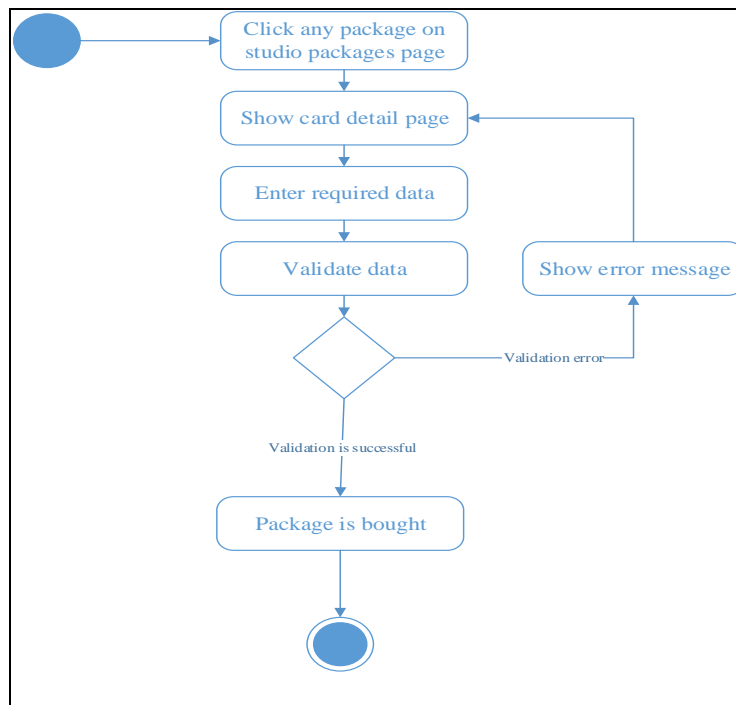


Figure A. 16: Add meal use-case activity diagram

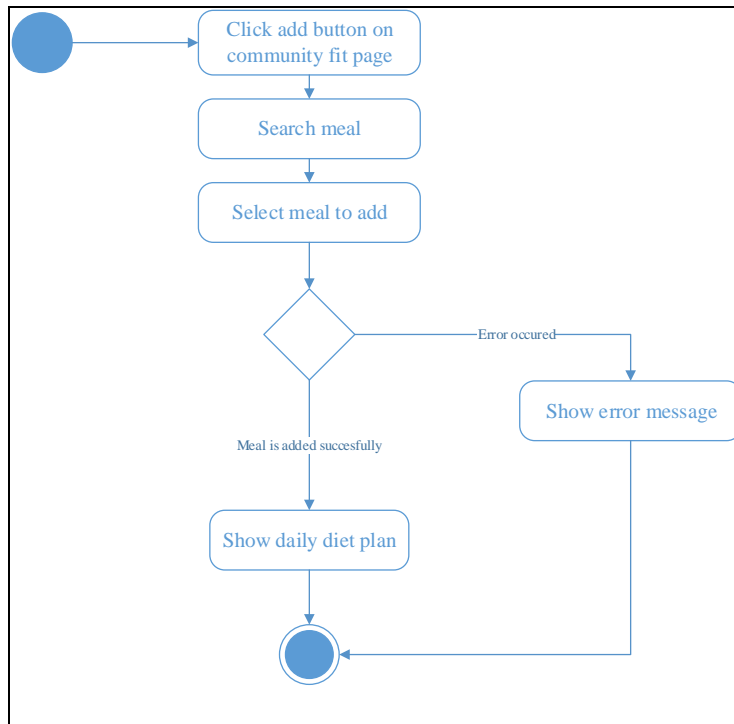


Figure A. 17: List current conversations use-case activity diagram

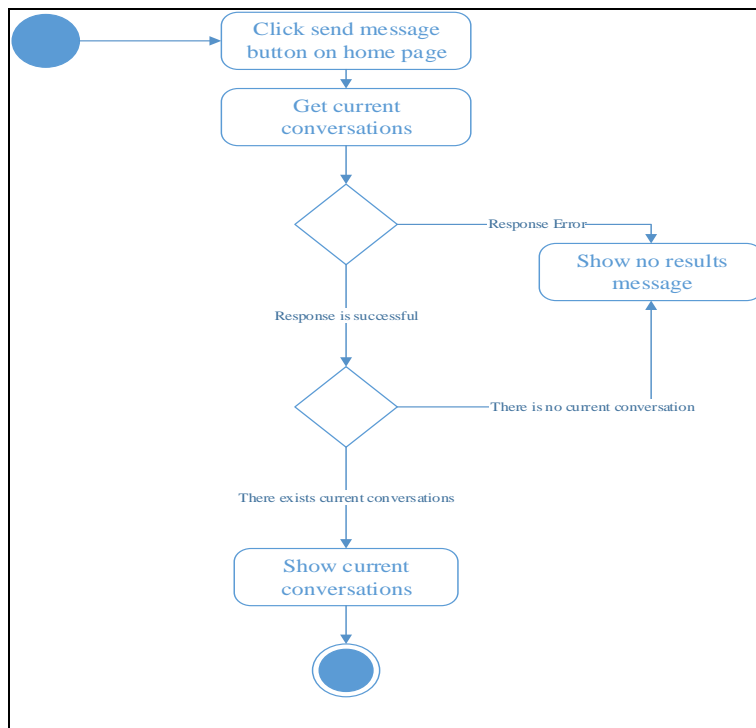


Figure A. 18: List available users use-case activity diagram

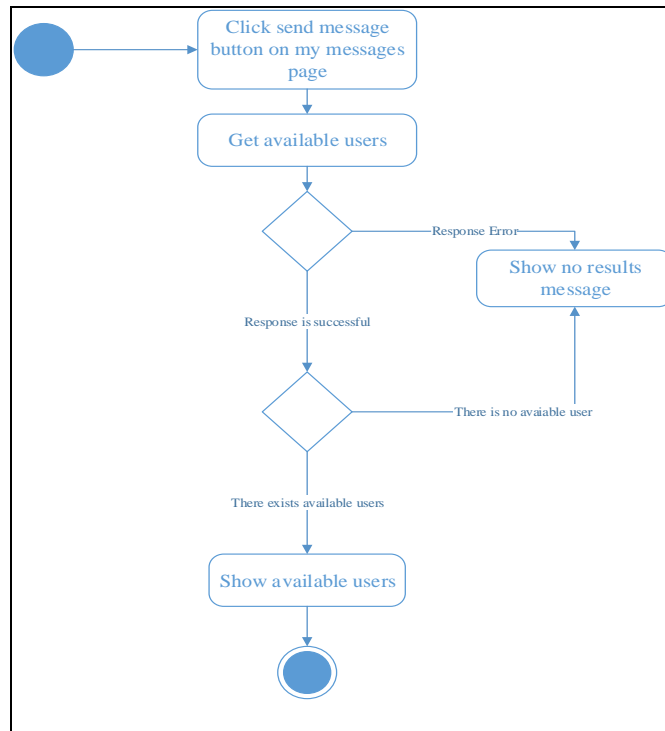


Figure A. 19: List older messages use-case activity diagram

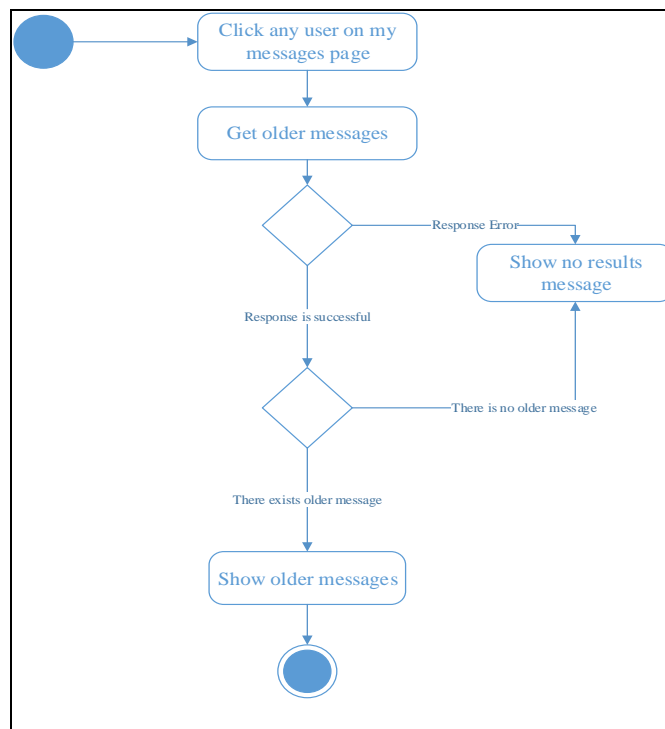


Figure A. 20: Send message use-case activity diagram

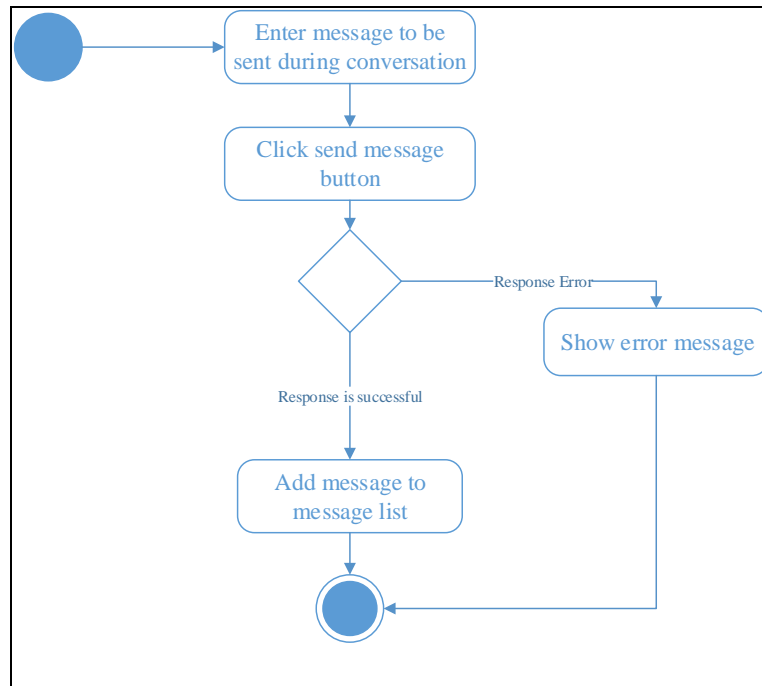
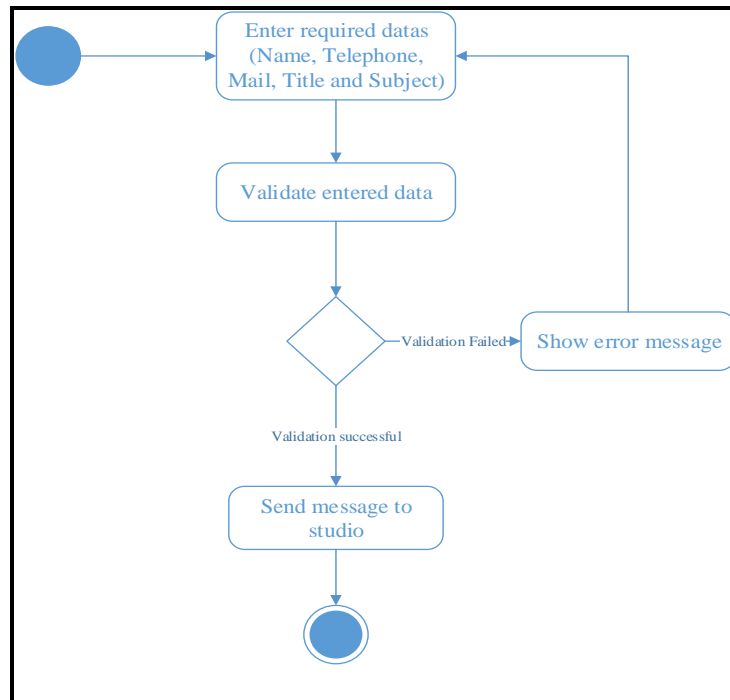


Figure A. 21: Send message to studio use-case activity diagram



Appendix A.3 SEQUENCE DIAGRAMS

Figure A. 22: Login use-case sequence diagram

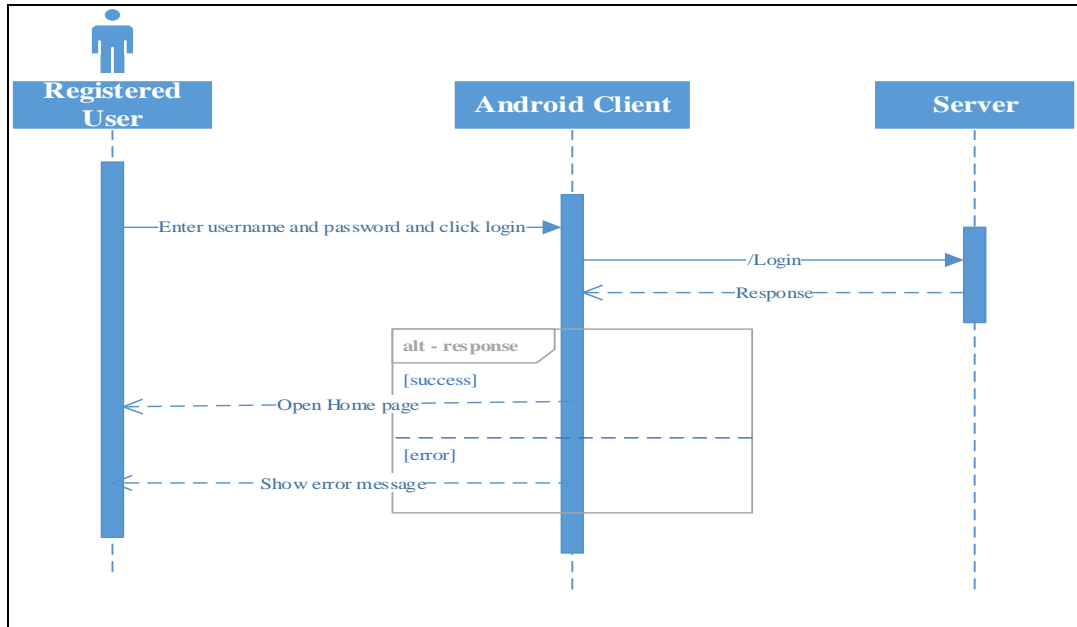


Figure A. 23: Register use-case sequence diagram

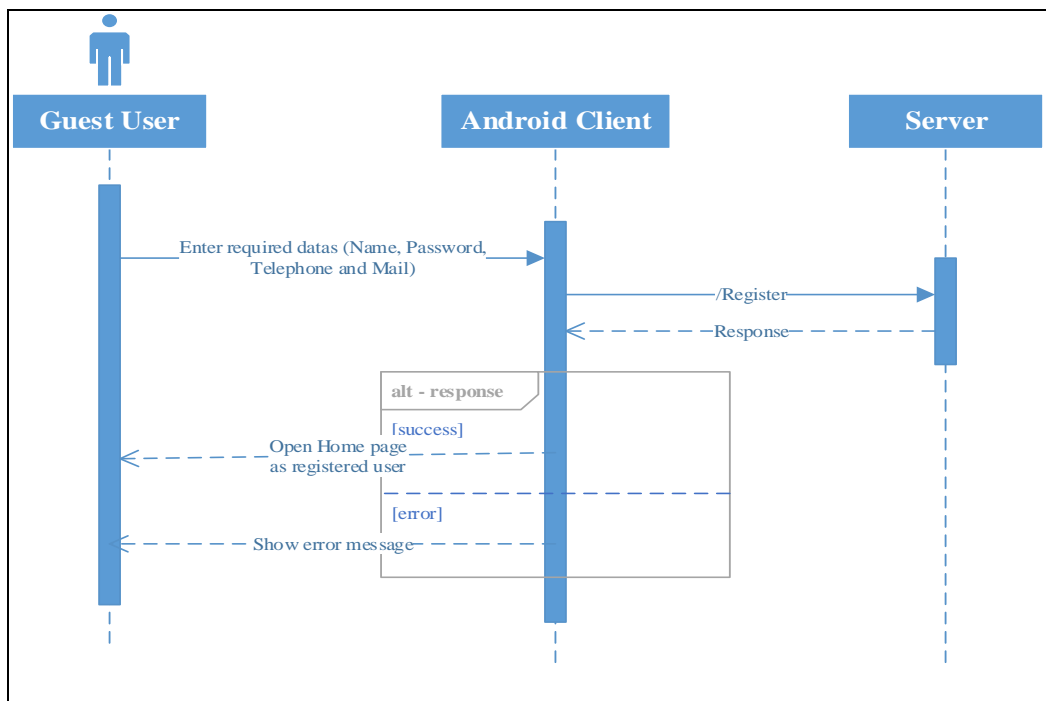


Figure A. 24: List lessons use-case sequence diagram

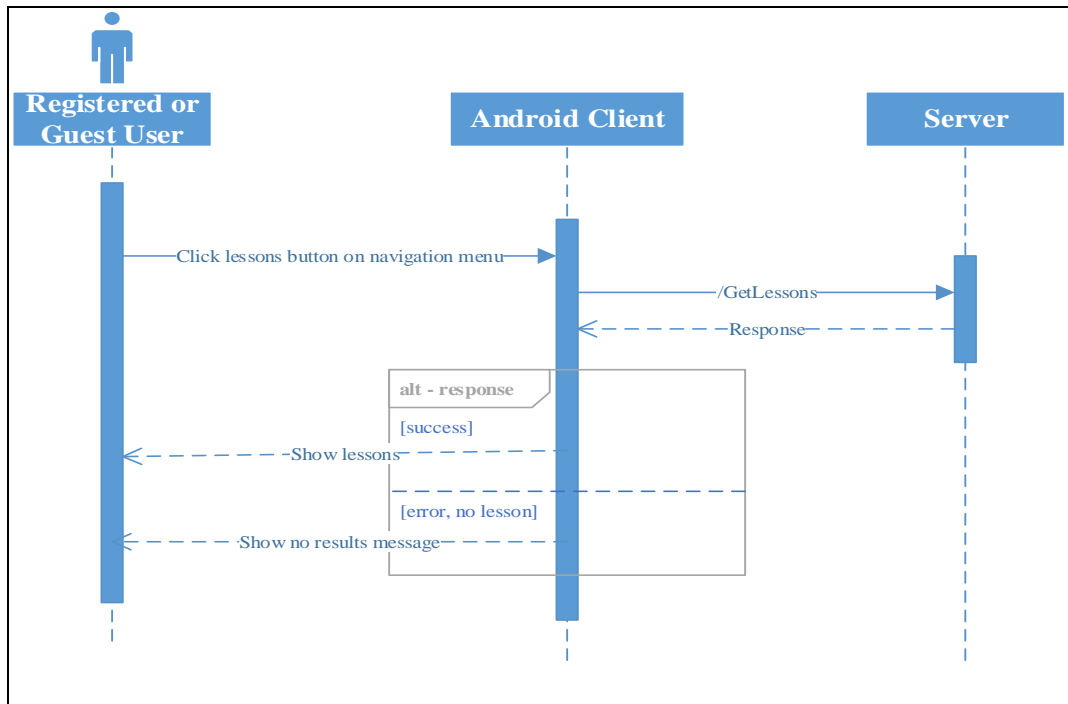


Figure A. 25: Show schedule use-case sequence diagram

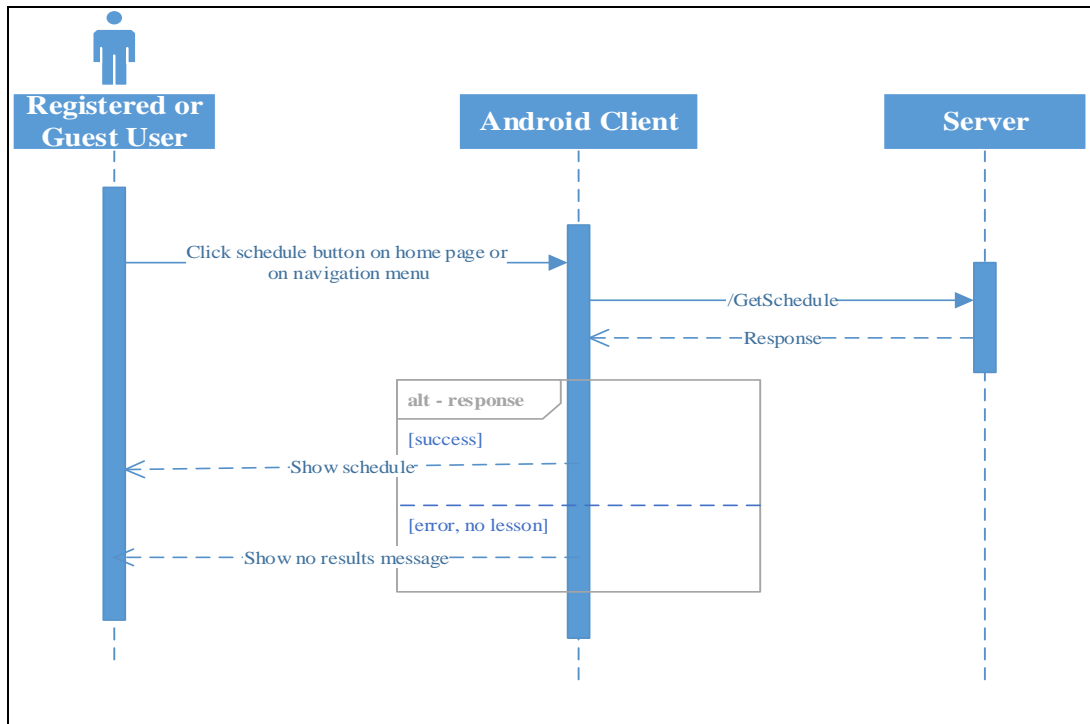


Figure A. 26: Make reservation use-case sequence diagram

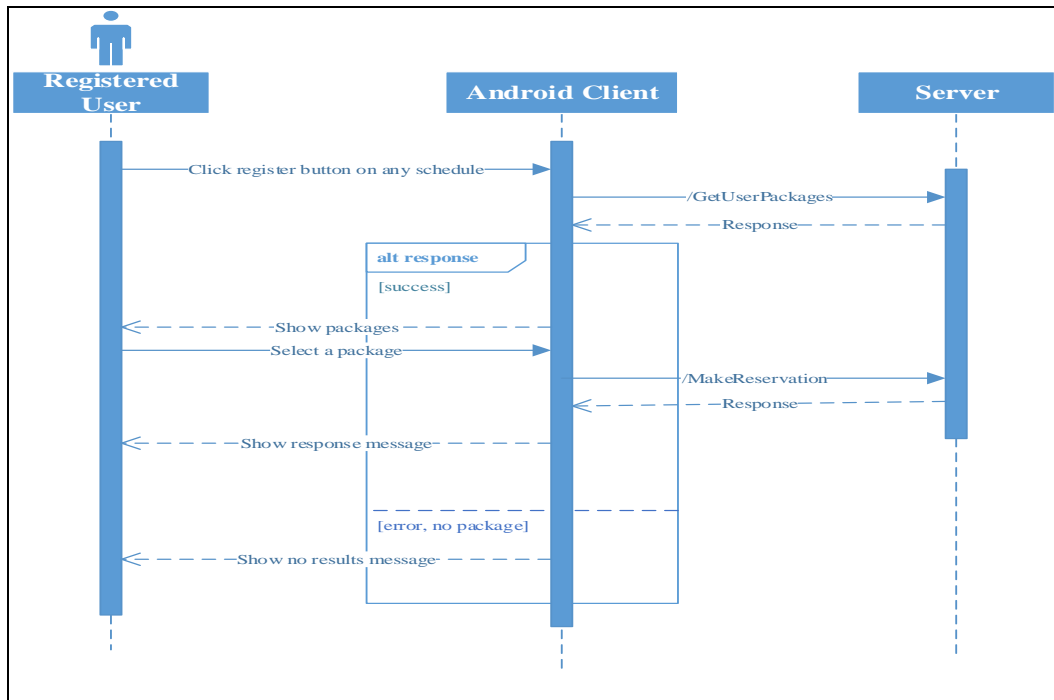


Figure A. 27: Cancel reservation use-case sequence diagram

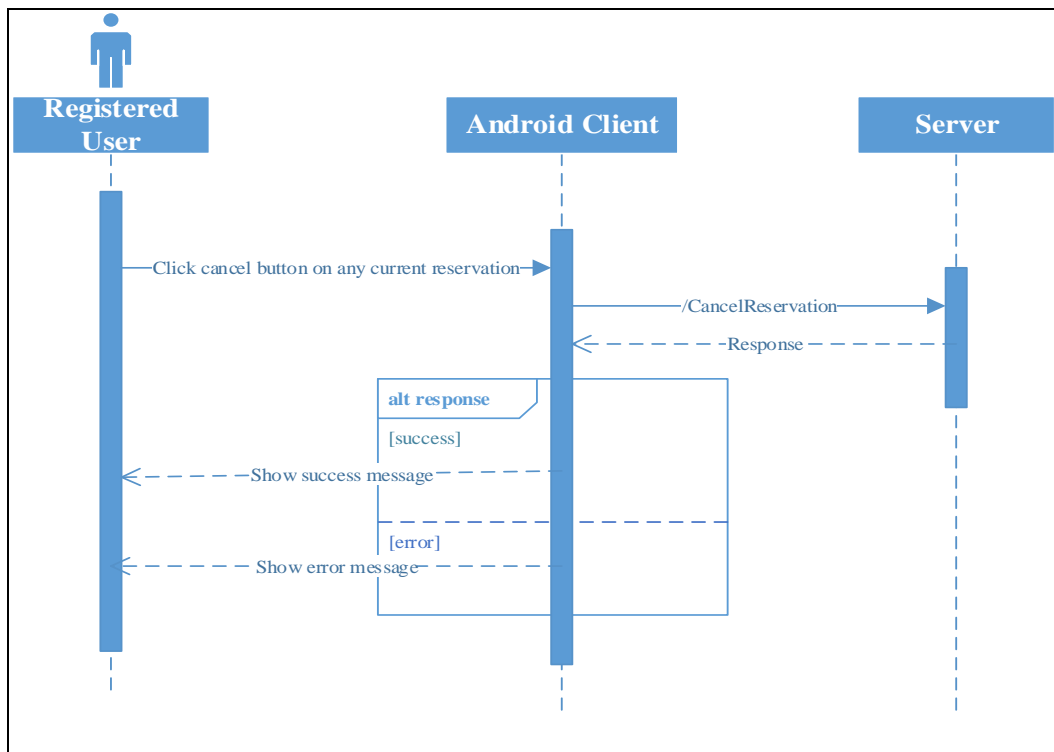


Figure A. 28: Show sport programmes use-case sequence diagram

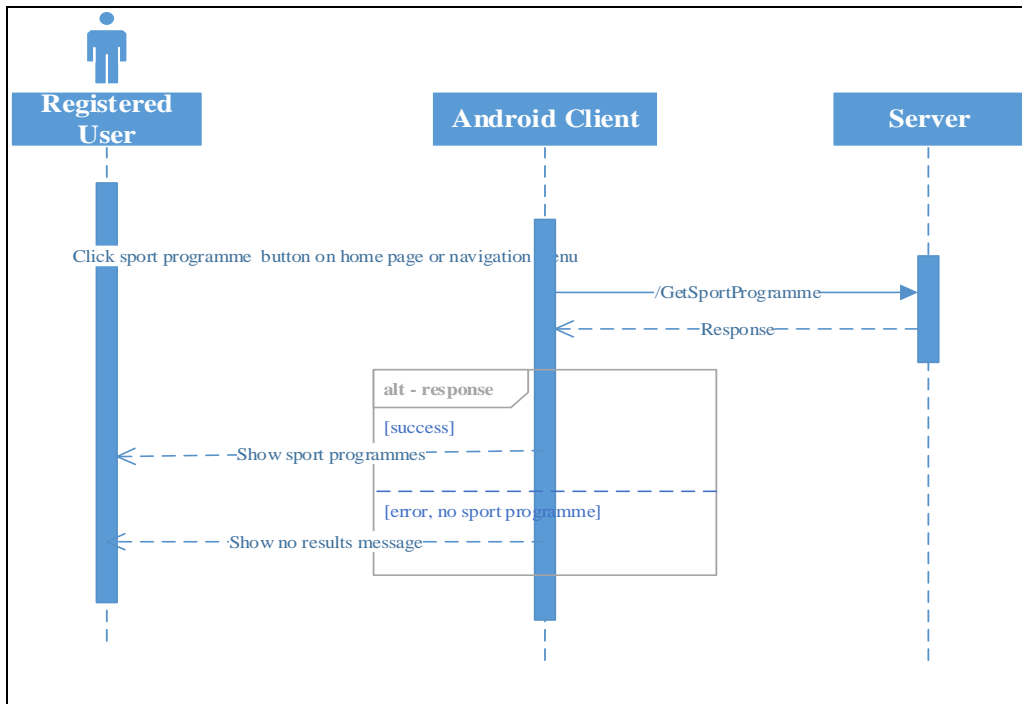


Figure A. 29: Show measurements use-case sequence diagram

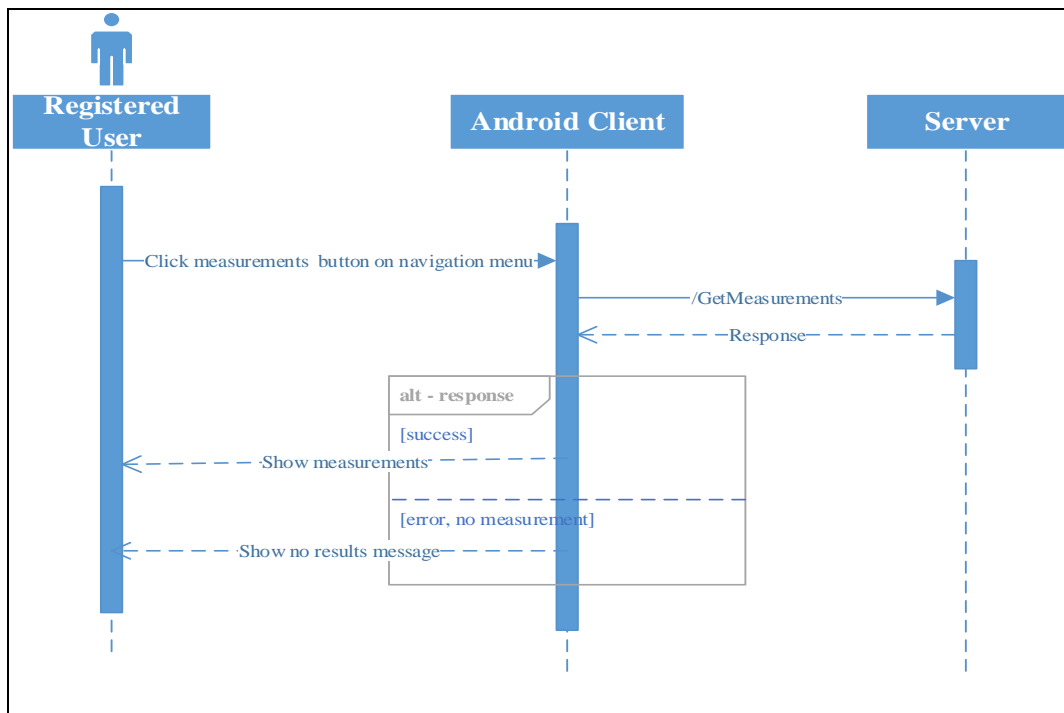


Figure A. 30: Show user packages use-case sequence diagram

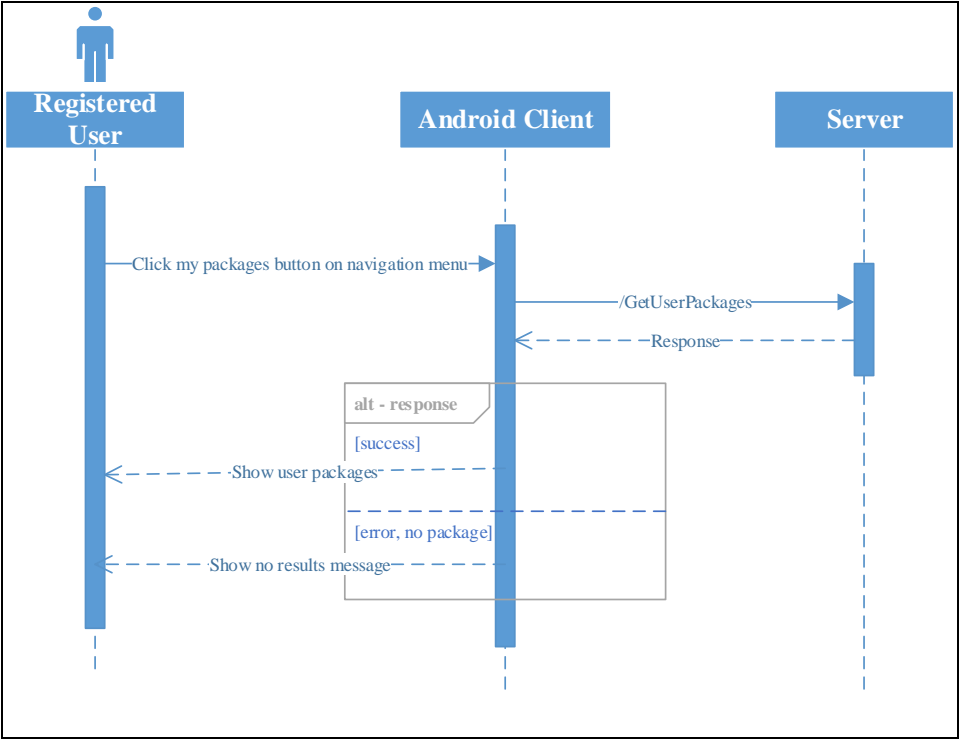


Figure A. 31: Show current reservations use-case sequence diagram

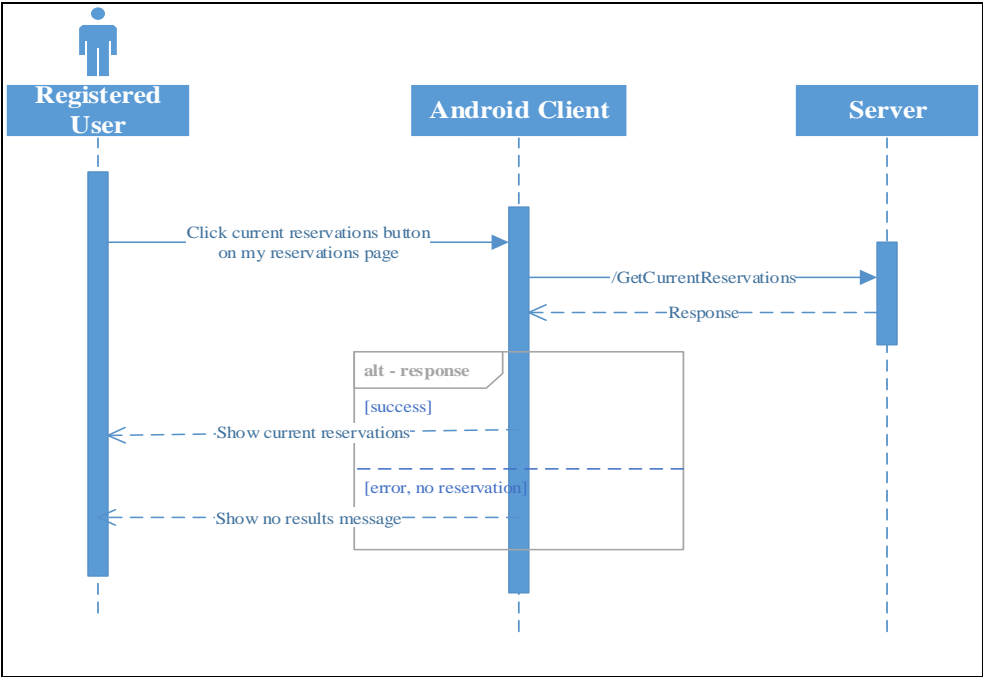


Figure A. 32: Show former reservations use-case sequence diagram

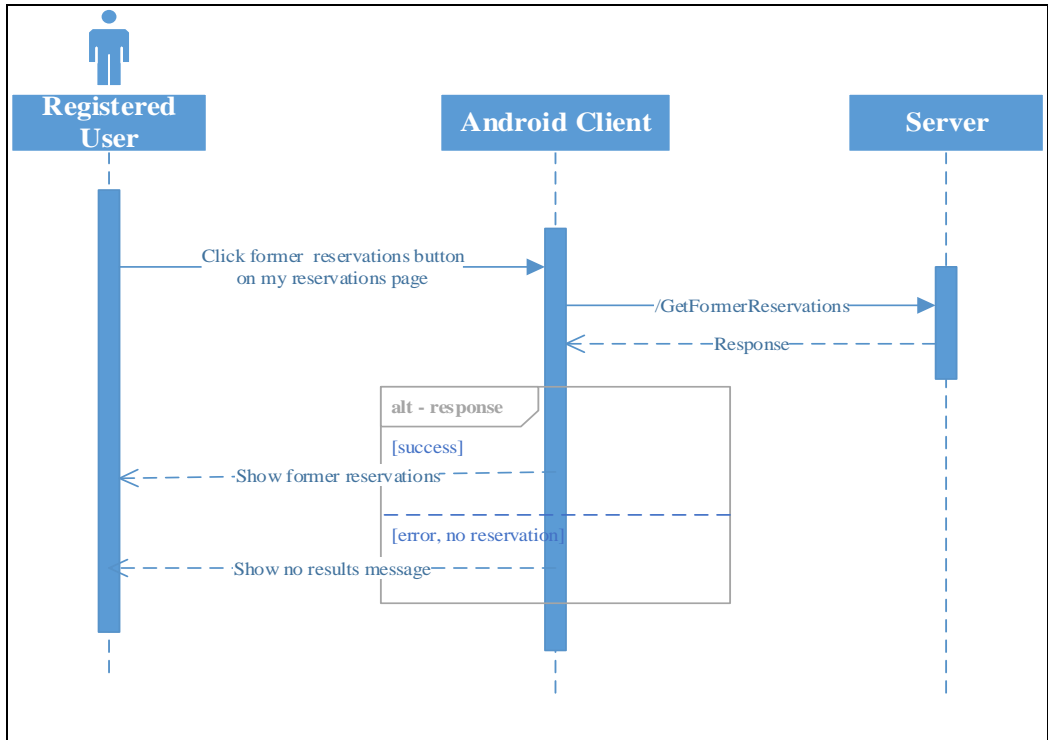


Figure A. 33: Show announcements use-case sequence diagram

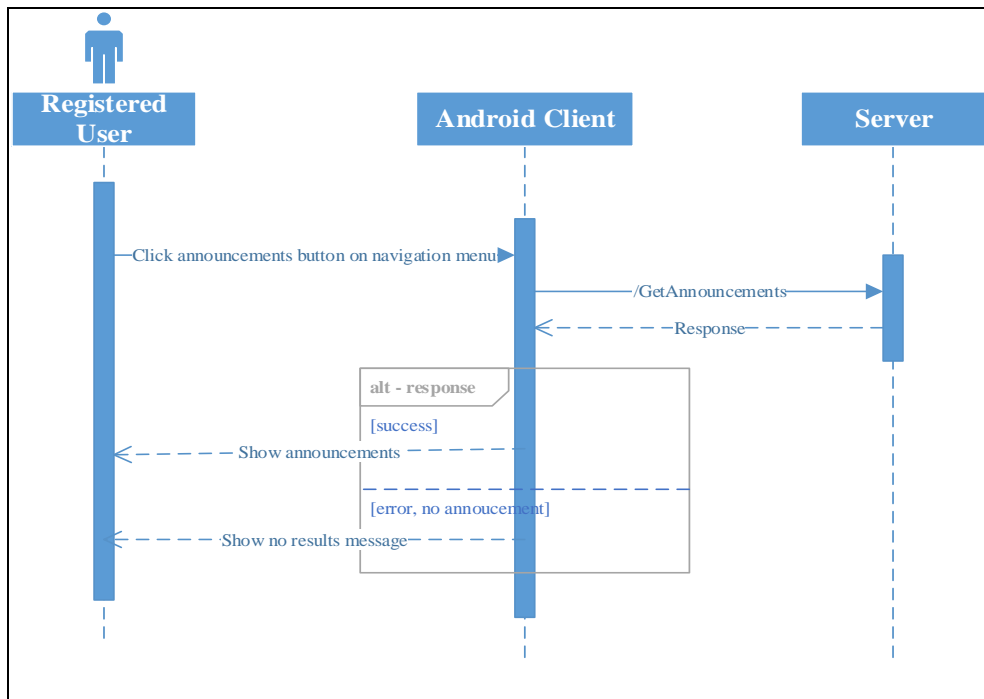


Figure A. 34: Buy package use-case sequence diagram

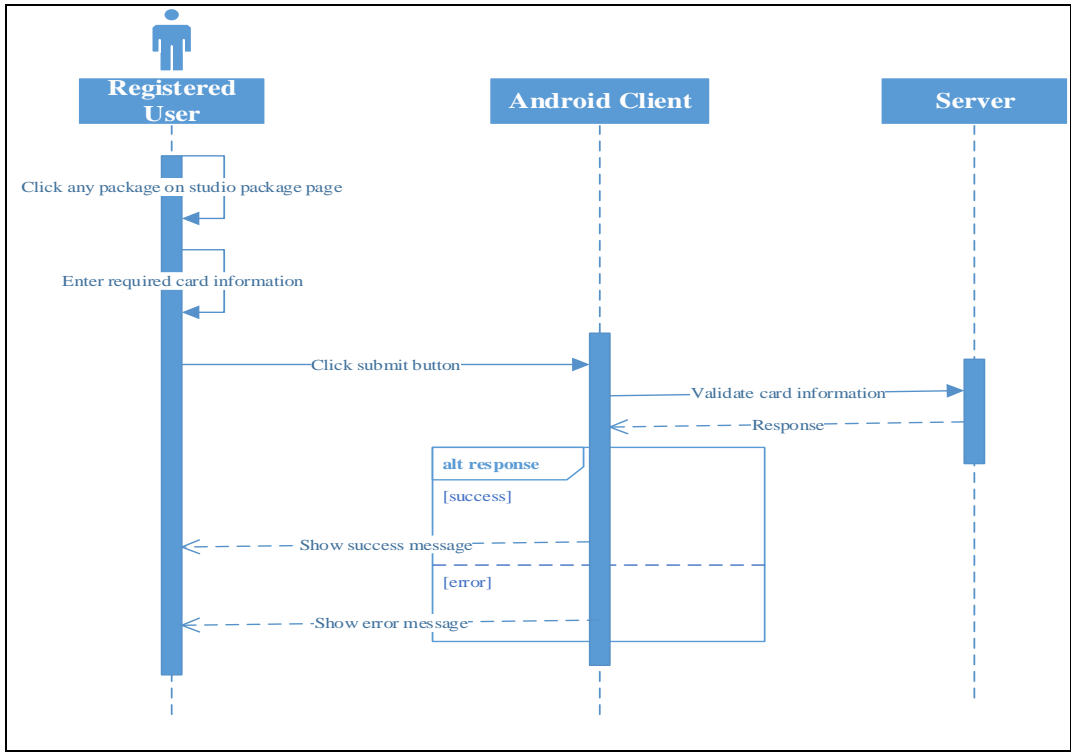


Figure A. 35: Add meal use-case sequence diagram

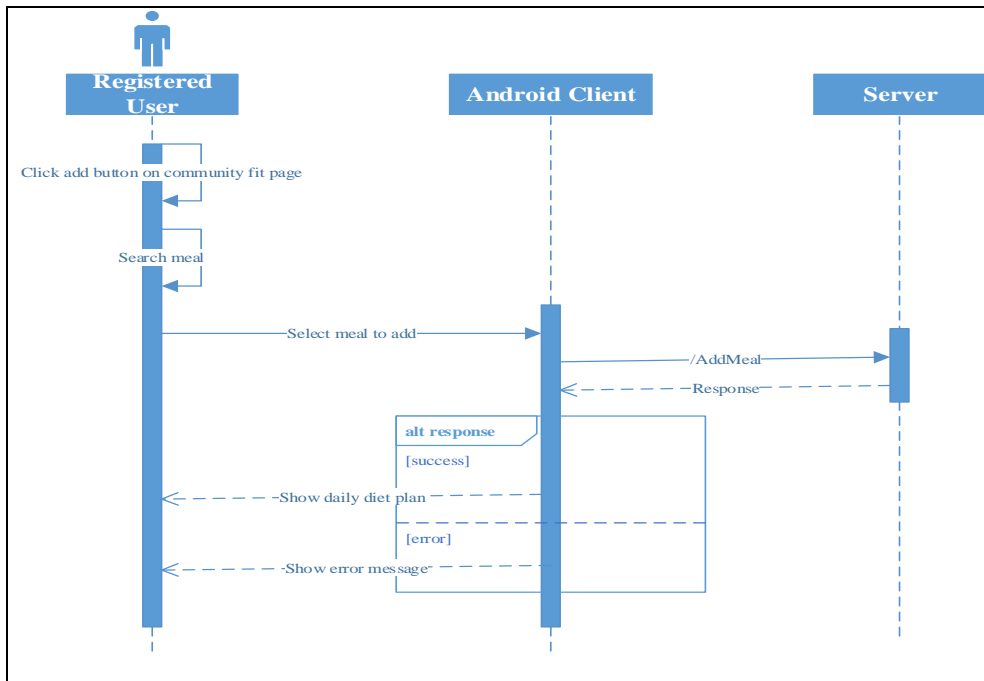


Figure A. 36: List current conversations use-case sequence diagram

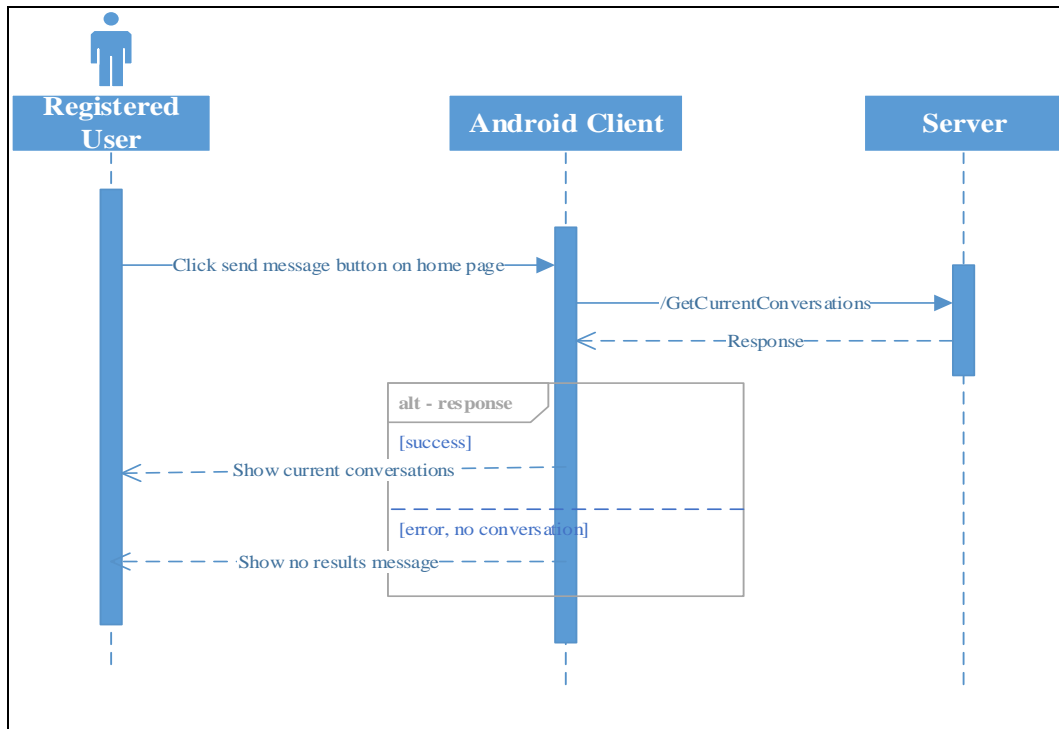


Figure A. 37: List available users use-case sequence diagram

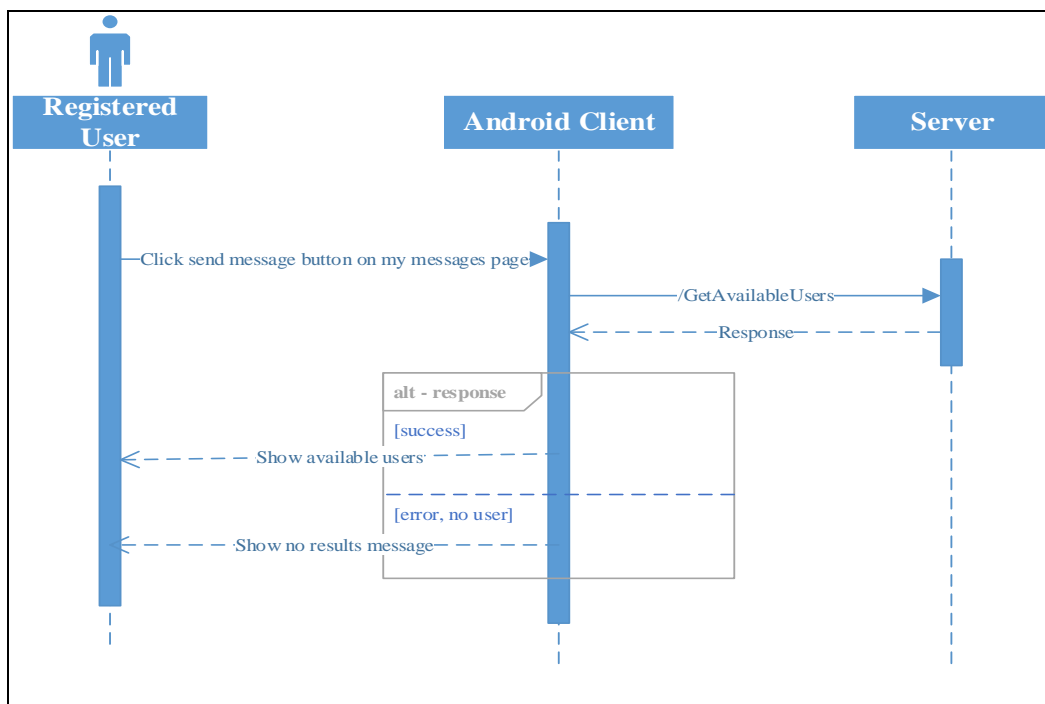


Figure A. 38: List older messages use-case sequence diagram

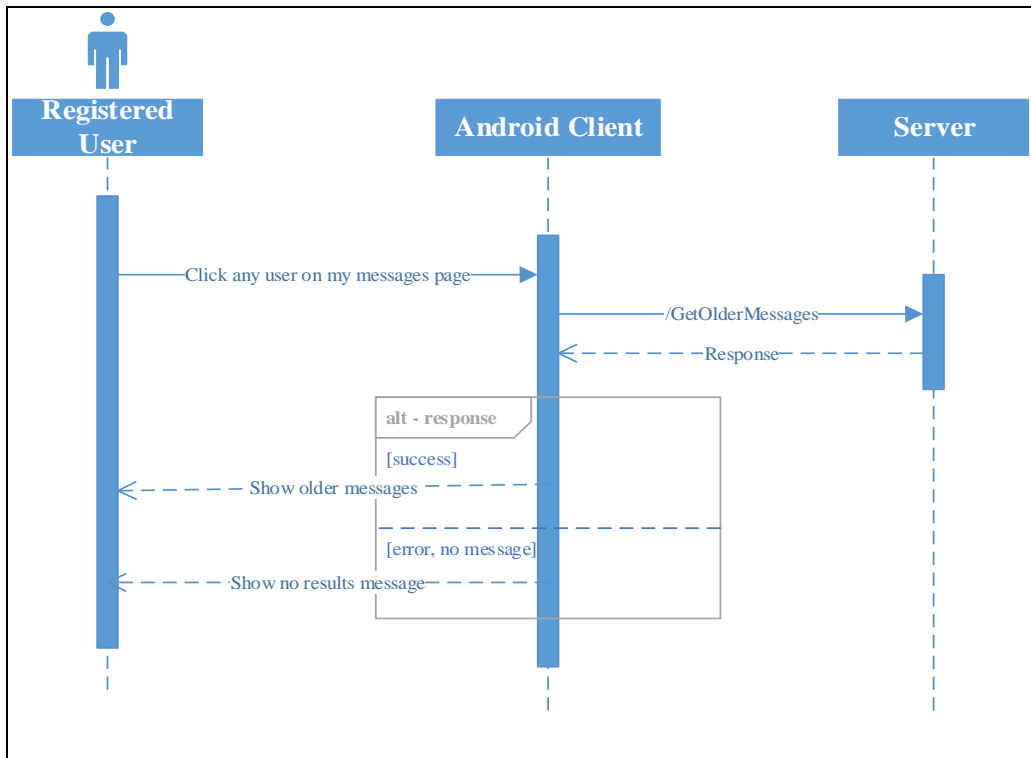
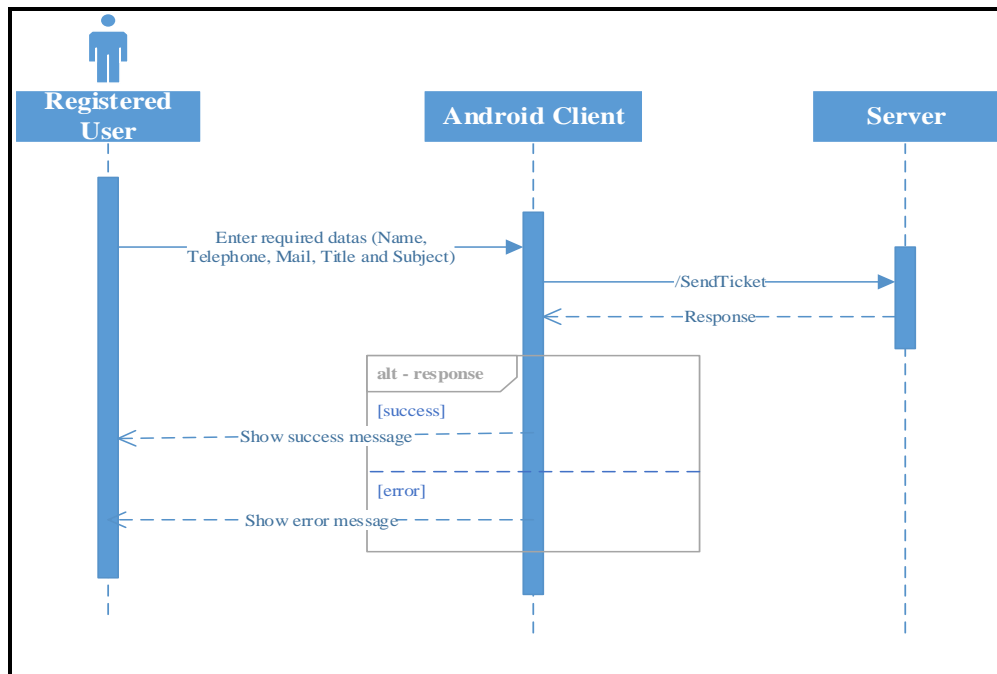


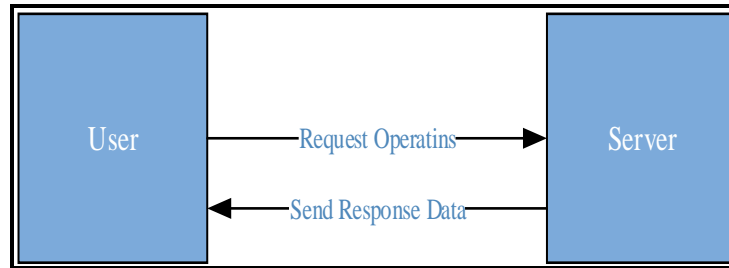
Figure A. 39: Send message to studio use-case sequence diagram



Appendix A.3 DATA FLOW DIAGRAMS

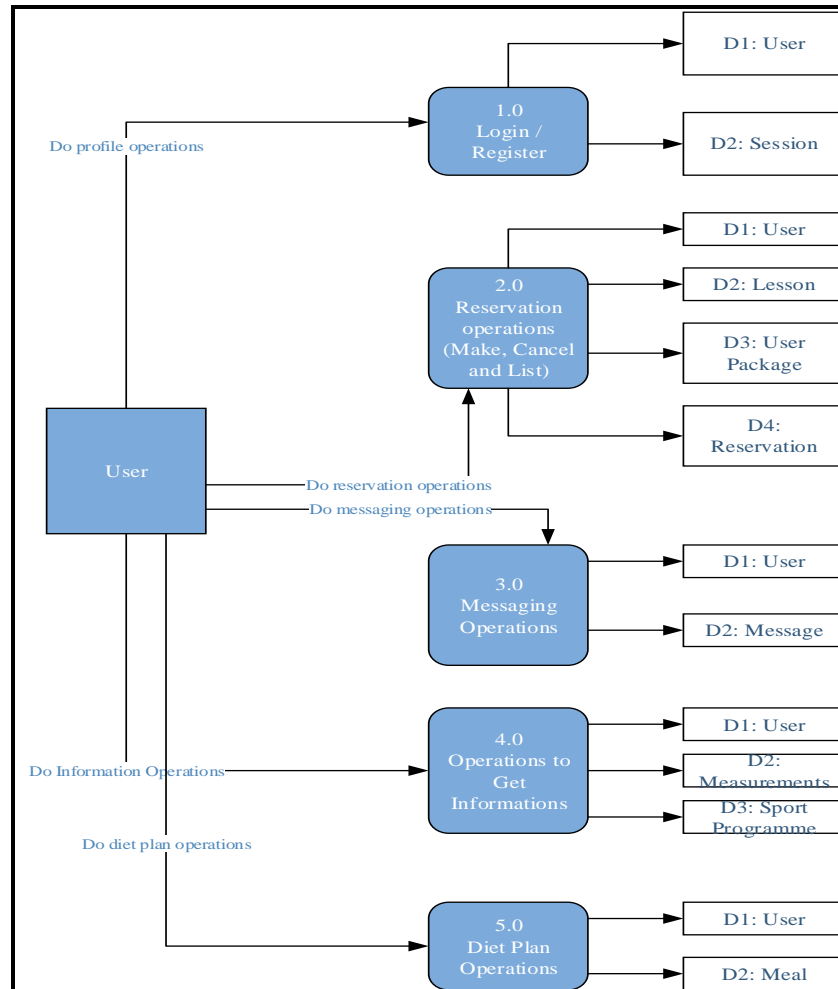
Appendix A.3.1 Context Level Data Flow Diagram

Figure A. 40: Context level data flow diagram



Appendix A.3.2 Level-0 Data Flow Diagram

Figure A. 41: Level-0 data flow diagram



CURRICULUM VITAE

Name & Surname: Yasin Tanrıverdi

Permanent Address: Maden Neighborhood, Kilyos Street, Number: 43, Floor: 4
Sarıyer/İstanbul

Place and Year of Birth: Sarıyer/İstanbul, 08.16.1994

Foreign Language: English

Primary Education: Sarıyer Fatih Primary School - 2007

Secondary Education: Sarıyer Mustafa Kemal Anatolian Teacher High School - 2011

Undergraduate: Bahçeşehir University Computer Engineering – 2015

Postgraduate: Bahçeşehir University

Name of Institute: Graduate School of Natural and Applied Sciences

Name of Master's Program: Industrial Engineering

Publications: -

Working Life :

Company: Huawei Telecommunications
Job: Android Developer
Date: 03.14.2016 -

Company: Netaş
Job: Software Design Engineer
Date: 02.02.2015 - 03.11.2016

Company: Netaş
Job: Intern
Date: 09.15.2014 - 01.30.2015

**THE REPUBLIC OF TURKEY
BAHCESEHIR UNIVERSITY**

**INFORMATION SYSTEM DEVELOPMENT
FOR A COMPANY IN FITNESS BUSINESS**

Master's Thesis

YASIN TANRIVERDİ

İSTANBUL, 2018