Project name:

MYTHIC YOGA FLOW

Done by:

Ivaylo Iliev Rashevski

Coach:

Christian Bell Bastlund

PBA Software Development

KEA-Copenhagen School of Design and Technology

February 2016 - Jun 2017

Hand in: Monday, 4th of September 2017

Vision

This is the final report on the Software Development Bachelor project called "Mythic Yoga Flow" , which involves a digital product that helps people, who are keen on yoga to practice easier, everywhere and with given professional instructions.

The report represents the choices, ideas, inspirations and the development process of "Mythic Yoga Flow". The required technical knowledge in theory and applying it in practice in order to be created the final product.

The final goal of the report is to present the relation and importance between the acquired theory knowledge from the Software Development study curriculum and the ability to apply it on practice and manufacture a product.

Glossary

Sanskrit - *the primary language of Hinduism*

Asanas - *comfortable sit for meditation*

Storyboard- *project file in Xcode, showing an overview of all user interface elements*

SDK - *Software Development Kit. Is typically a set of software development tools that allows the creation of applications.*

WBS - *Work Breakdown Structure. Is used for breaking down a project into easily manageable components.*

Xcode - *Integrated development environment for Mac operating system*

Version control system - *System that tracks and provides control over changes to source code.*

IDE - *Integrated Development Environment. Is an application that facilitates application development.*

Swift - *Open-source programming language that can be used on the Mac to target all of the Apple platforms: iOS, macOS, watchOS, and tvOS.*

MVC - *Model View Controller. Is a software architectural pattern.*

UI - *User Interface. Is the series of screens, pages, and visual elements, like buttons and icons, that you use to interact with a device.*

Table of Contents

Reading guide

This report is separated in ten main sections. In first one I will introduce you to the project. I will meet you with the company, their product and I will describe my first meeting with them.

In the second section, I will talk about how and why I chose this project, my inspiration and a little description of what actually is yoga about.

In part number three are placed the objectives and project scope.

Fourth part contains a detailed FURPS+ requirements.

In part number five I make analysis. I talk about risk management and potential threads. I define target audience and describe possible stakeholders.

Sixth section describes what kind of development methodology I use. Why and how I used it.

Section number seven is about my choice of technologies and development tools used for building the product.

Eight part contain patterns with detailed information and examples of how I apply the design pattern and a diagram showing the behaviour of the application. Plus detailed navigation step by step through the application.

Section number nine represents testing. Section includes usability and acceptance testing.

The last part number ten is an overview of the development process, personal evaluation and further opportunities for the product.

“Appendix” of this report is delivered as a separate document. There you can find curious information about the images, topics in the application and list of references, which contains literature, websites and tutorials that helped me making this project.

1. Project outline

In the final semester of my educational program Software Development at Copenhagen School of Design and Technology, I have to carry out a final project. The project has a limited timeframe. It has to represents development of a real life software project, usually for a company, using the gained knowledge and skills during my studies. The project must be manufactured together with the company that needs a project solution in order to achieve the project objectives. I was accepted by a company named “Nu Boyana Film Studios”, which I describe in next sections, for developing an iOS mobile application. In the report I documented the workflow and process of this project, and showing in practice the knowledge and skills I gained during my studies at Copenhagen School of Design and Technology.

1. 1 Company description and their products

Nu Boyana have been making movies in their idyllic location for more than half a century. Their enthusiasm, experience and skills had defined them to be one of the most important film studios in Europe. Nu Boyana Film Studios are owned by one of the longest-running independent film companies in Hollywood - Nu Image and Millennium Films. Over the years they have serviced hundreds of feature films, including big-budget movies like : Criminal, The Expendables 1,2 and 3, London Has Fallen, 300 and many others.
Their special effects department can offer a complete SFX package including mechanical, animatronics, pyrotechnical, atmospheric and underwater effects. They have fully equipped sound mixing studio and video post-production facility in-house. Their company owned Worldwide FX has grown in size and stature to become one of the major visual effects houses in Europe. You can check and see more on their web site : **https://nuboyana.com/**

Besides constructing stage decors, shooting movies and creating special effects, Nu Boyana has a web department. It is responsible for most of the digital products and support. The web development team supports the main web site, they are developing casting web site, which is still under construction in my opinion - **https://nuboyana.com/casting-home/**
Shortly, everyone is free to create a CV there. The user fill out their outward, skills, knowledge, experience, upload photos and many more. After that, when it come the time for selection of actors or static actors, the producer goes in the search engine and

start looking for persons who covers the requirements for the scene or the movie. The department also prepare presentations, making researches and developing side projects. They already have couple of yoga mobile applications, which unfortunately I can not present to you. But here I will provide you a link to the web version of the application I was involved in, which is also improving **https://myf.udaya.com/**

1. 2 Meeting the company

In this section I will talk about the working place, the development team, how I met with them and about the project opportunities they gave me and last but not least the project requirements.
After acceptance of my portfolio, I met with the head of the department for an interview. I presented myself, my knowledge and gained skills until the moment. He introduced me to their current projects and let me choose one in which I can join in. I picked up a project they have not started yet, because their developers had no knowledge in that area. The head of the department was the only developer who knew how to do it, but he was busy with other projects, which had higher importance and sooner deadlines. After we agreed on starting this project, I met with the rest of the web development team. We all were located in one big office, except the head of the department, he had separate office. The team consisted 3 developers, 2 designers, a system administrator and the head of the department who had many roles like developer, scrum master and responsible for QA. After all we started discussing the project requirements and their vision about the project. Acceptance criteria was done and the direction of the project was clear, together with the team we made a simple prototype, to visualise the project. All of us were daily in the office, so our communication was on a very high level. After discussion for choice of development methodology we had to follow, we were ready to start with that project.

NOTE: Link to the prototype ***: https://projects.invisionapp.com/share/MEATWBECQ#/screens/223197790_00***

2. Inspiration

Here I am going to tell you about why I chose exactly this project, what was my motivation and inspiration that were keeping me during the development of that product.

The current product - “Mythic Yoga Flow” is part of a course during my “Software Development” programme, called - “Internship”. I had various of projects (offered by the internship company) I could work over during the course, but at that point for me there was a critical situation with another course from the education programme - “iOS mobile development” . I was about to misspend my last chance for passing this course, because of lack of knowledge. Fortunately one of their suggestions was to develop an iOS mobile application. Here comes the moment when I said to myself - this is my chance to gain some knowledge and experience, which can help me dealing with the trouble I was in. And I took the decision. I accepted the task to build an application from scratch with lack of knowledge. It was challenging, interesting and exciting, which was keeping my attention. My motivation was that I was actually doing two in one - doing my internship course and gaining knowledge, which will help me for the other one, at the same time. I utilised every second in exploring and learning. In the end the skills and the knowledge acquired helped me finishing successfully both courses.

In the next section - “Background” , I will try to explain what is yoga about.

2. 1 Background

Yoga is a group of physical, mental and spiritual practices and disciplines which originated in ancient India. It is a 5000-year-old Indian body knowledge. Driven from the *Sanskrit* word *yuj,* Yoga means “union” in the most common sense. There is broad variety of yoga schools, practices and goals in Hinduism, Buddhism and Jainism. One of the yoga principles meaning is : “Yoga, as techniques of controlling the body and the mind”. There you reach the union, the symbiosis between body and mind. Yoga is not just exercise and *asanas* . It is the emotional integration and spiritual elevation with a touch of mystic element, which gives you a glimpse of something beyond all imagination.

3. Objectives and Project Scope

The main purpose of this project is to make it easier for people who want to practice yoga, but they face some difficulties. People who are travelling frequently and can not attend the classes at yoga club. Busy people, who has no time to lose for travelling and prefer to practice at home. Those ones who can not afford it. And similar situations, which can be an obstacle for them to do yoga.

The goal of the project is to eliminate these stumbling blocks and allow more people to practice yoga freely. For this aim is developed mobile application which can be uploaded/downloaded on a portable device (mobile device or tablet) and people can obtain professional instructions anytime and anywhere.

The development of the product is meant to provide a functional mobile application of professional yoga instructions for practicing. With the knowledge and skills gained during the Software Development course, I can develop this product and turn it into a final project. Acquired abilities gave me the opportunity to develop myself as a professional and showed me different directions where I can continue with my further professional growth.

4. FURPS+

After the discussion around the requirements with the team, I decided to create FURPS+ model. It is a useful technique, that represents the requirements after understanding with clients needs and necessities. You can always categorise the requirements by using that model, no matter the choice of development methodology in your project.

- **Functionality**
  All the customers can use the software application inside iOS devices to follow many different yoga classes, and dive in yoga world. All the information and videos can be found behind very nice, self explanatory and user friendly interface, which the user is able to personalise by setting a profile picture. The using of the application require registration. The system does not support payment functionality.

- **Usability**
  The User Interface, must be light, without distracting and unnecessary components. Easy to navigate through. And it has to be theme related.

- **Reliability**
  The application must work properly all the time. If any bugs or improvements occurs, a new updated version will be released.

- **Performance**
  Application performance must satisfy user needs. Navigating through the application should happens instantly. The load of the videos depends on users quality internet connection.

- **Supportability**
  The software should be flexible and understandable. Should be easy to be updated with any kind of changes - adding and removing new functionalities and features.

- **Supplementary Specifications**

- **Design constraints**
  Design must be detailed related with every stage of the application.

- **Implementation constraints**
  For manufacturing iOS mobile application we need IDE called Xcode. It contains the iOS SDK, which extends Xcode to include the tools, compilers and frameworks we need specifically for iOS development. The programming language is called Swift and for storage we agree on using Firebase Database.

- **Interface constraints**
  Database connection should be implemented reliable and the user must be logged in correctly.

- **Physical constraints**
  The system is intended to perform on mobile iOS devices.

5. Analyses

In this section I will make analysis before starting the project. I decided this analysis are important and going to help me to shape my project and lead me to the correct way I should proceed with it. I will analyse the risk management and the possible threads that can impact on the project. I will define the target audience. And I will define the possible involved parties in this project together with their power and interest.

5. 1 Target Audience

One of the beauties of the physical practice of yoga is that the poses support and sustain no matter how old or young, fit or frail you come to your mat. As you age, your understanding of *asanas* become more sophisticated. You move from working on the external alignment and mechanics of the pose to refining the inner actions to finally just being in the *asanas*.

Preferable pre conditions are :

- Find comfortable and private spot
- Your mat
- Comfortable, breathable clothing
- Bottle of water
- Be calm, do not stress

5. 2 Risk Management

The main purpose of making risk management analysis is to identify potential risks before they occur. After I analyse the potential threads that can impact negative on my project and can be obstacles for achieving the project goals, I make a plan how to mitigate the negative impacts. Risks can have a critical impact on the project, which may end up with successful project or failure. There are two different risk management methods : the reactive approach and proactive approach. Reactive approach is method where you identify the risks when they occur during the project development or after it. Which I think is kind of late and not responsible. So I decided to use the second one - proactive approach. With this method I identify the potential risks before starting the project. In this way, I try to avoid the threads and I am prepared with a mitigate solution in case they occur.

Risk ID	Type	Description	Probability	Impact	Risk Factor	Mitigation
1	TR	The videos are not streaming	2	5	10	Good support on the server, where the videos are streaming from
2	TR	The application size is too large	2	2	4	Optimise the images
3	BR	Company is not interested in project	1	4	4	Before starting the project, make a meeting and get known with everyones expectations
4	PR	Company requires many changes	2	3	6	Set the main requirements before starting the project
5	PR	Development environment stop working	2	5	10	Make an often backup. Reinstall the environment or change the machine
6	GR	Team members getting sick	2	4	8	Split the work between the rest of the team. Remote working.
7	GR	Lack of experience and knowledge in team	3	4	12	Quick exploring and practice. Help each other.
8	GR	Sudden leave of a team member	3	3	9	Leave clear code with comments.
9	PR	Some requirements are not achivable	1	3	3	Find or be prepared with backup option
10	PR	The application is not understandable	2	4	8	Make a usability testing

In the risk management table above you can see a list with possible risks. I decided to use numbers from 1 to 5, lower to higher, to indicate the probability of thread occur and the impact of it on the project. The risk exposure, or risk factor is calculated by multiplying the probability with the impact of the risk. Each identified risk is concluded with a mitigation plan, which explains a solution to avoid the risk or to lower the impact on minimum. The risks are organised in four different categories:

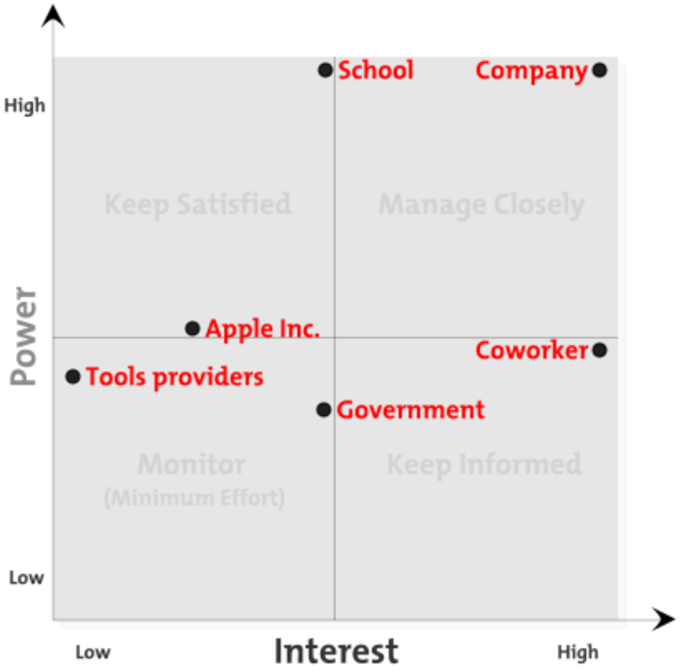
- **General risk (GR) :** Risks that generally can happen in every kind of project and are not specific to this project.
- **Project risk (PR) :** Risks that can directly postpone project delivery.
- **Technical risk (TR) :** Risks that influence the software quality and performance.
- **Business risk (BR) :** Risks that are directly related to products viability.

5. 3 Stakeholder analyses

Stakeholders are all individuals or groups who are involved and can impact the project in positive or negative way. It is preferable to create this analysis at early stage of the project. By doing it, I can identify who are the most powerful and interested parties, this can help me to improve the quality of the project. Here is a list with possible stakeholders that might impact with the project:

- School
- Company
- Coworker
- Apple Inc.
- Government
- Tools providers

Now I will place this possible stakeholders in a grid, depends on their power and interest.



This grid helps to identify the stakeholders and their influence over the project.

- **High power, interested stakeholders:** these are the people you must fully engage and make the greatest efforts to satisfy.
- **High power, less interested stakeholders**: put enough work in with these people to keep them satisfied, but not so much that they become bored with your message.
- **Low power, interested stakeholders**: keep these people adequately informed, and talk to them to ensure that no major issues are arising. These people can often be very helpful with the detail of your project.
- **Low power, less interested stakeholders**: again, monitor these people, but do not bore them with excessive communication.

6. Development methodology

The selection of development methodology was not such a problem. They have already worked with SCRUM and the workflow process was going really smooth. Anyway we had to be careful with the choice of methodology we were going to follow. The following several factors we took in consideration were going to lead us to the correct method:

- The team - who is going to take part in this project.
- The time frame.
- Flexible method adaptive to changes.
- Level of communication with customer
- Duration of the iterations
- Planning phase

We were looking for an agile methodology that following the principles behind the Agile Manifesto, which are the following :

Our highest priority is to satisfy the customer through early and continuous delivery of valuable software.

Welcome changing requirements, even late in development. Agile processes harness change for
the customer's competitive advantage.

Deliver working software frequently, from a couple of weeks to a couple of months, with a preference to the shorter timescale.

Business people and developers must work together daily throughout the project.

Build projects around motivated individuals. Give them the environment and support they need, and trust them to get the job done.

The most efficient and effective method of conveying information to and within a development team is face-to-face conversation.

Working software is the primary measure of progress. Agile processes promote sustainable development. The sponsors, developers, and users should be able to maintain a constant pace indefinitely.

Continuous attention to technical excellence and good design enhances agility.

Simplicity--the art of maximizing the amount of work not done--is essential.

The best architectures, requirements, and designs emerge from self-organizing teams.

At regular intervals, the team reflects on how to become more effective, then tunes and adjusts its behaviour accordingly.

Scrum is a management framework designed for incremental product development that follows Agile Manifesto. The main advantage of SCRUM is that you can adapt it to your own project. It is a framework that you can use different techniques and process by your own choice. SCRUM is not a defined process for building software or any kind of project, it is agile and this is the main reason we ended up with it. There are couple of things that are important to be understand and clear before using SCRUM.

- Roles - Different roles are assigned to the team members
  - Product owner
  - Scrum master
  - Developers

Product owner - Holds the vision for the product. Determines what needs to be done and sets the priorities to deliver the highest value.

Scrum master - Helps the team. Protecting the Scrum process and preventing distractions.

Development team - Builds the product. Takes on and determines how to deliver chunks of work in frequent increments.

In this project, the owner of the company had the role product owner. The head of the department acted like SCRUM master. Me and one of the designers were the development team. The rest of the people in that department were also taking parts, but not that often and important.

- Process - SCRUM follows a process with several phases, events and artefacts.
  - Sprint
  - Product backlog
  - Sprint backlog
  - Sprint planning
  - Daily scrum
  - Sprint review

The Sprint is a short, consistent cycle no longer than four weeks. The goal is to have an iteration short enough to keep the Development Team focused, but long enough to deliver a meaningful increment of work.

The Product Backlog is an ordered list of everything that might be needed in the product and is the single source of requirements for any changes to be made to the product. The Product Owner is responsible for the Product Backlog, including its content, availability and ordering.

The Sprint Backlog is the set of Product Backlog items selected for the Sprint, plus a plan for delivering the product increment and realizing the Sprint Goal. The Sprint Backlog is forecast by the Development Team about what functionality will be in the next increment and the work needed to be deliver.

Sprint Planning is the work to be performed in the Sprint. This plan is created by the collaborative work of the entire Scrum Team.

Daily Scrum is a 15 minutes time-boxed event for the Development Team to synchronize activities and create a plan for the next 24 hours.

Sprint Review is a meeting that is held after the Sprint in order to evaluate and inspect what has been done, to change the product Backlog if needed and brainstorming what can be done to improve the next sprint.

After reviewing SCRUM, we were already definitely sure that it will fit our needs and lead the project in the right way with high quality and organisation. We were ready to follow it. The team decided to have 3 iterations (sprints), each 10 working days long. Together with the product owner we made user stories and split them into tasks. After the product owner prioritise the tasks, the needed hours were estimated for each one of them. Below you can see a table representing user stories, their priority and time required.

User story	Priority	Estimated hours
As a user of the mobile application I want to be able to		240
Create a registration account	2	20
Log in successfully	2	20
Log out successfully	3	18
Change my password	4	20
Access the phone gallery	5	8
Access the phone camera	5	8
See all types of yoga courses	1	16
Able to watch videos from all the courses	1	16

We left almost half of the estimated time free in case of unexpected changes, improvements and corrections. We agree on that based on our previous experience with other projects. In the Sprint Planing phases the entire scrum team is involved with taking decisions which items should be worked on during the sprints and delivered first. At Sprint Review meetings we evaluated and inspected what has be done during the past Sprint and what can be improved for the next one in order to increase our productivity. We had daily face to face meetings to discuss what each of us has done

previous day and to check if anyone needs help with his task. The idea is to get closer to the goal and not to leave behind.
During the final sprint we face a problem. It was an obstacle for the team to connect the project with already existing database which was used by the web version of the application. We were pressed by the time frame and after discussion with the team and product owner we ended up with a backup solution. The new task was to connect the application with independent database. We handled the problem and could finish the sprint successfully after the acceptance test was approved.

7. Choice of Technologies

Before you start developing a certain product - application, website etc. , you have to be sure in the correct choice of tools and technologies. And also the ones you feel most comfortable with, that matches your style as a developer.

One of the requirements for the current product were to be an iOS mobile application. First step from the research leads you to the fact, that you need an “Apple” machine (laptop/computer) which runs macOS(operating system). Second step is to find an *IDE* which has the features you need to design, develop and debug an application. This “Apple” *IDE* is called *Xcode*. *Xcode* also contains the iOS *SDK*, which extends *Xcode* to include the tools, compilers and frameworks you need specifically for iOS development.

NOTE: You can find and download Xcode from Apple store. Your macOS and Xcode versions has to be compatible. Is recommended to keep them updated and have latest versions.

Here we come to the programming language. It is called “*Swift*”. *Swift* is a compiled programming language developed by Apple Inc. for iOS, macOS, watchOS and tvOS. It is open source, so all you need is an idea, which you have to translate it in *Swift*.

For storing information (usernames and passwords) I decided to use Firebase Database. From the short research I have done for how to connect database to my iOS application and store information, this was the process I understood very quick. I do not know if it is the most efficient and security way, but I had time pressure, the deadline was close, it suited my requirements and I really liked it. The data is stored and synchronised with NoSQL cloud database. Data is synchronised across all clients in real-time and it remains available when the application goes offline. Instead of typically

HTTP requests, the Firebase Realtime Database uses data synchronisation - every time data changes, any connected device receives the update within milliseconds. Firebase applications remain responsive even when offline, because Firebase Realtime Database *SDK* persist the data to disk. Once connectivity is reestablished, the client device receives any changes it missed, synchronising it with the current server state.

NOTE: Firebase Database does not go together with Xcode, so you have to install it to your project. In the next lines I will try to to explain shortly how this can be done.

First step is to go to : *console.firebase.google.com*, and log in with your Google account. Second step is to create a project and then click on - Add firebase to your application.

NOTE: Your Firebase project name must match with your Xcode project name.

After that you follow four steps. Everything is clearly explained on the website.

1. Type iOS bundle ID (you can find it in *Xcode*)

2. Download *GoogleService-info.plist* file and move it into the root of *Xcode* project.

3. Open the terminal and navigate to the *Xcode* project.

A. Create a "Podfile" by typing into terminal : ***$ pod init***

B. Open "Podfile" (in your project folder) and add : ***pod 'Firebase/Core'***

C. Save the file and run in terminal : ***$ pod install***

4. Go to your project in *Xcode* -> AppDelegate class and add the following code:
import Firebase
FIRApp.configure()

8. MVC as a Design Pattern

One of the most common, powerful and useful design patterns is Model-View-Controller. That is the pattern this project is build on. The *MVC* is an architectural pattern that separates an application into three main logical components: the model, the view and the controller. Each of this components is responsible for to handle specific development aspects of an application.

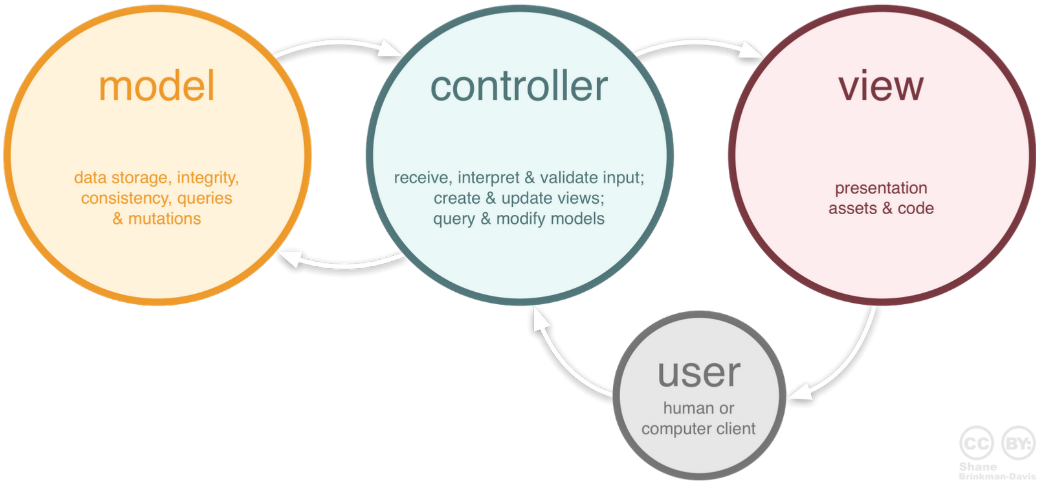

model
data storage, integrity, consistency, queries & mutations
controller
receive, interpret & validate input; create & update views; query & modify models
view
presentation assets & code
user
human or computer client
CC BY:
Shane Brinkman-Davis

Figure 1.

Model component communicates with all the data-related logic that the user works with. This can represent either the data that is transferred between the View and the Controller or any other business logic-related data.
View component is responsible for all the *UI* logic of the application. The View will include all the *UI* components such as text fields, check boxes, buttons etc. , that the end user interacts with.
Controller is used as an interface between the Model and the View to process all the logic and incoming requests, manipulate data using Model and interact with Views to render the final output.

In the following sections I will describe the structure of the project and how I apply the pattern with all its components.

8. 1 Models

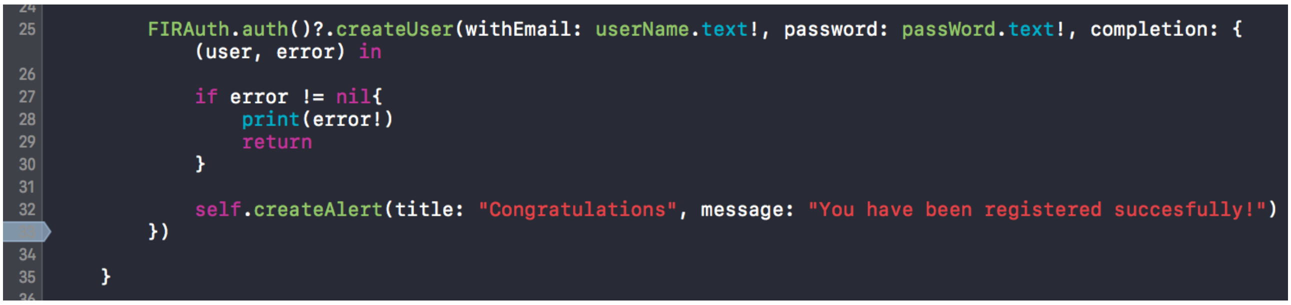
Following the product requirements, there is no need of complex database. It is required to perform successful log in and log out. For that purpose Firebase Database provide service called "Authentication". This service has three options.

1. Users - place where all users are stored in a table with information related for each one of them. Admin right as following : Add user, Reset password, Disable account, Delete account.

2. Sign-in-method - Here you define what kind of sign in you prefer for your application. Eg. Email/Password, Phone, Google, Facebook etc.

3. Templates - structures of "noreply" emails, which you can modify according to your application. Emails for : email verification, email change, password change. There is also an SMS verification option.

After successful installation of the Firebase Database to the application, it is a must to import "FirebaseAuth". This is framework which provides already built-in methods, which create the code connection between application and database.



```
25          FIRAuth.auth()?.createUser(withEmail: userName.text!, password: passWord.text!, completion: {
                (user, error) in
26
27              if error != nil{
28                  print(error!)
29                  return
30              }
31
32              self.createAlert(title: "Congratulations", message: "You have been registered succesfully!")
            })
34
35      }
```

Figure 2.

The image is showing one of those already built-in methods - creating a user. Line 25 is the build-in method. The properties - "*withEmail*" and "*password*" , are waiting to retrieve information from the application. It has to be given the correct textfields - "*userName.text!*" and "*passWord.text!*" according to the information the properties required, in order to inherit the information from the textfields and create new account.

8. 2 Views

The views represents all the visual components that are visible to the end user and he can interact with them. The following image shows the whole overview of the application. All screens, the connection between them (arrows) and all *UI* elements.

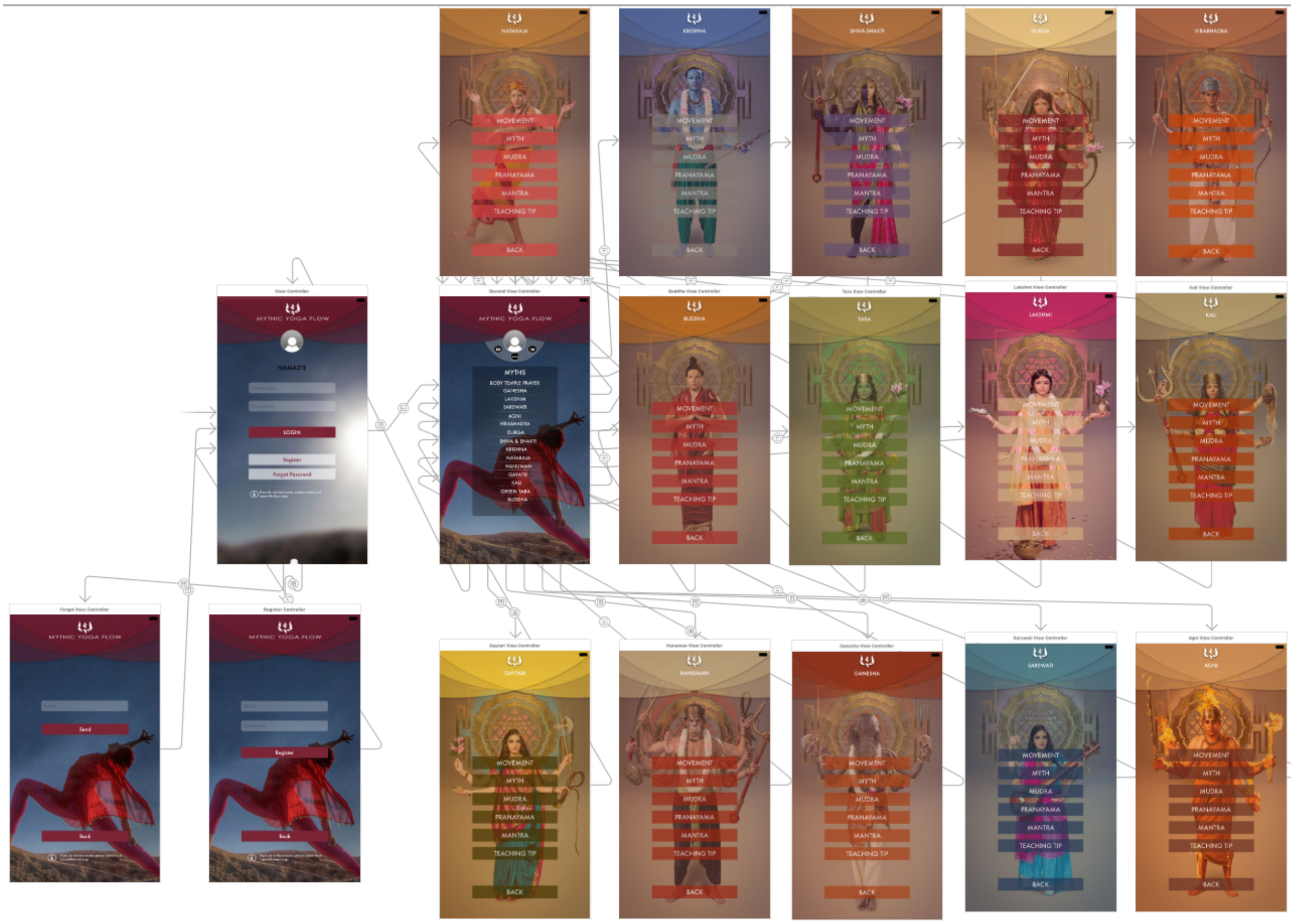


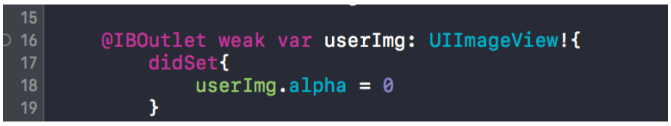
Figure 3.

NOTE: You can find and see more clear the user interaction expressed as state machine/state diagram in one of the next sections.

8. 3 Controllers

In *MVC* design pattern Controllers are in the middle of the process. Everything goes through them, nothing happens without their knowledge. They are responsible for many tasks and I decided to show you two of them. First one is “Update Views”.

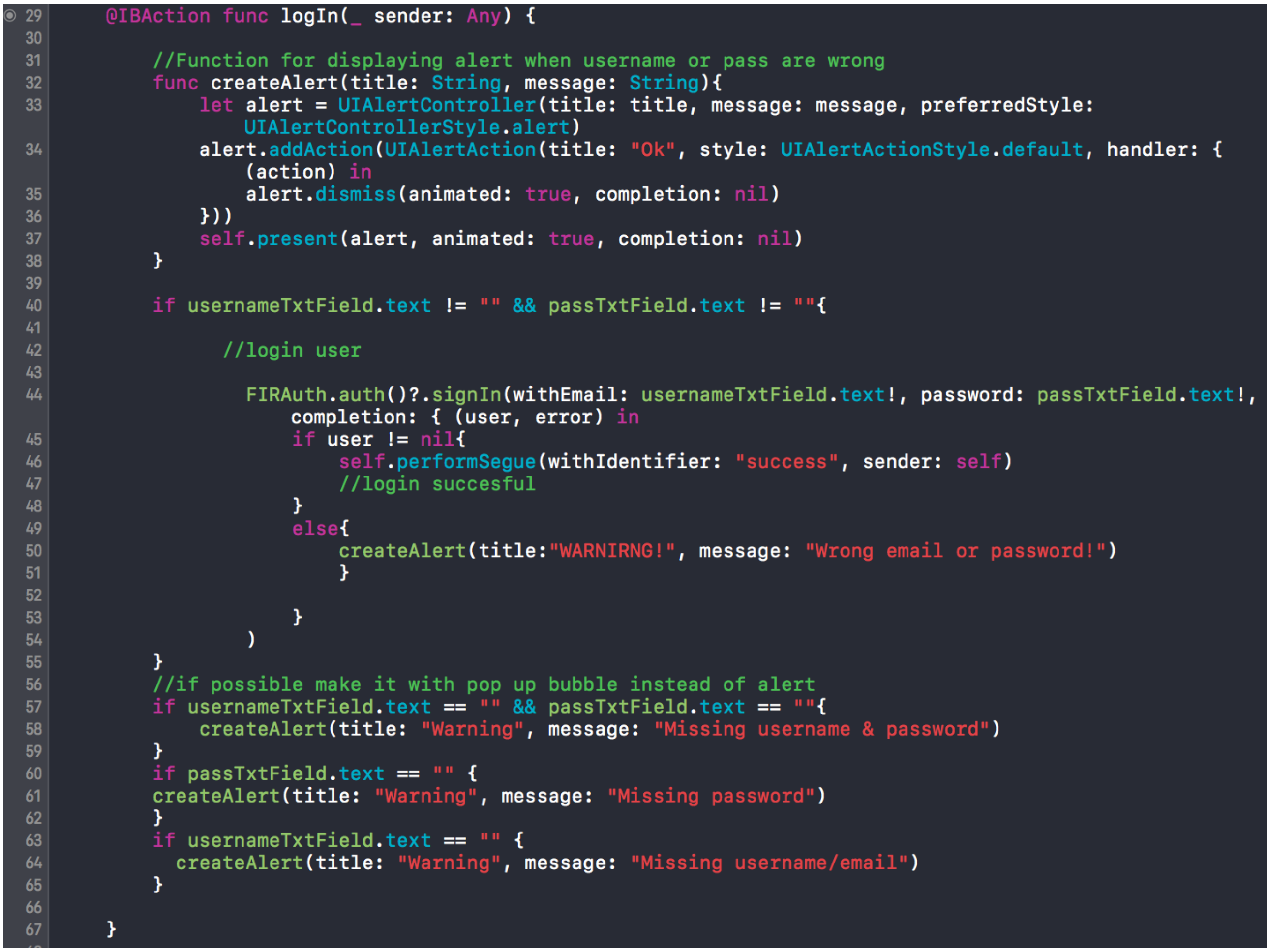
”Outlet” and “Action” those are the status of the *UI* elements. When is “Outlet” is a static and different properties can be set to it like shape, color, visibility etc. When it has “Action” status, means it is going to perform some action. In Figure 2 is shown create account action performed by “Registration” button. An *UI* element can have both status.



```
    @IBOutlet weak var userImg: UIImageView!{
        didSet{
            userImg.alpha = 0
        }
```

Figure 4.

Figure 4 shows an *UI* element with “Outlet” status. It is image variable *“userImg”* , which is from “*UIImageView*” class (e.g. for buttons there is “*UIButton*” class, for textfields is “*UITextField*” etc.) . The alpha (visibility) property of the image is set to “0” . Later this property will be set to “1” in an update view function.



```swift
@IBAction func logIn(_ sender: Any) {

    //Function for displaying alert when username or pass are wrong
    func createAlert(title: String, message: String){
        let alert = UIAlertController(title: title, message: message, preferredStyle:
            UIAlertControllerStyle.alert)
        alert.addAction(UIAlertAction(title: "Ok", style: UIAlertActionStyle.default, handler: {
            (action) in
            alert.dismiss(animated: true, completion: nil)
        }))
        self.present(alert, animated: true, completion: nil)
    }

    if usernameTxtField.text != "" && passTxtField.text != ""{

          //login user

            FIRAuth.auth()?.signIn(withEmail: usernameTxtField.text!, password: passTxtField.text!,
                completion: { (user, error) in
                if user != nil{
                    self.performSegue(withIdentifier: "success", sender: self)
                    //login succesful
                }
                else{
                    createAlert(title:"WARNIRNG!", message: "Wrong email or password!")
                    }

                }
            )
    }
    //if possible make it with pop up bubble instead of alert
    if usernameTxtField.text == "" && passTxtField.text == ""{
        createAlert(title: "Warning", message: "Missing username & password")
    }
    if passTxtField.text == "" {
    createAlert(title: "Warning", message: "Missing password")
    }
    if usernameTxtField.text == "" {
      createAlert(title: "Warning", message: "Missing username/email")
    }

}
```

Figure 5.

Second property of Controllers I am going to show you is validate input. In Figure 5 is created an alert function, which is going to pop up when input data is missing. This check is done by “if” statements. When the data is correct, it performs the segue successfully and if the data is wrong, the alert pop up again with warning message. This check is done by “if-else” statement.

8. 4 State pattern

My decision for showing the application in a finite state machine is because I want to visualise the application behaviour. You can see on Figure 6, how a state changes provoked by actions and inputs.

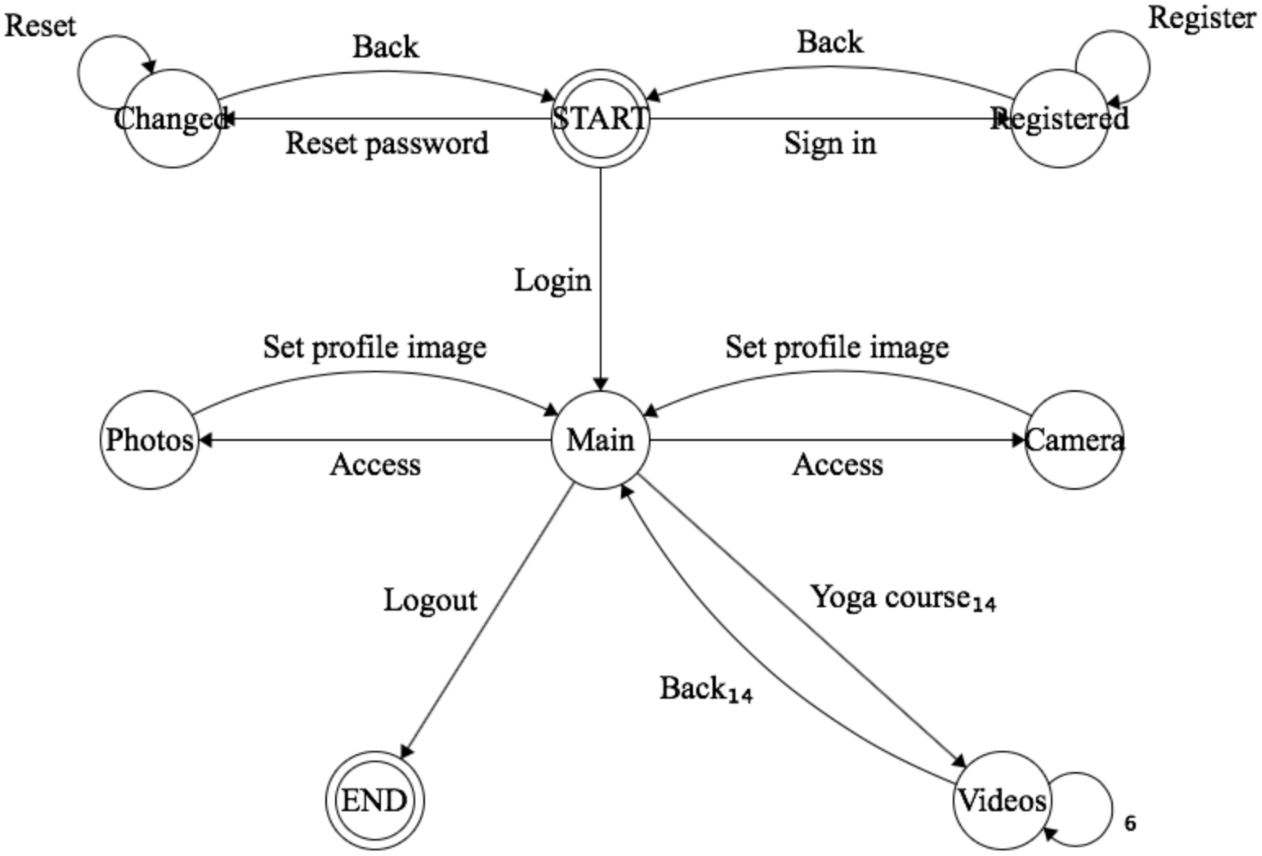


Figure 6.

Circles represents different states of the application. Arrows are transitions that changes the state from one to another. This change is provoked by an action (pressing a button) or combination of input (data) and action. The text next to each transition (arrow) is transition condition. It explains where the transition is leading to. The indexes are showing the amount of the same states and transitions.

8. 4. 1 Navigating through the application

Start point is the screen that appears to the user when he turn on the application. The following options from that stage are:

- To create an account. After he press “Register” button, he is leaded to new screen with fields required to fill out for new account. After sending the request by pressing the button “Register”, the user can go back by pressing button “Back” and log in successfully.

- To reset the password in case he forgot it. After he press “Forgot password” button, he is leaded to new screen with fields required to fill out in order to get new password. After sending the request for new password by pressing the button “Send”, the user can go back by pressing button “Back” and log in with the new password successfully.

- To log in by typing the correct username and password in the required fields and press the button “Log in”, which is going to lead him to a new screen - main screen.

In the main menu, the user will find three “bullet points” with different functionalities. Two of them are for setting a profile picture.

- One is for accessing the mobile device camera, so you can shoot a photo and set it as a profile picture.

- The other one is for accessing the mobile device gallery-the place your phone stores the photos you have taken. You can choose between them and set a profile picture.

- The third “bullet” is log out function, which leads to the “Start” screen.

The core of the application comes with the “MYTHS” button. By pressing it, a drop down menu opens with 14 different yoga types. By selecting one of them, a new screen appears with thematic 6 videos for the current type. The user can navigate through this videos and by pressing “Back” button, he is back to the main menu, where he can select another type.

9. Testing

9. 1 Usability testing

I decided to make this type of testing in order to reduce the risk in my project. I wanted to see if my application is user friendly and enough understandable. The test was done among relatives, friends and their parents, people of different age and interests. The flow of the usability testing has three main stages which I followed :

- I created a list of tasks that the users are likely to perform in the application. The person who are testing is following the list.
- I observe and listen to any comments by the person who is performing the test.
- I reveal if people are able to perform the list of tasks and if they have any difficulties while doing it.

My list of tasks was the following:

A. Create an account.
B. Log in.
C. Log out.
D. Reset your password.
E. Log in with the new password.
F. Set a profile picture by using the camera.
G. Set profile picture by using the gallery.
H. Go to "AGNI" yoga clases.
I. Play "MANTRA" video. Go forward and backward in the video.
J. Go to "KALI" yoga classes.
K. Log out.

The usability test was done among 10 people. I will share my conclusions of the test. 1% of the people failed the test due to lack of experience with smartphone device. 2 % of the people were confused at stage "F" and "G". Their comments were that the icons are too small and not very clear. The majority of the people, 7% could perform all the tasks easily.

9. 2 Acceptance testing

I decided to create acceptance testing upon the acceptance criteria. The acceptance test was evaluated by the product owner after checking if the acceptance criteria was met and the task was done. The following table represents test procedures and acceptance requirements.

Test no.	Test procedure	Acceptance Requirement	Critical	Test Result
1	Log in	The application must have a register functionality	Yes	Accepted
2	Log out	The application must have a log out functionality	Yes	Accepted
3	Change password	The application must have a reset password functionality	Yes	Accepted
4	Access mobile device gallery	The user must be able to set profile picture from gallery	Yes	Accepted
5	Access mobile device camera	The user must be able to set profile picture by using camera	Yes	Accepted
6	Navigation through all the yoga classes	The user must be able to navigate forward and backward	Yes	Accepted

10. Conclusions

In this final section i am going to evaluate the project, my work and talk about future perspectives of the product - “Mythic Yoga Flow”.

When we take in consideration the short period of time, the lack of knowledge in the beginning and the fact that one person was developing the application from scratch, it is really hard to cover all the requirements and to lead the application to its final stage. I consider the product at its current stage as a primary prototype of “Mythic Yoga Flow” iOS mobile application. There are possible improvements and changes that can be implemented in order to frame the product at its final stage and putting it into use.

Further perspectives for the application are uploading it on “Apple Store”, where people can download it for free or they have to pay. As far as I know about their business plans(the company I developed it for), they have several more mobile yoga applications, with different interface design and behaviour. I took a look over the prototype of another yoga application they have. It was designed like a game. You need to practice one yoga class (and perhaps cover some requirements) in order to unlock the next one. Their plans are to provide those applications only to their customers. And another plan I heard about is to expand and develop them in cross-platform software. That means - the applications can be used on different devices with different operating systems.

I believe the choices I made during the development process of this product are the most efficient. I grown up as a developer more then I expected in this short time frame and critical for me situation. The experience and the knowledge i gained are really valuable for me. The development of iOS mobile applications really suits my desires and helped me to see my future perspectives. I can say I am glad and proud of my work. Because in the beginning I could not imagine I am capable of manufacturing product like this. There is infinite more to be learned, but it happens step by step and every step takes its time.