Design and Implementation of an Information System Using NFC

GYM_Watch 3.0

Functional Specifications, Technical Specifications, and Approaches.

Last Update: January 20, 2017

Offenburg University of
Applied Sciences, Germany

M.Sc. Communication and Media
Engineering (Team Project)

Team Members:
Enayetullah Rasuli, Maxim Abdullin

Project Guide:
Prof. Katharina Elisabeth Mehner-Heindl

Table of Contents

General Notes

This documentation provides a detailed information about the steps in a software project called "Design and Implementation of an Information System using NFC – Gym_Watch 3.0". The project is made as part of M.Sc. Communication and Media Engineering study program in Offenburg University of Applied Sciences, Germany. As a starting for the project, we determined the functional specification which includes goals to be achieved by the project. In the second part, different technical approaches used in this project are explained including architecture, language, and libraries.
In the third part, we describe different managerial and technical approaches to achieve the goals of the project. Finally, a brief explanation of the project code is included and the code itself will be provided along this document for further reference.

Overview

"Gym Watch" is the codename for the project developed as a team project work in M.Sc. Communication and Media Engineering during the Winter Semester 2016/17 with this vision:

> **Gym Watch allow manager, coaches and athletes to interact with system to get or submit relevant information to meet their specific requirements.**

Manager role does registration/deregistration, report of attendance, screening (daily/historical) etc. A coach role accesses the list of athletes they are couching, produces reports, gets feedback, and posts the training routines. An athlete role access their profile, registration status, training routines and online payment confirmation.

Design Goals

All design and engineering decisions will be taken with this principle in mind:

Beauty is in simplicity and simplicity is ultimately the winner.

The design of user interfaces will be focused on reducing the number of choices the user has to make.

About The Rollout

The first version of the product will be available by end of January, 2017. This means that the initial version will certainly not be the optimal in every aspect and will not contain every feature. Throughout this spec, **v2** will be used to indicate features will not be part of initial product but will be added later.

Functional Specifications (Requirements)

Functional specification describes business goals from a user perspective without considering deep technical implementation.

Here is a brief summary of how Gym Watch 3.0 functions:

1. A person with manager role accesses the system via their desktop application, or web browser. Under sub windows or pages they have the options to register new clients, retrieve client's information, and access the data, new or historical. While registering members, they assign a coach to each one.
2. NFC card is used client registration to store the data in their NFC tag as well as the database.
3. After NFC tag is scanned, the data stored in the card will be retrieved and displayed on the window.

Functional Components

There are three major components:

a) The backend database that will store the client and other information related to gym.
b) Desktop application.
c) Website.

The clients who use the product will install the desktop application on office computer. For the web application, the data can be hosted on cloud.

User Experience

Desktop Application:
When the person in manager or admin role run the Gym Watch on the office PC, a window with login request is displayed. In this window there will be three menus. These menus are used to interact with the system and the NFC tag. The information fields include:
ID, Name, Date of Birth: Date selector, Membership (Type, Start Date, End Date), Address etc. For GUI, additional features such as generating reports, well designed windows with graphics can be included in further development.

Web Application:
On the web application it will have multiple PHP pages that will make it more dynamic. As of the final stage, using bar charts the application can show statistics of the clients. The web application can be extended to include the core functionalities implemented in desktop application with the same logic. The webpage also includes general information about the gym and offers.

Technical Requirements

This section provides information about technical requirement. It is targeted for software developers of the project in order to have the idea about using specific technology to achieve the goals.

Software Architecture

Design concerns encountered during the project was whether to start from user interface, model or database. After a research about this in software projects, we found out that through close cooperation of team members a combination of both top-down and bottom-up approaches can be used to speed up the development while ensuring consistency with functional requirements (project business goals).

To decrease the complexity of the code, and increase modifiability and reusability we tried to integrate a Model View Controller (MVC) architectural approach in the project. This was decided with a vision in mind to quickly integrate individual code lines developed independently.

Desktop Application: The desktop application is written in Java. We will use Eclipse IDE for compiling. For Object Relational Mapping, we use Hibernate Framework.

Website Application: The website will be based on HTML/CSS for the client side and PHP for the server side. CSS will be used for formatting and positioning.

Components

The project include the following component from a technical perspectives:

Software Components

The main software components includes:

- Core (Model, View, Controller)
- NFC (Libraries and APIs for integration of NFC smart card reader into Java environment)
- Hibernate Framework for object mapping to the database (xml files, Object Mapping classes)
- Database (My

Hardware Components

The hardware components includes:

- NFC smart card reader (ACR122U)
- NFC tags (cards)

ACR122U Smart Card Reader supports ISO 18092. The standard ensure the compatibility of different card manufacturers.

ACR122U acts as intermediary device between PC and contactless tags via USB. The followings are its hardware specifications:

- Read/Write speed of 424 kbps
- Built-in antenna for contactless tag access
- Reading distance of 50mm
- Support for ISO 14443, part 4, card type A and B

- API for PC/SC

NFC header of data frame exchanged when communicating with NFC tag consists of the following fields listed in the table:

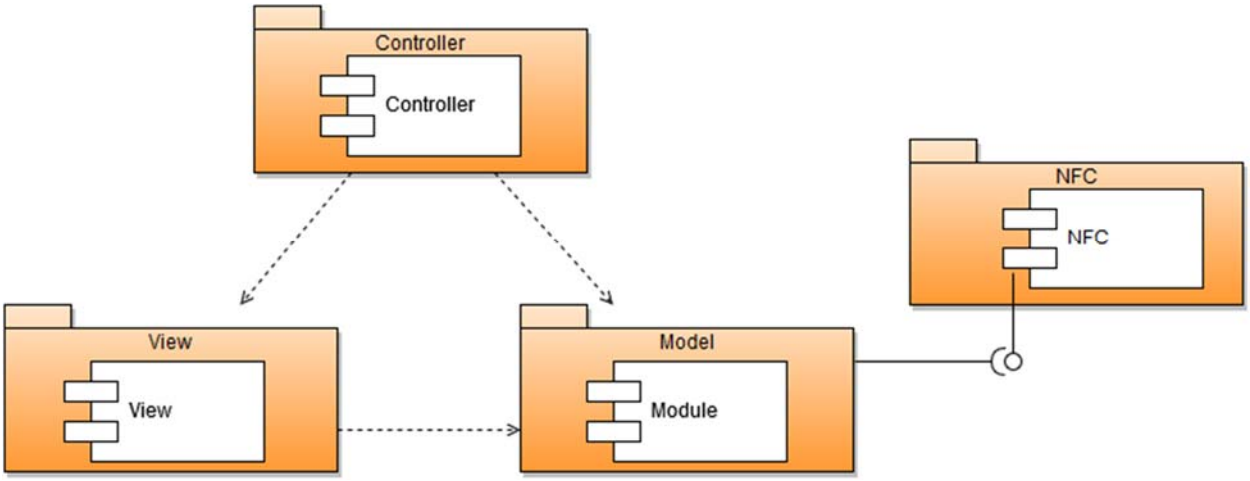
ATR											
Initial Header	T0	TD1	TD2	T1	Tk	Length	RID	Standard	Card Name	RFU	TCK
3Bh	8Fh	80h	01h	80h	4Fh	0Ch	A0 00 00 03 06h	03h	00 01h	00 00 00 00h	6Ah

Each NFC tag data read into Java code, consists of a number of rows and columns. Each row can be used to store a specific data related to a client such as id, name, and subscription data which can be mapped using a separate table in the database.

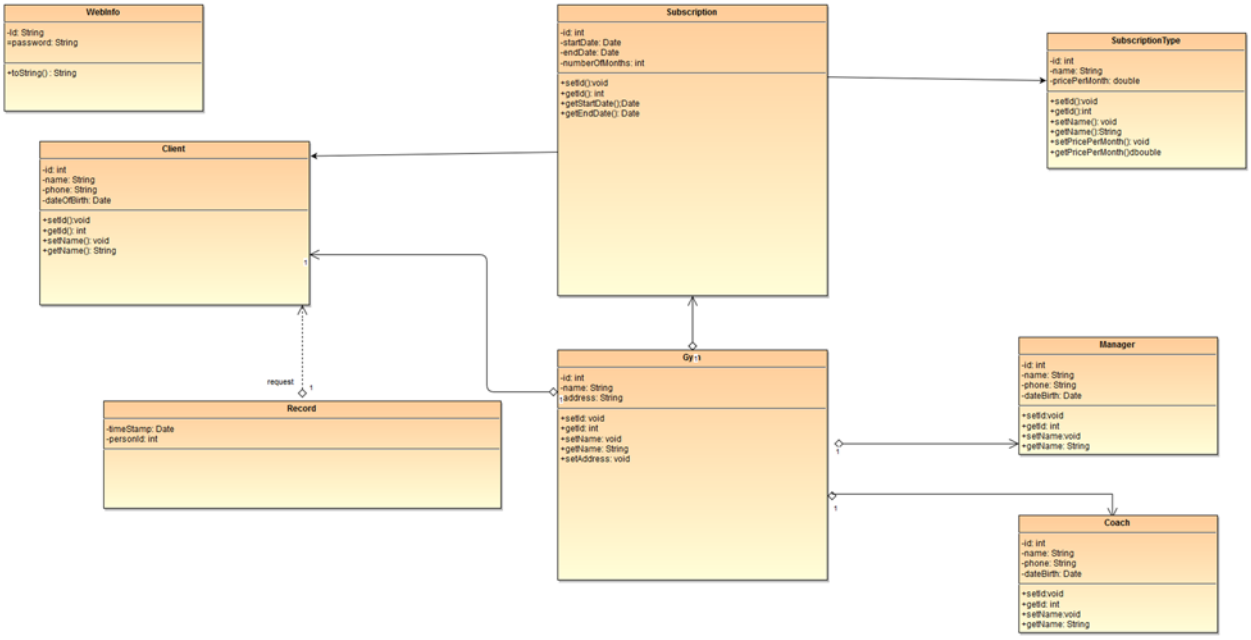
Diagrams

The following UML, relational diagrams were used for this project:

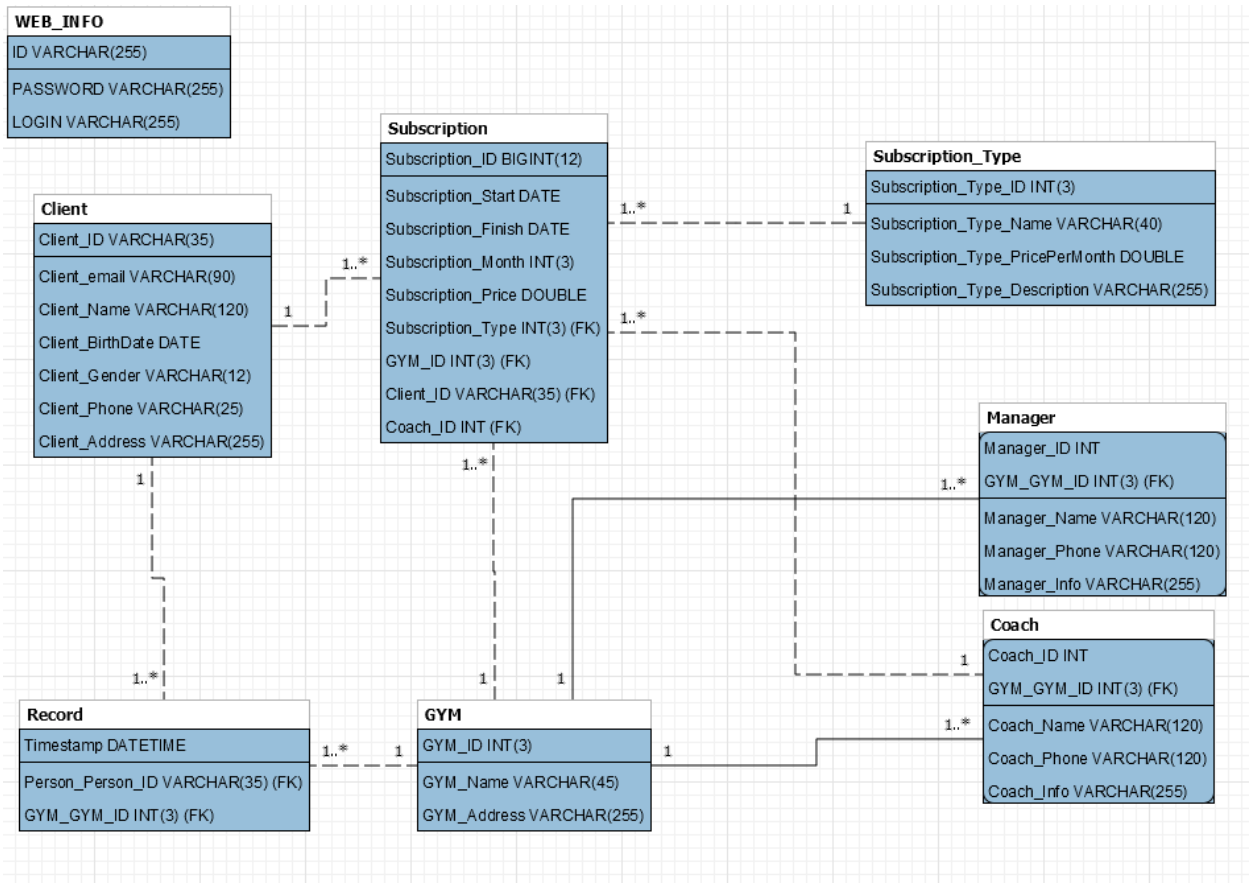
MVC (package) diagram



Core Model Class Diagram



ER Diagram



Code Convention

Always use the same name for the same thing in order to avoid confusion and complexity. For example the variable names used in HTML should correspond to the variable it gets stored in, which correspond to the name of column in the SQL database.

Java naming convention is important as it will make the coder easier to read. It implies that less time is spent to figure out what the code is doing.

Name	Convention
Class name	should start with uppercase letter and be a noun e.g. String, Color, Button, System, Thread etc.
Interface name	should start with uppercase letter and be an adjective e.g. Runnable, Remote, ActionListener etc.
Method name	Should start with lowercase letter and be a verb e.g. actionPerformed(), main(), print(), println() etc.
Variable name	should start with lowercase letter e.g. firstName, orderNumber etc.
Package name	should be in lowercase letter e.g. java, lang, sql, util etc.
Constants name	should be in uppercase letter. e.g. RED, YELLOW, MAX_PRIORITY etc.

SQL Tables are always in upper case and named in the singular form, for example, Client, Subscription and etc.

Approaches

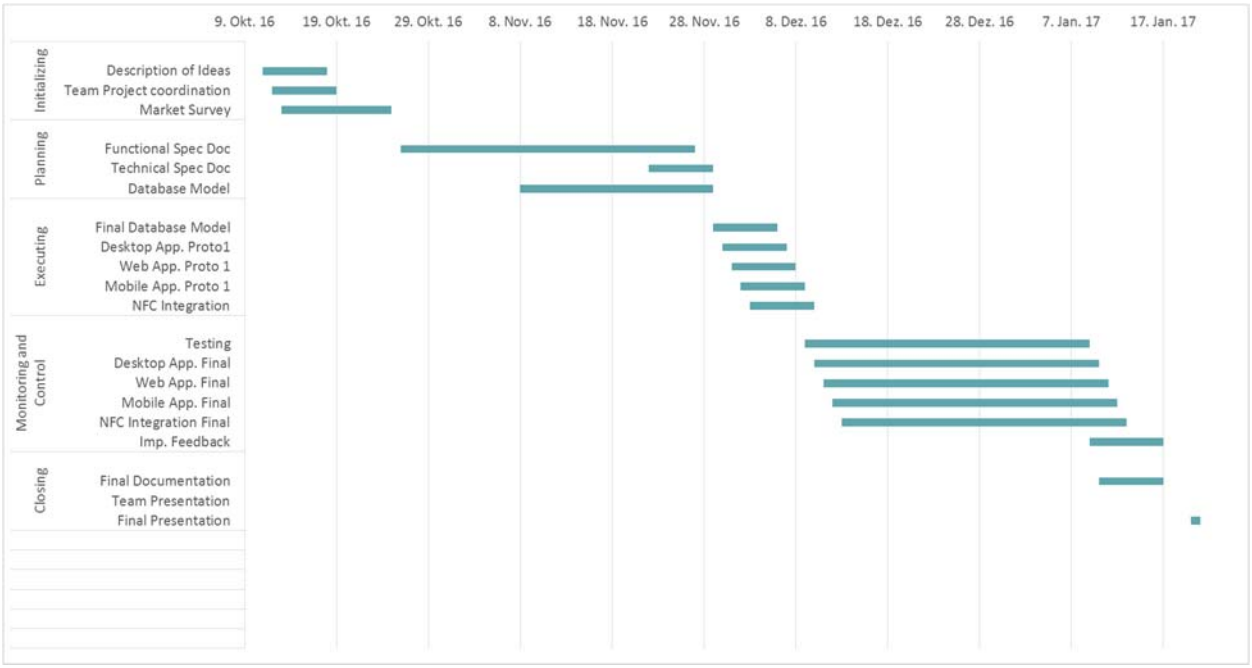
This section provides information about managerial aspect of the project and how the work was done in a specific time frame.

Project Team Roles and Responsibilities

As the project included different responsibility aspects, for better management, the roles were initially defined though not fully separated. Nonetheless, there was shifting between the responsibilities between the members.

Project Implementation

To achieve the project's goal, we divided the projects into work package in a specific time frame. The Gantt chart for the project is show below:



Project Schedule and Milestones

To realize the goal of our project we set the milestones from the beginning of our project

Results

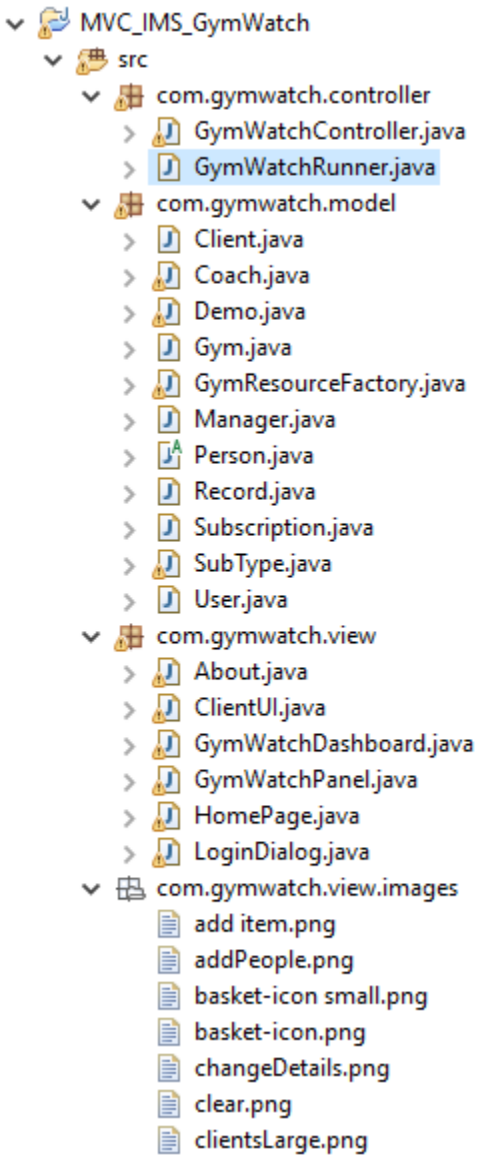
The project source code development went through two phases (trials). This initial phase was done before NFC integration. Due to Object Relational Mapping issue, the second trial by technical lead of the project was done in order to facilitate the NFC integration.

Initial Desktop Application with fine GUI

The structure of this source code is shown below. This was written by project architect.

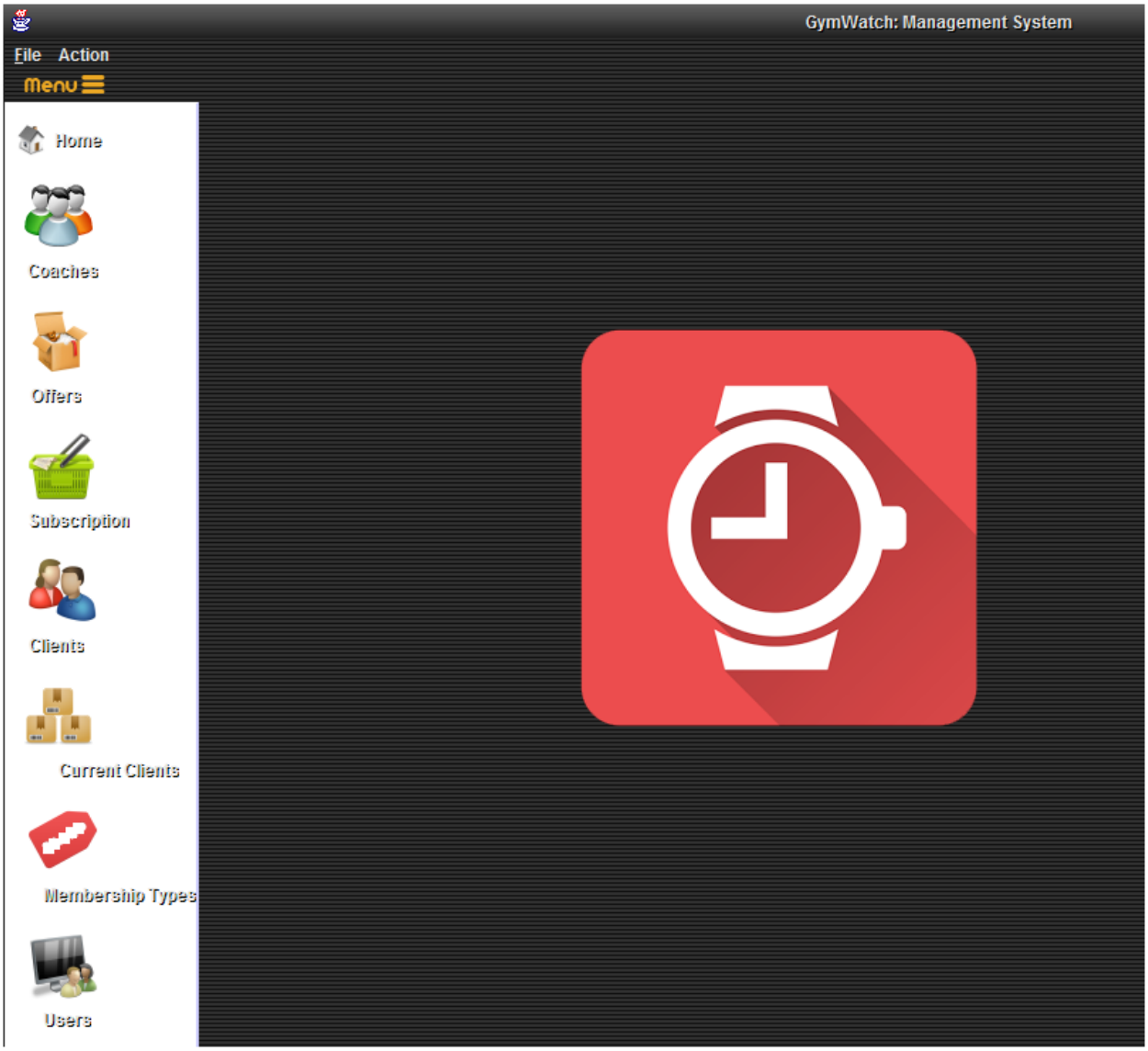
Note:

The inconsistency with database through hibernate suspended this code from being used in NFC integration. Below are some the screenshot of source code hierarchy and screenshots:

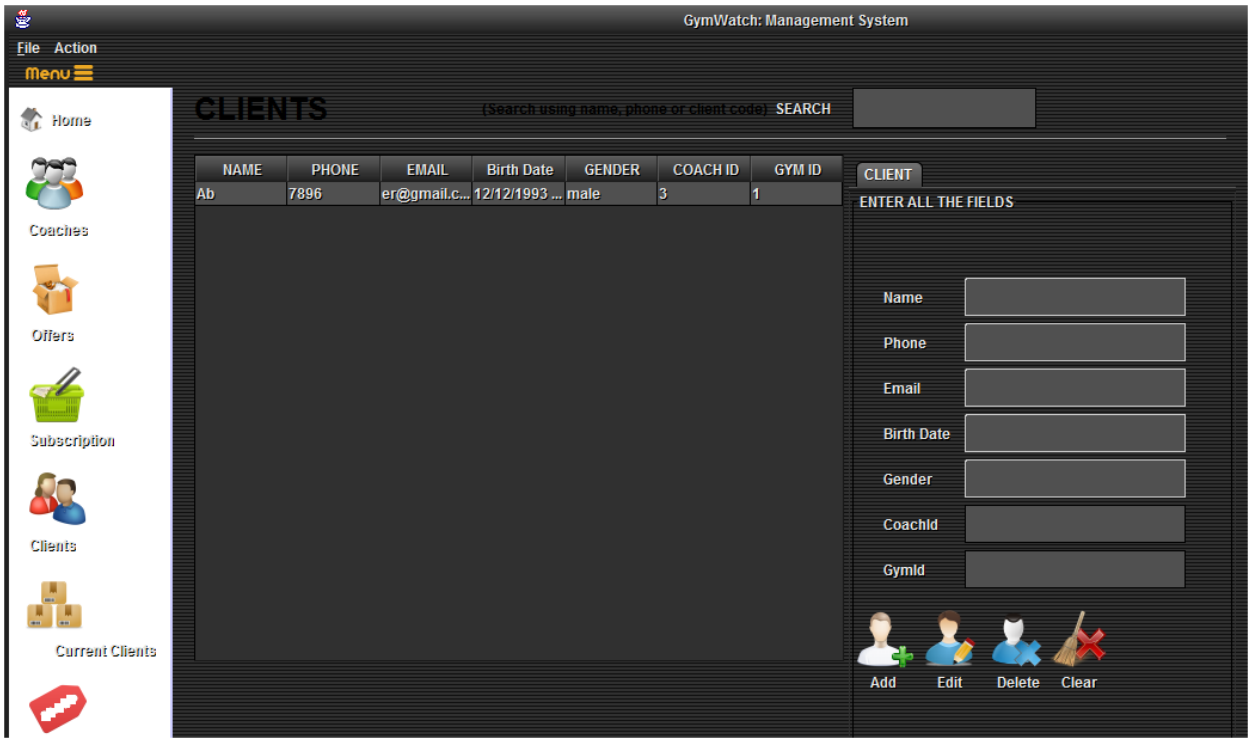


Login Page

Main Page


GymWatch: Management System
File
Action
Menu
Home
Coaches
Offers
Subscription
Clients
Current Clients
Membership Types
Users

Editing, Adding, Searching and Deleting Clients



Final Desktop Application with NFC Integration

The second source code, which includes NFC integration is shown below:

Website

Conclusion

Challenges Faces:

- Managing the growing lines of code
- Object Relational Mapping
- Integration of External Tools in Java

Future Scope:

The project can be developed further for more functionalities into GymWatch 4.0.

Reference

- Oracle documentation on specific technology used in this project.
- Dzone website for standard approaches to software projects.
- API of ACR122U