Requirements Specification

  Graded Unit 2

New computerised system for simply rugby

Aleksy Ruszała

2018

Table of Contents

[*1.* *Introduction* 2](#_Toc528539020)

[*1.1.* *Purpose* 2](#_Toc528539021)

[*1.2.* *Scope* 2](#_Toc528539022)

[*1.3.* *Research* 2](#_Toc528539023)

[*2.* *Specific requirements* 2](#_Toc528539024)

[*2.1.* *Functional requirements* 2](#_Toc528539025)

[*2.2.* *Non-functional requirements* 4](#_Toc528539026)

[*3.* *Resources* 4](#_Toc528539027)

[*3.1.* *Software* 4](#_Toc528539028)

[*3.2.* *Hardware* 5](#_Toc528539029)

[*3.3.* *Miscellaneous* 5](#_Toc528539030)

[*4.* *Project Timeline* 6](#_Toc528539031)

[*5.* *Business Model* 7](#_Toc528539032)

[*5.1.* *UML Diagrams* 7](#_Toc528539033)

[*5.1.1.* *Use Case Diagram* 7](#_Toc528539034)

[*5.1.2.* *Class Diagram* 11](#_Toc528539035)

[*5.1.3.* *Databinding Diagram* 14](#_Toc528539036)

[*5.1.4.* *Activity Diagram* 15](#_Toc528539037)

[*6.* *View Model* 26](#_Toc528539038)

[*6.1.* *Wireframe* 26](#_Toc528539039)

[*7.* *References* 54](#_Toc528539040)

1. Introduction
   1. Purpose

The purpose of this document is to provide a detailed specification of requirements for new computerized system that is creating for Simply Rugby organization. This specification is intended to be presented and approved by the client and as a reference point for the development team to create applications to meet customer requirements.

* 1. Scope

The project is to create new system called “Simply Rugby Management System” with the intention of facilitating trainers and other people managing the organization keeping updating records, personal data of members of this organization, detailed games played and training effects of each player. It is also designed to completely replace the current paper-based system. Currently only the administrator manages the data and in the initial iteration, only he will have access to the system after entering the password.

The advantages of this application will be centralized system, much faster and more efficient storage, modification and viewing of data. All data will be stored in the database.

* 1. Research

Before beginning the work of designing the application, research was carried out, which will be a source of inspiration during the development of the application. We managed to find several similar systems for managing club members that have very interesting functionalities.

One of the most interesting functionalities that comes from the SportEasy website offering the above-mentioned services is to use the calendar view to visualize events. (SportEasy, 2018)

The other is to define groups of members, i.e. Senior, Junior or Member, in order to better filter and prevent mistakes when completing forms, similar to the functionality of the Coach. (Coacha, 2018)

The research allowed for a more accurate analysis of what the future system might look like and what problems may occur during the design based on the old paper system.

1. Specific requirements
   1. Functional requirements
      1. Add, modify or delete an organization member

The system user should be able to add modify or delete organization members along with their personal data. Because in Simply Rugby you can name three different groups of members, the program should be able to assign a member to the appropriate group. The system should store data on three types of organization members: player member, junior member, non-player member.

* + 1. Displaying the list of organization members, trainings, games

The program must be able to view the list of members, trainings, games. It is necessary to be able to view the contents of the database and to be able to manage this data.

* + 1. Searching and filter players list
    2. To enable users to delete, edit and browse club members, the program should be able to search for them by filtering them.
    3. Add, delete and modify doctor, kin, guardian and adding or add or delete health issues details for each player

It must be possible for each player and junior to be able to hold contact details to his doctor and health issues records, only for senior player kin contact details and for junior player two guardian details and signature date of parental consent form.

* + 1. Parental consent form must be signed for each season

The system should store expiry date of the consent of the guardians

* + 1. Assign player to his main position on the pitch

The user should be able to assign to each player his main position played on matches assigned by the trainer

* + 1. Each member of the organization must have a SRU number

This should be required for insurance reasons.

* + 1. Judging individual player based on the skill list

The trainer should be able to assess the skills of each player and save results in the program. This assessment should be one for one user and it must be possible to update it. The player rating should be in the range from 1 to 5. The list of assessed skills should be determined in advance, and the trainer should be able to mark score for each of those skills. It should also be possible to add a comment for each marked score.

* + 1. Record undertaken trainings

The user should be able to document the training for each team. It is required to check the presence of players, all skills and activities undertaken also any potential accidents that happened. For each training must be required to provide date, time and place.

* + 1. Adding game records

The user can save the course of the game between the teams. The list of games should include date, team names and start time. Also, it need to track results and comments to every half.

* + 1. Access to the system

The first iteration will be made available to the administrator of the current system. He will be able to access the system will be given after providing static password.

* + 1. Connection to the database

All program data will be stored in the database.

* 1. Non-functional requirements
     1. Listing of trainings and games using the calendar view

Creating a special view that will visualize past workouts and matches in the timeline / calendar

* + 1. Sending relevant information to all members

User can prepare list of members and save or sent it forward. i.e. list of players, list of injury players

* + 1. List of payments of organization members fee

The program should track the payments of members of the organization

* + 1. Adding teams and assigning a coach and players.

The program must be able to assign each player / trainer to at least a team. The list of teams must be able to add and remove teams.

* + 1. Login panel

Only the user who has the access password can have access to the program. After entering your login and password, the program will allow him access to protected data.

* + 1. System of matching players based on their achievements

Based on the individual achievements of the players, the program will be able to match each player potential positions based on the requirements that must be met. The user on this basis will be able to make changes.

1. Resources
   1. Software

List of software used to create the project:

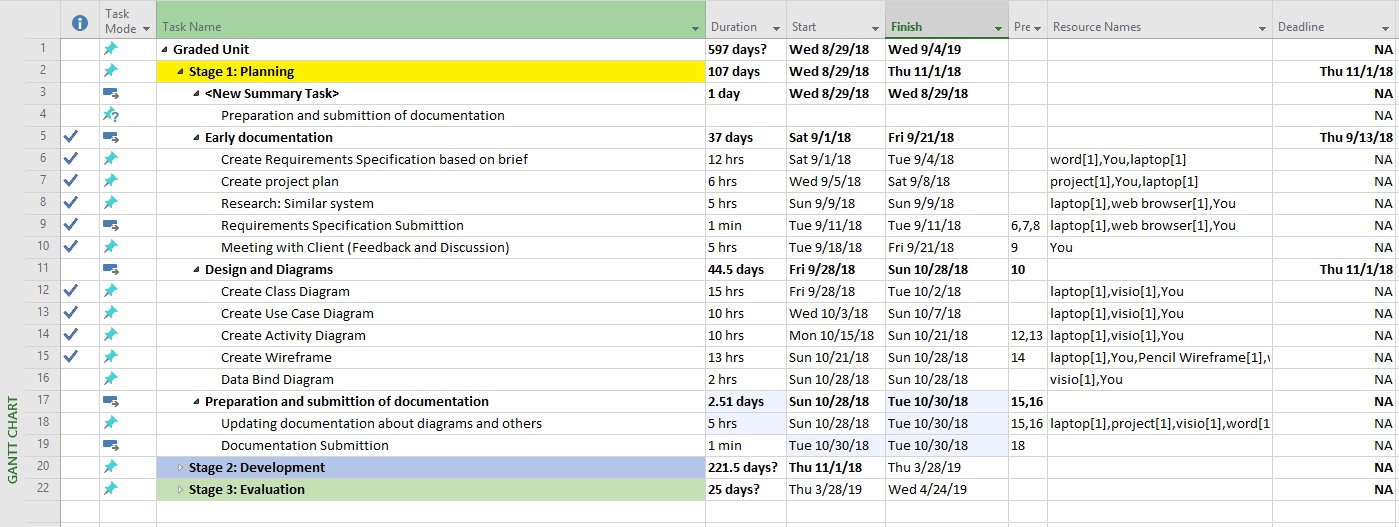
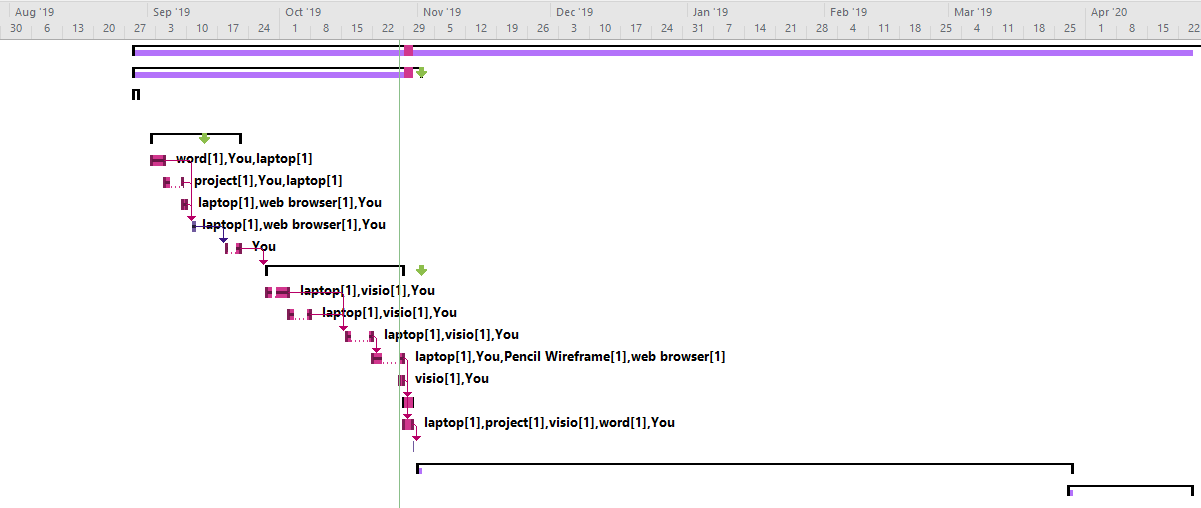
* Microsoft Office 2016
* Microsoft Visual Studio 2017
* Microsoft Visio 2016
* Microsoft Project 2016
* Evolus Pencil 3.0.4
* Google Chrome Browser
  1. Hardware

List of hardware used to create the project:

* Collage PC
* Dell Inspirion 15 Laptop
* MySQL Server
  1. Miscellaneous

Other resources include only the old paper documentation of the association

1. Project Timeline

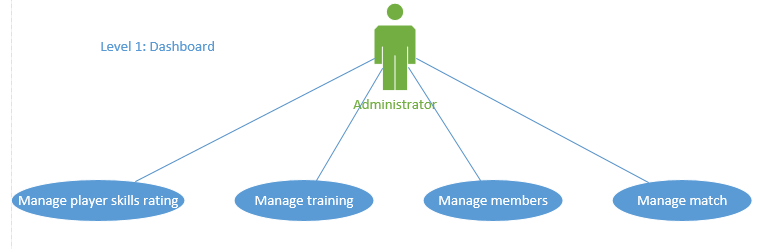
The picture presented below presents the tasks and course of these tasks during the design of the application.

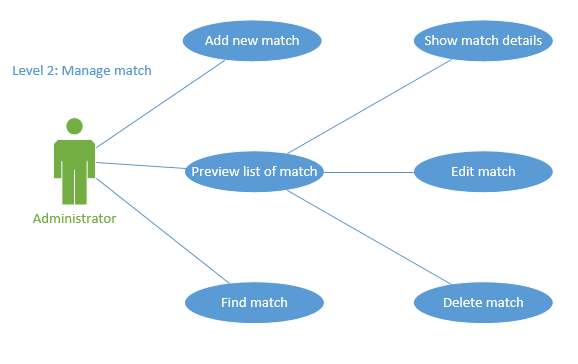
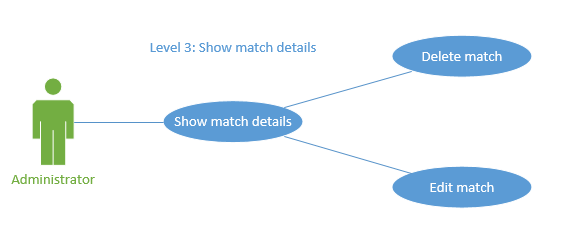
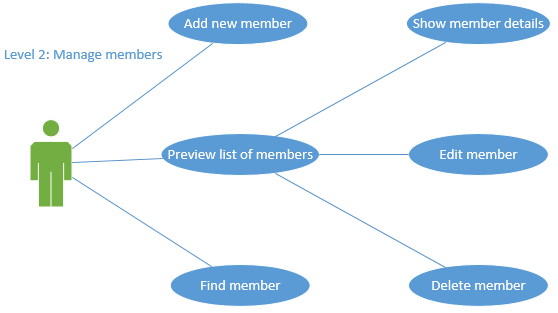
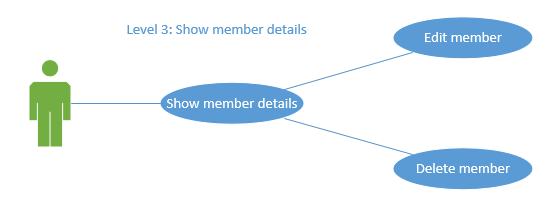
1. Business Model

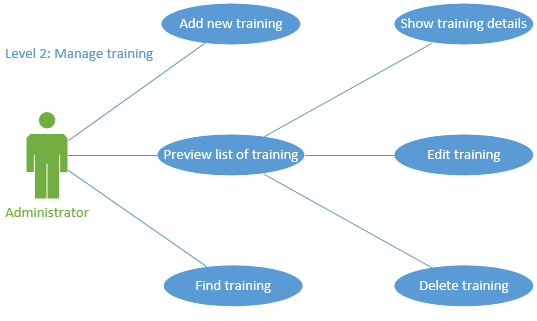
This section is dedicated to present the relationship between the program, the database and the user, and how they interact with each other. This section show the state for the first iteration of the program and may change during further development of the application, therefore it should be considered as an overview in the initial phase.

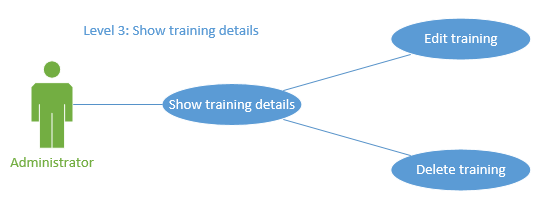
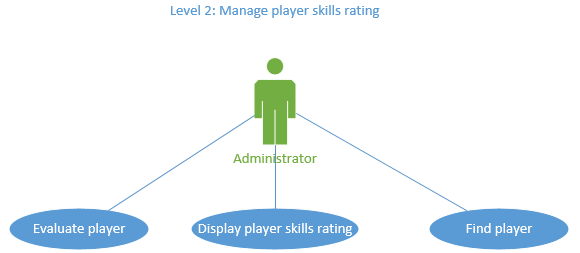
* 1. UML Diagrams
     1. Use Case Diagram

The diagram shows the administrator's interaction with each element of the application



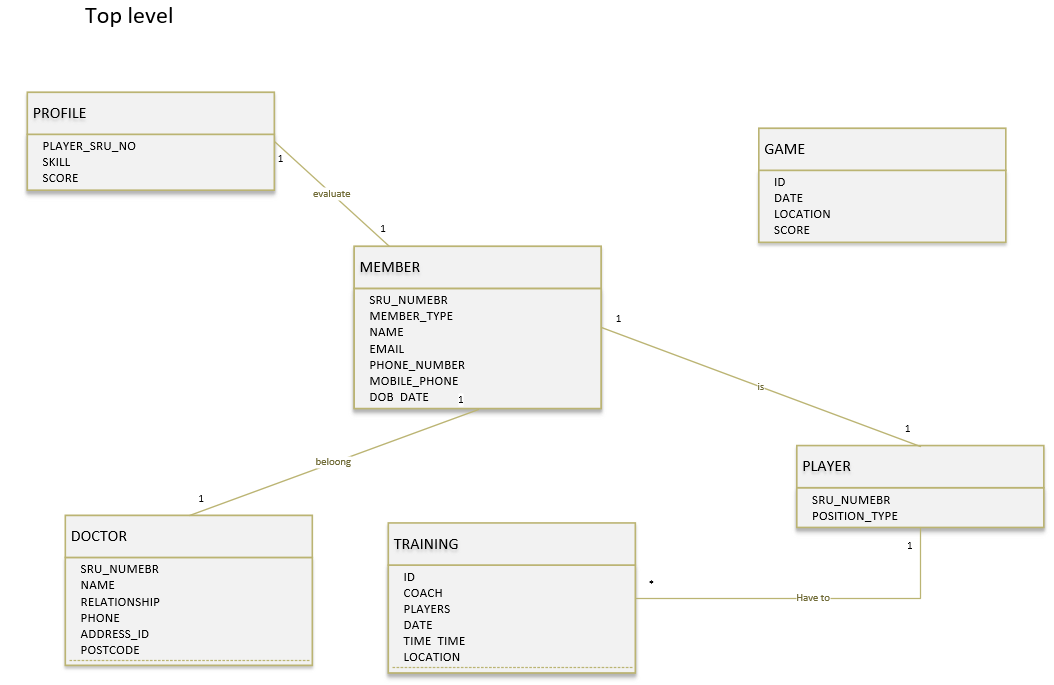
   

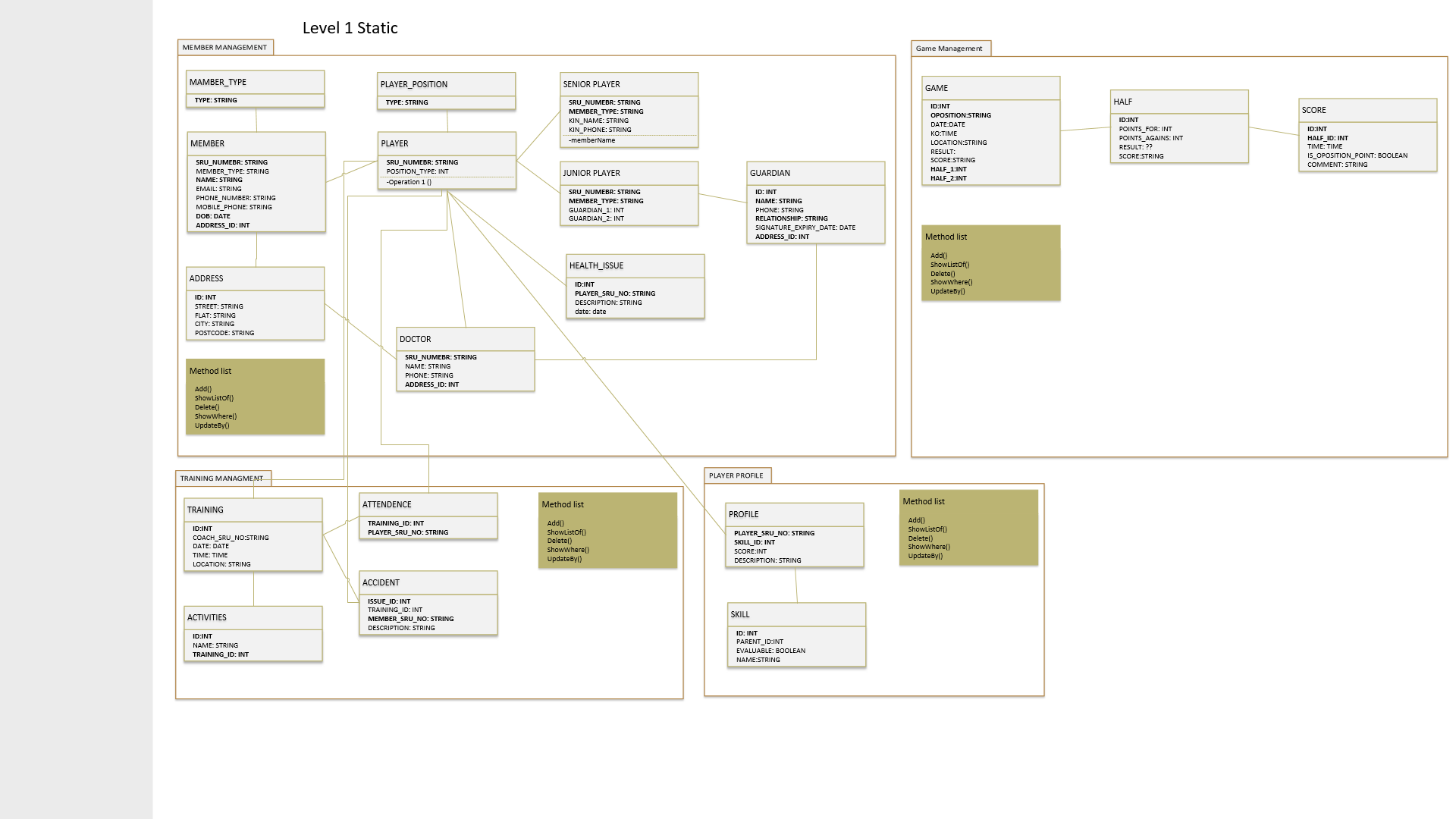


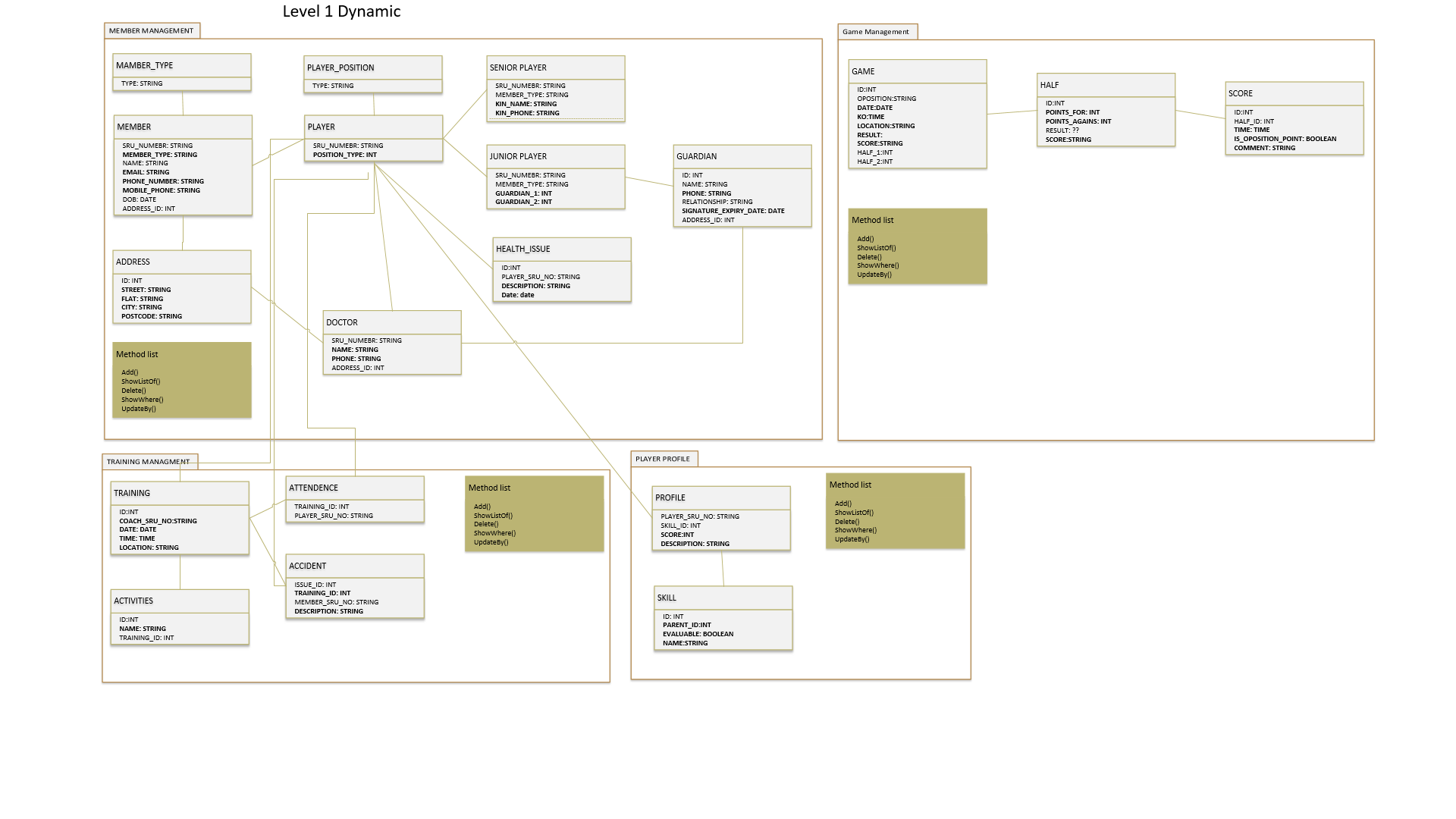
 

* + 1. Class Diagram

The diagram shows the list of classes that are needed to describe the stored data and actions

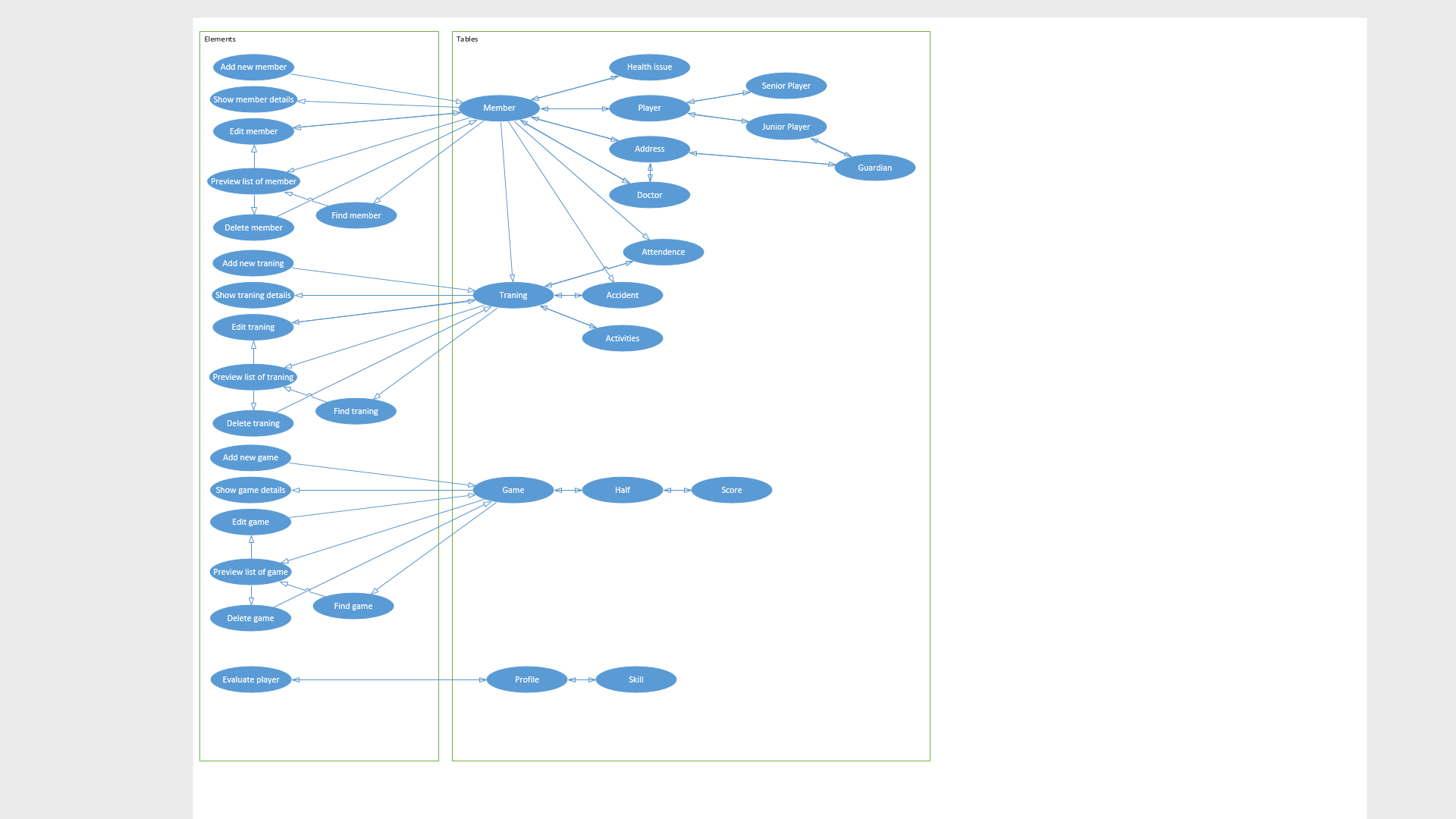






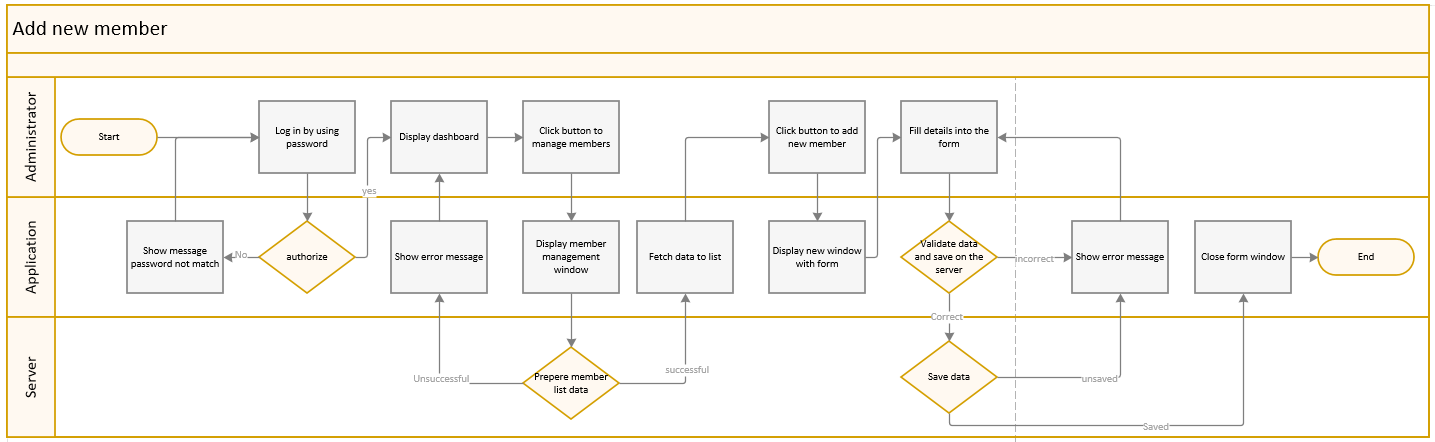
* + 1. Databinding Diagram

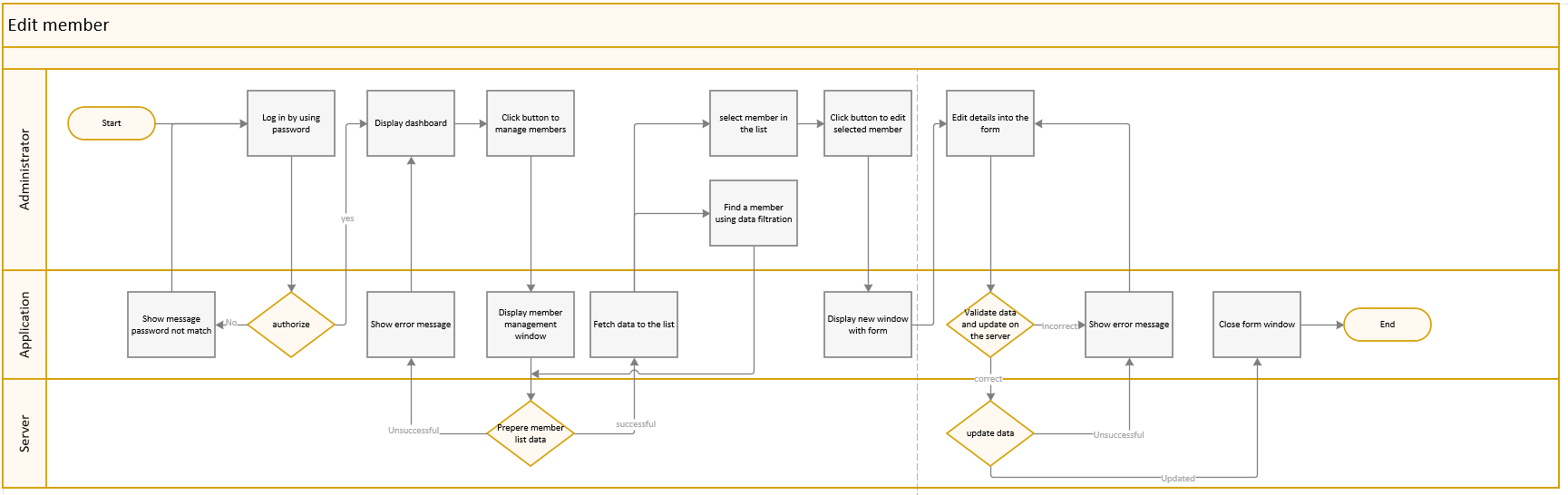
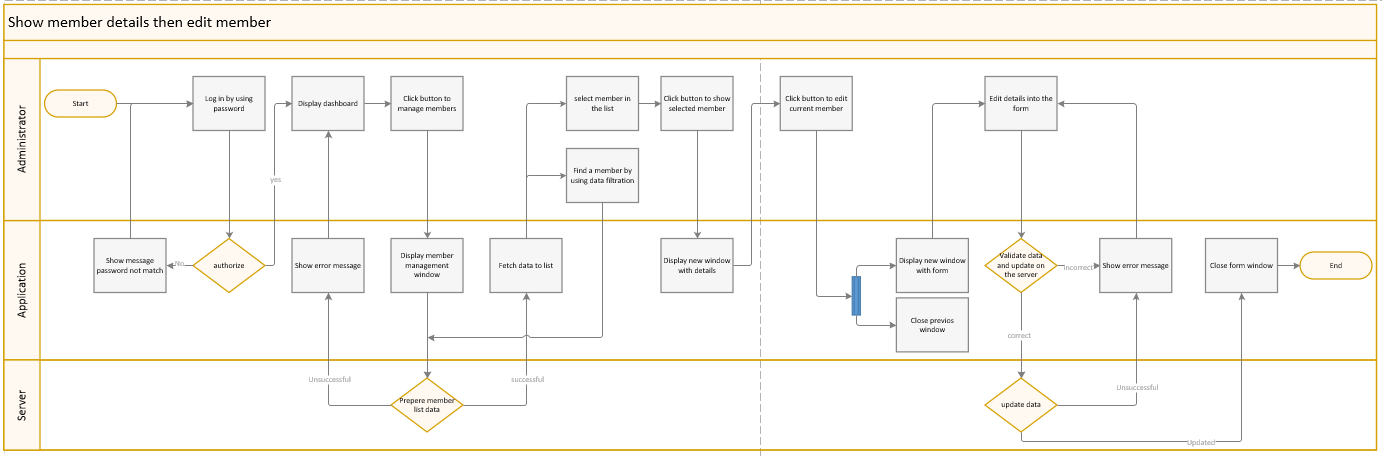
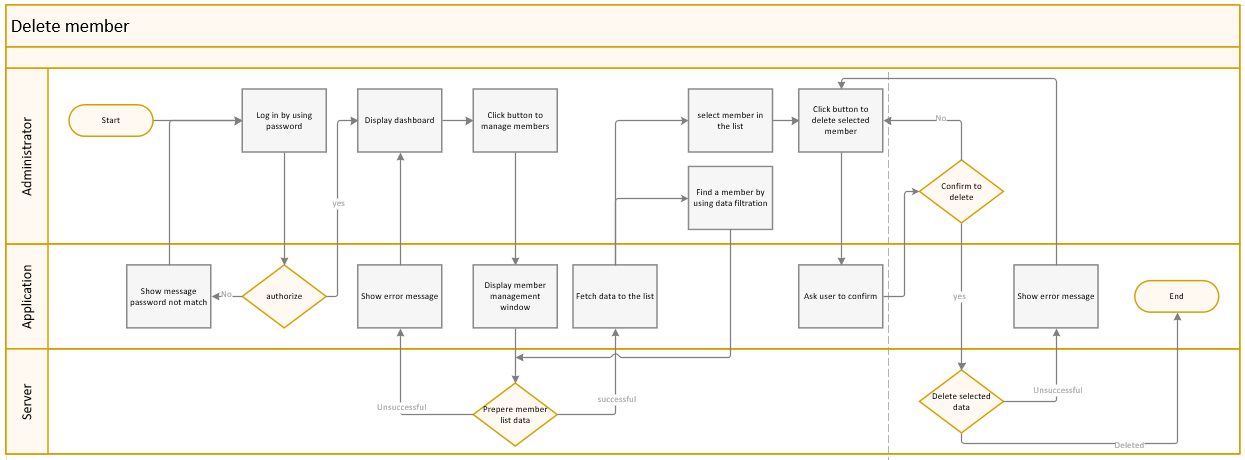
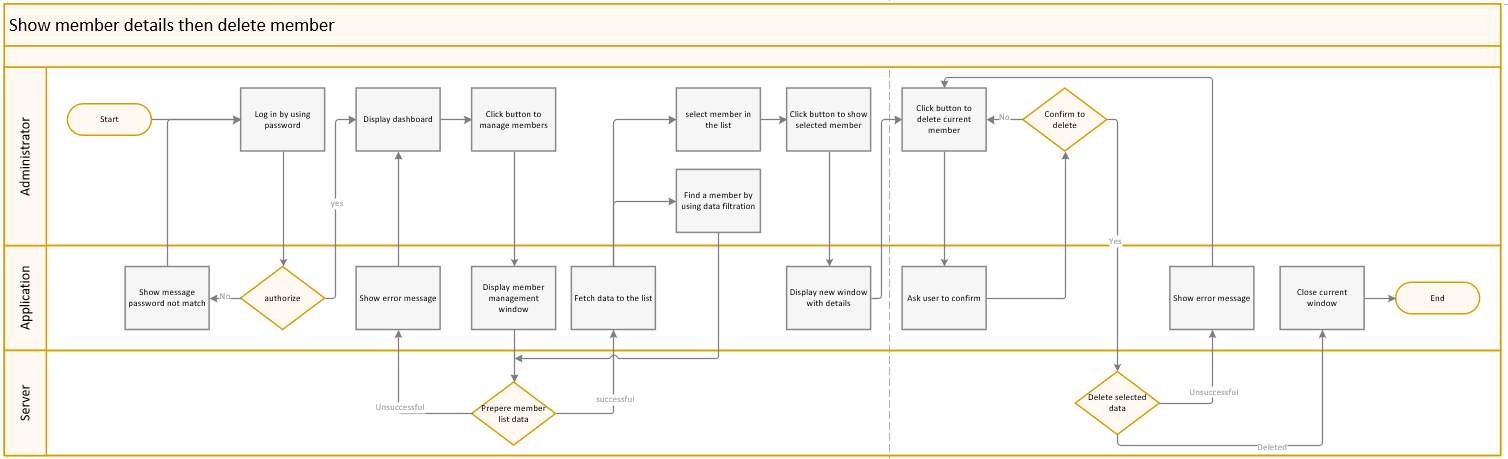
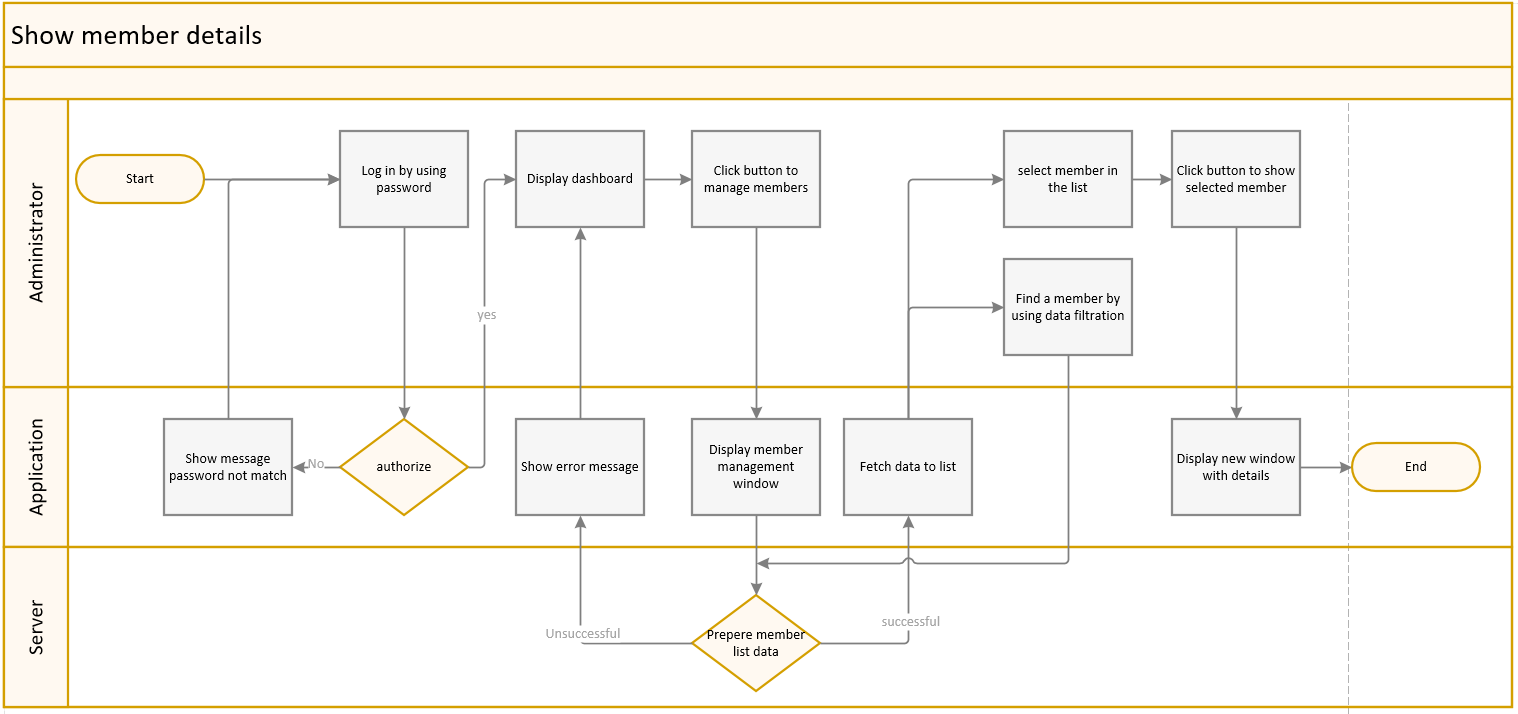
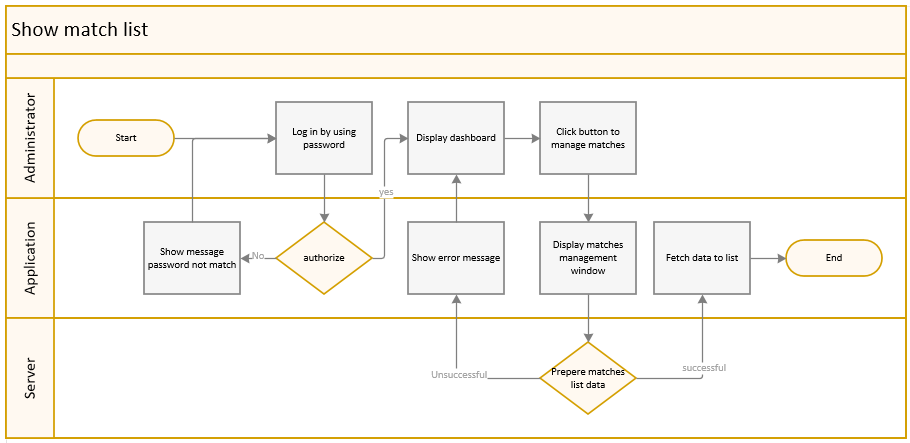
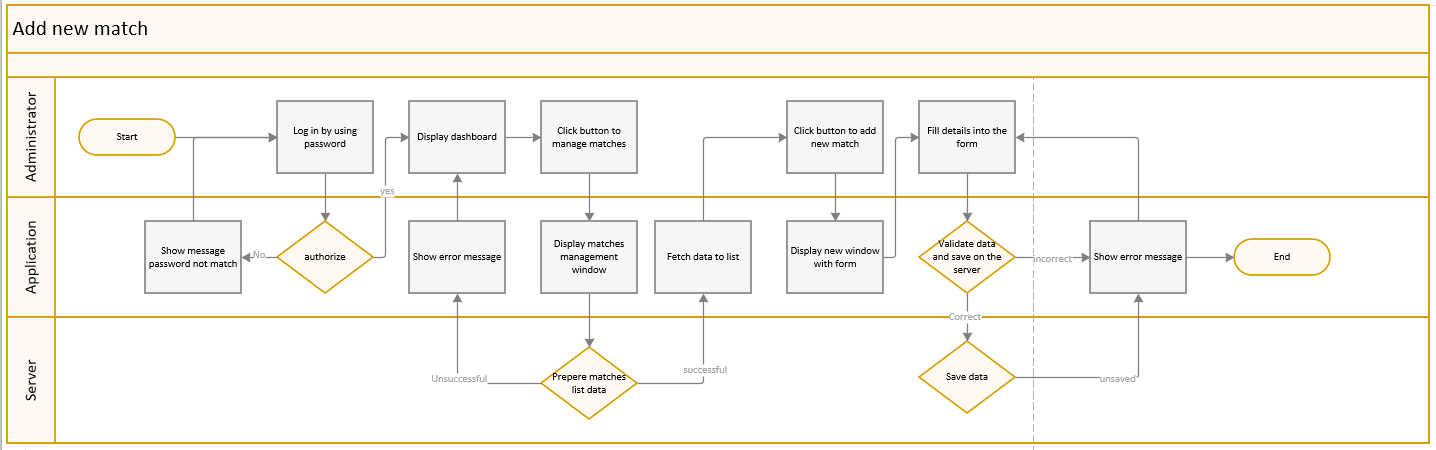
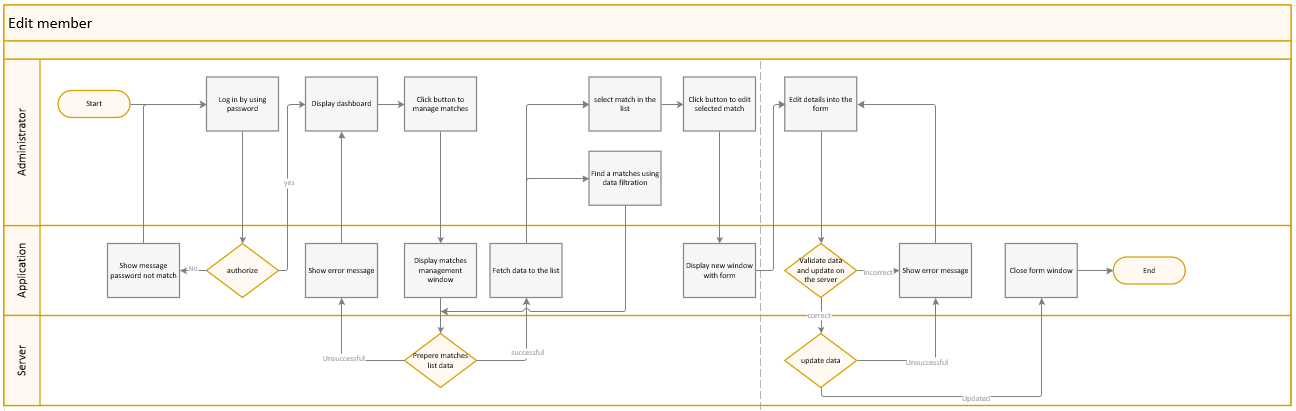
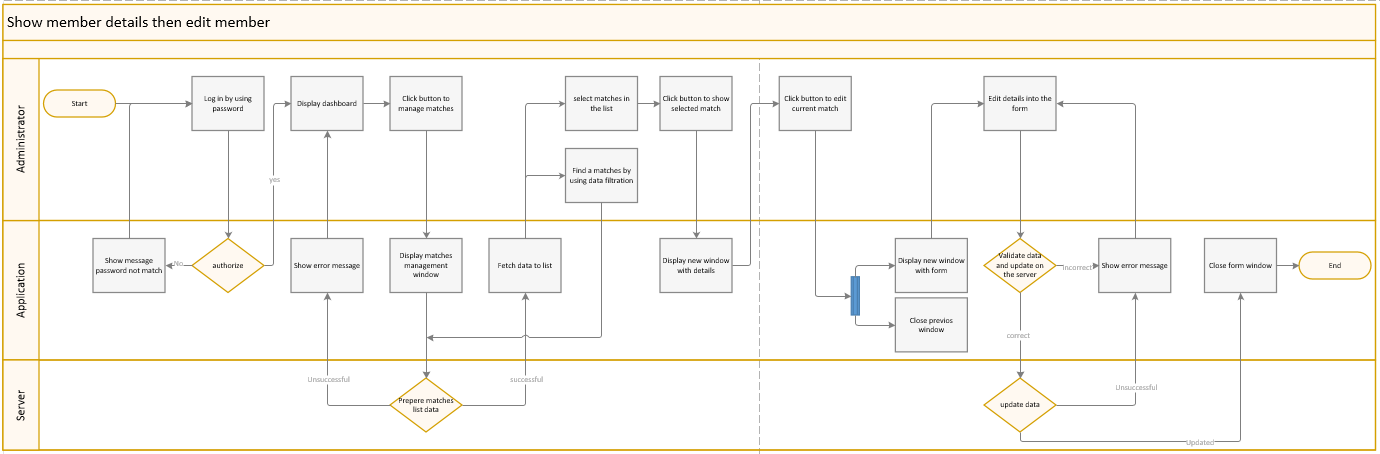
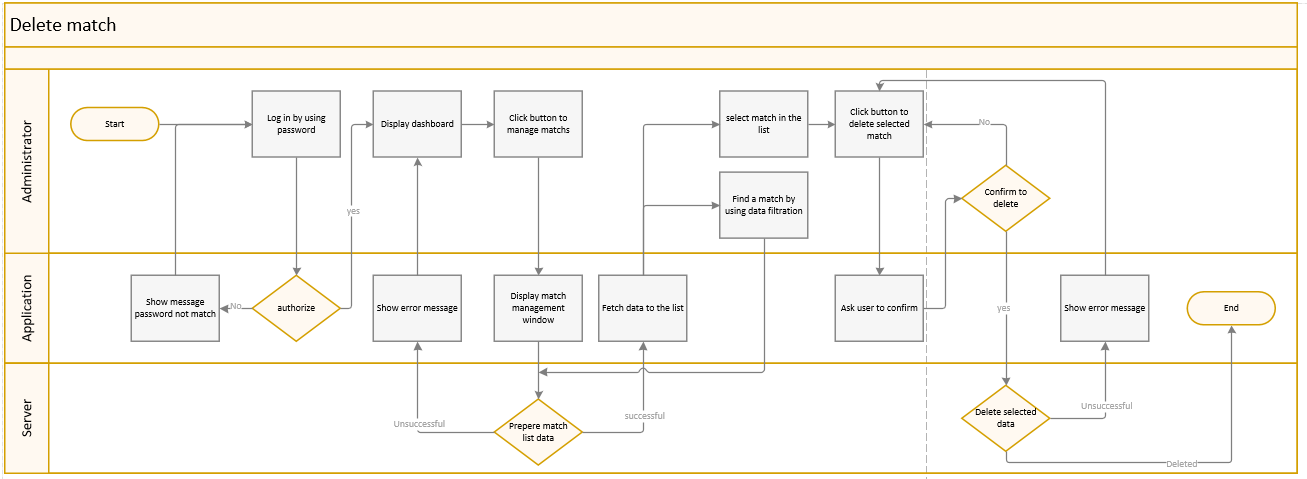
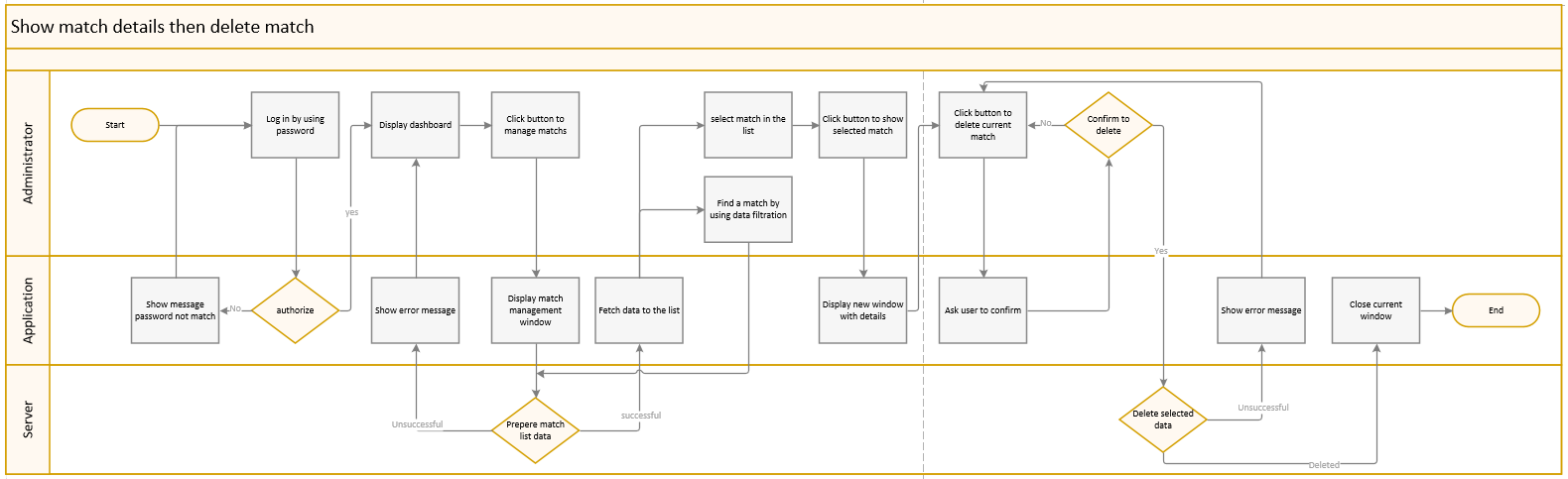
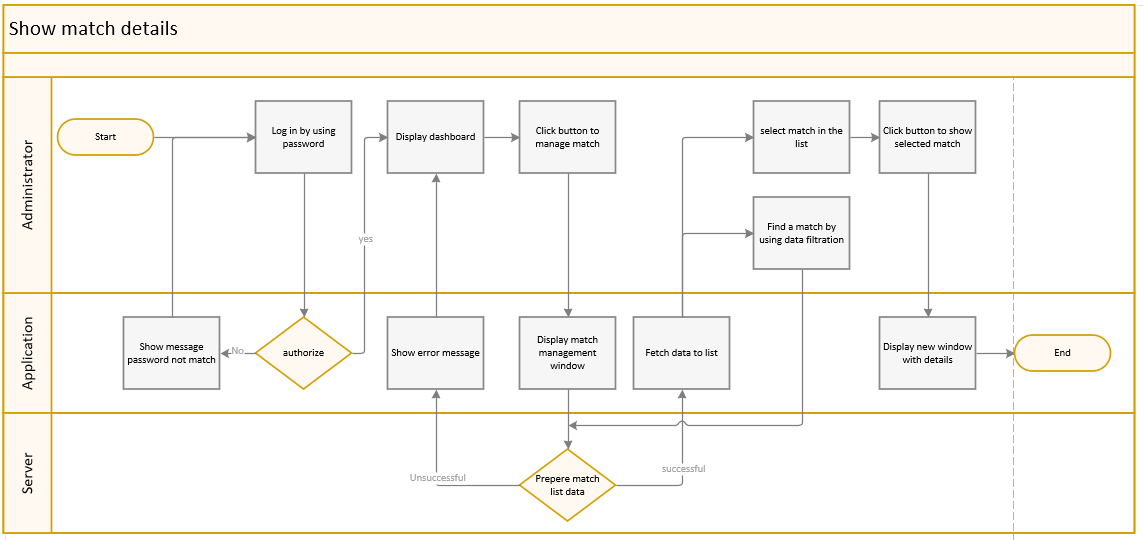
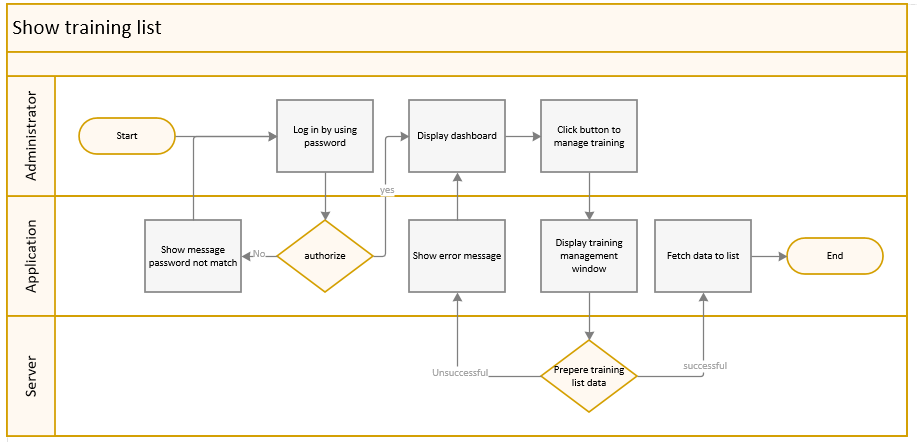
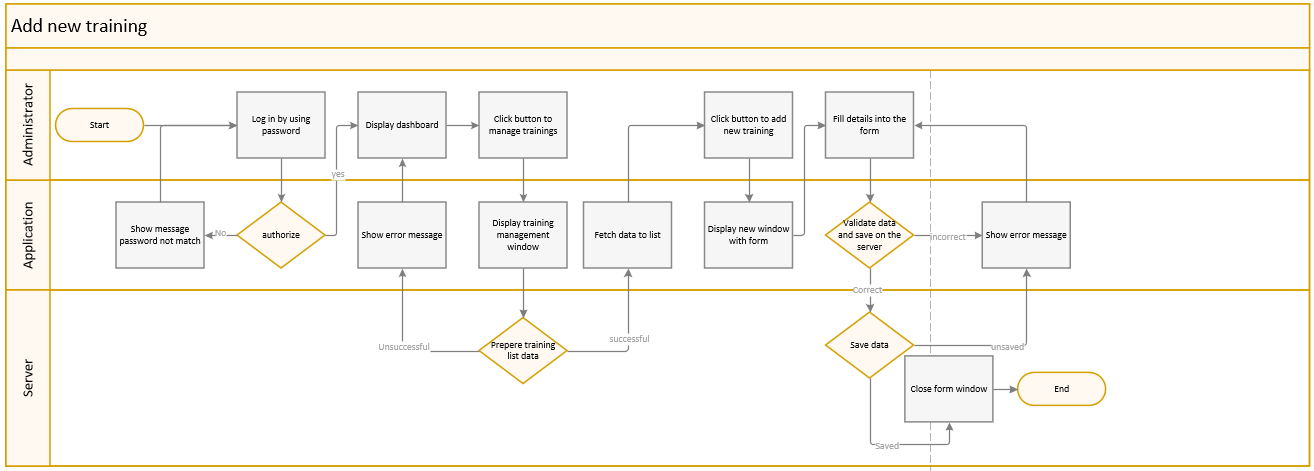
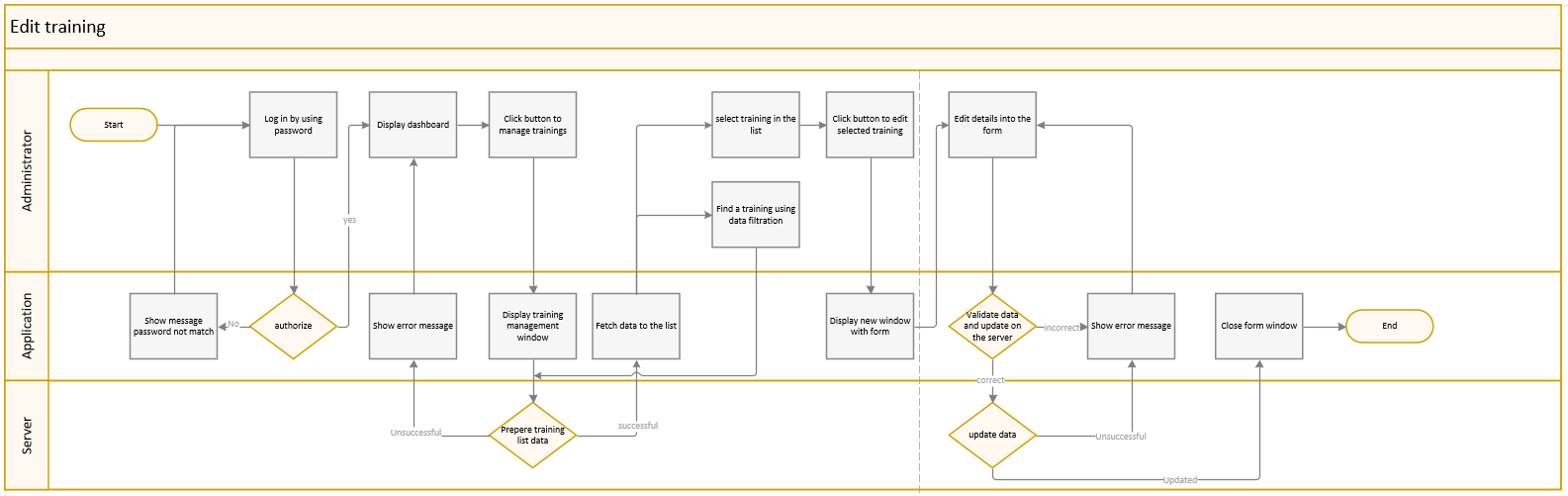
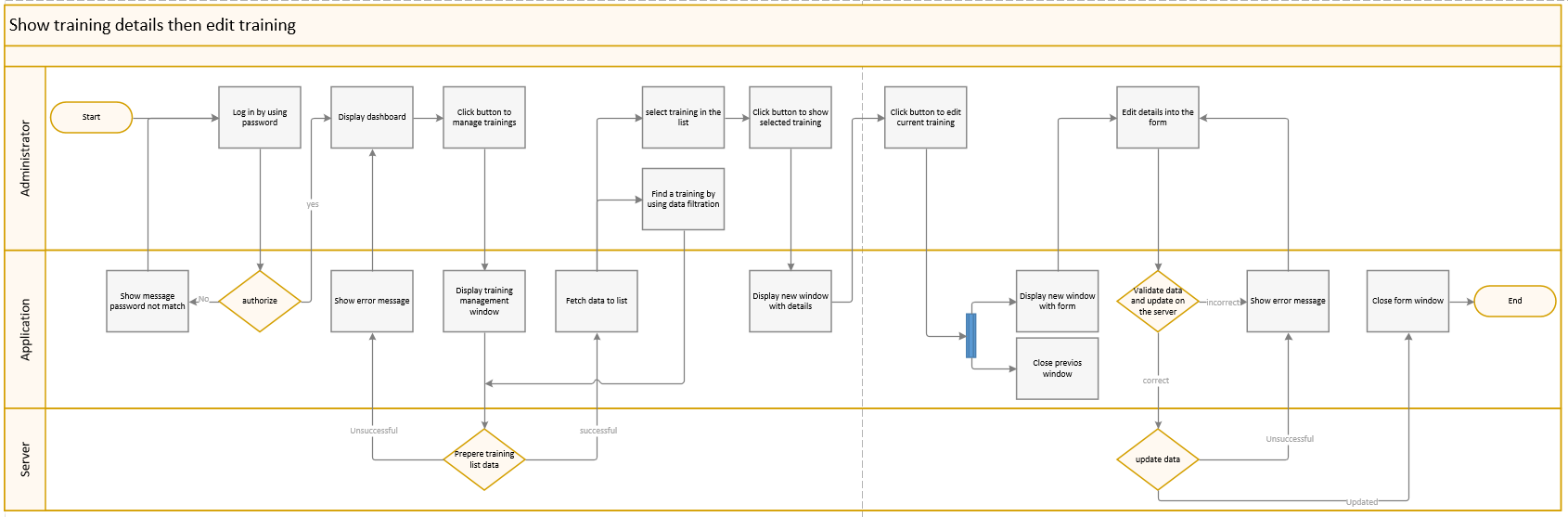
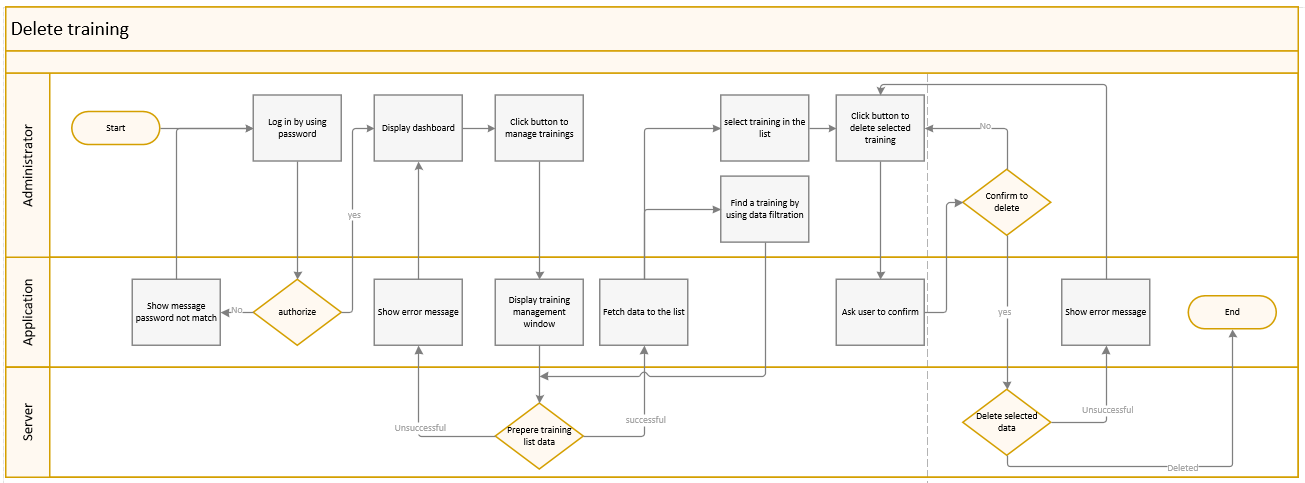
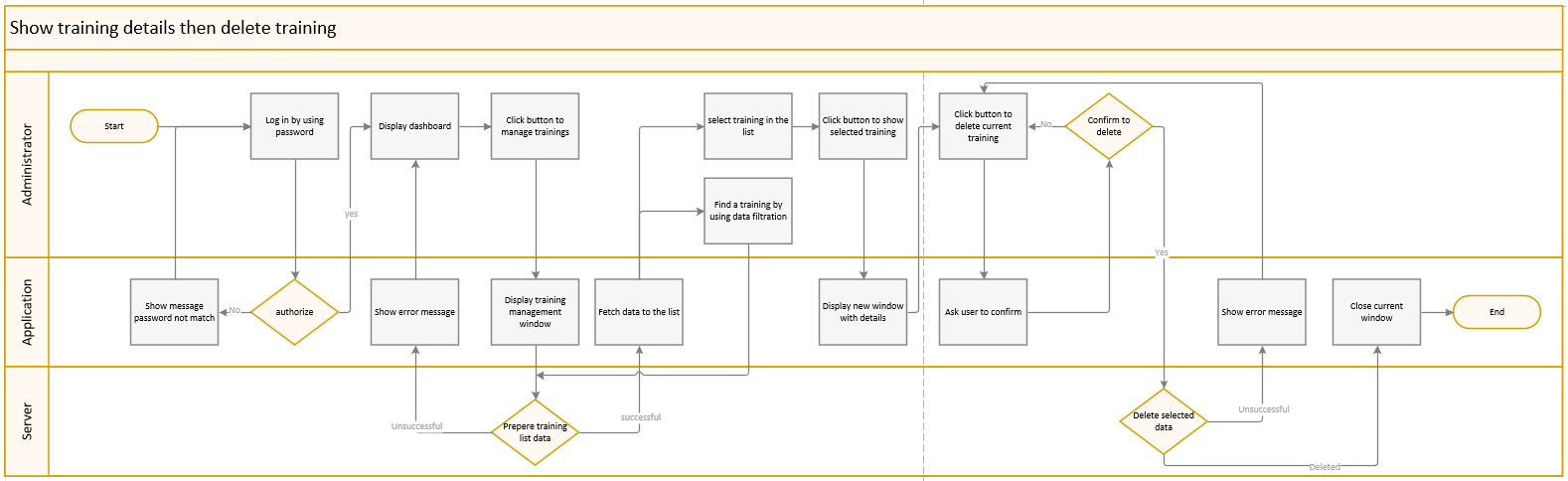
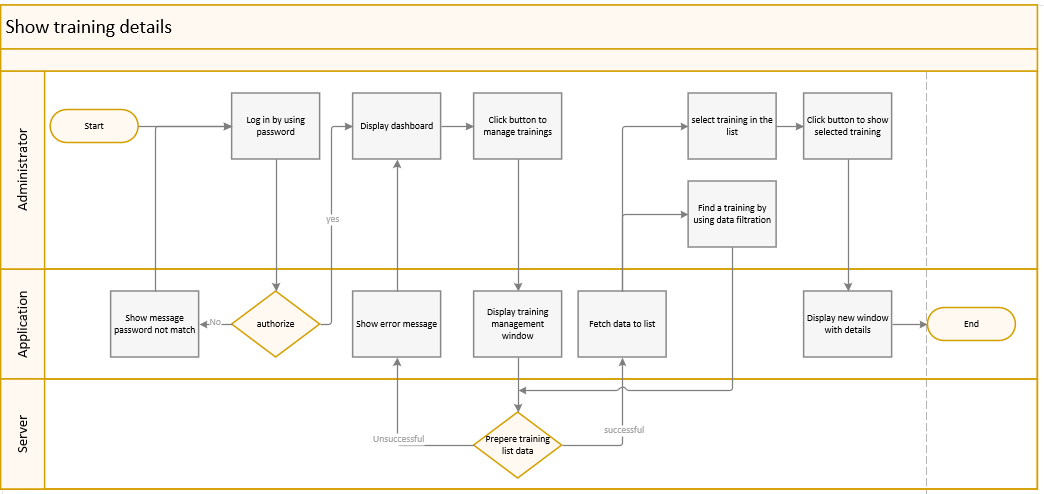
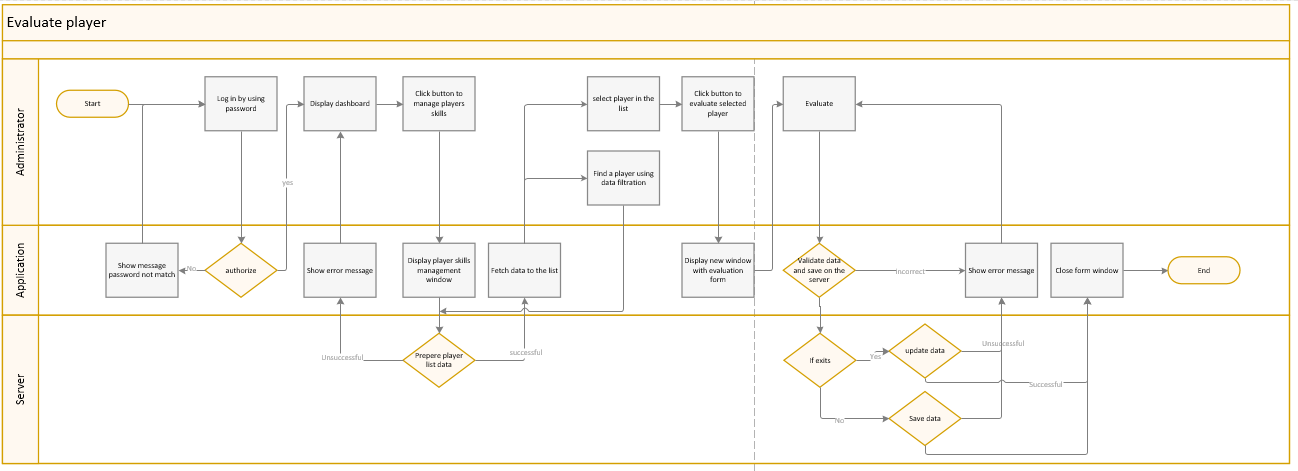
The diagram shows the binding of data for each of the activities.



* + 1. Activity Diagram

The diagram shows the course of the process for each activity and show how it deal with Administrator, Application and Server



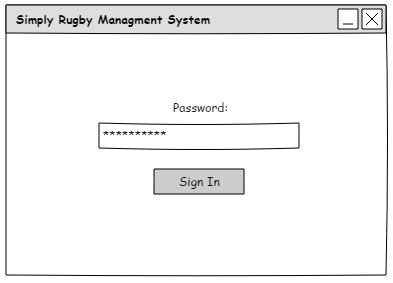
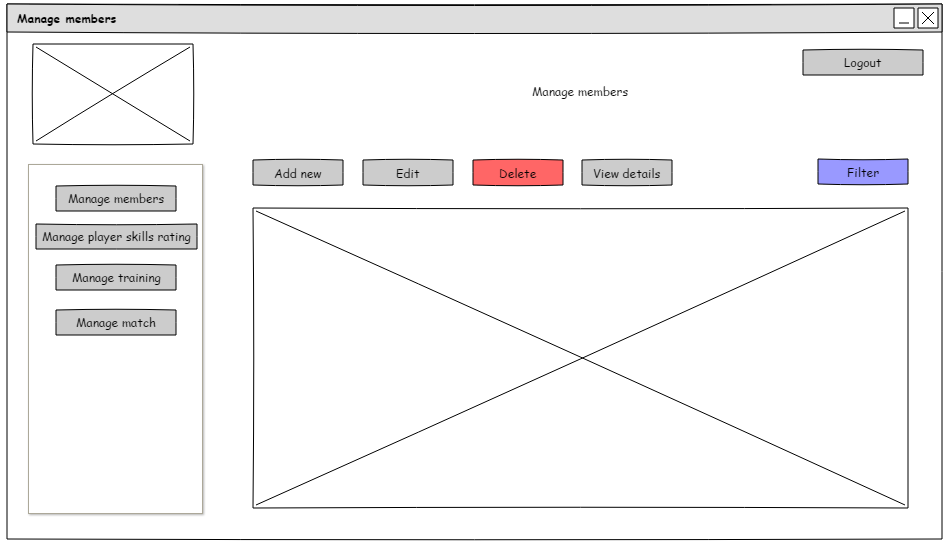
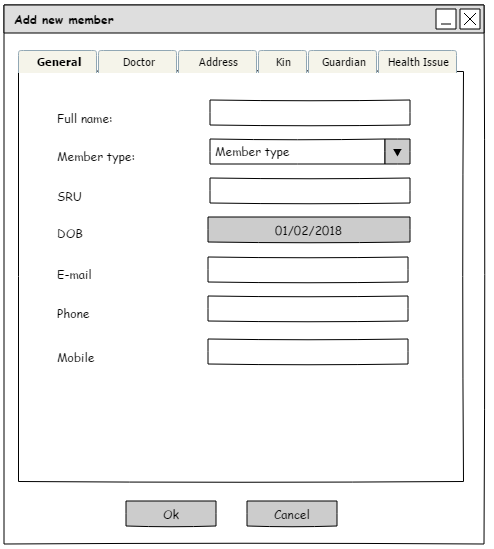
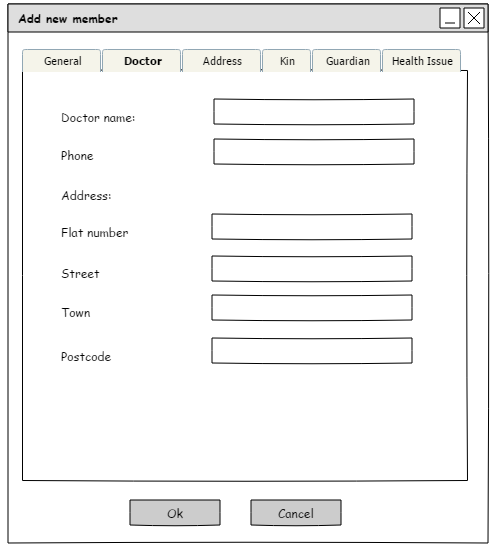
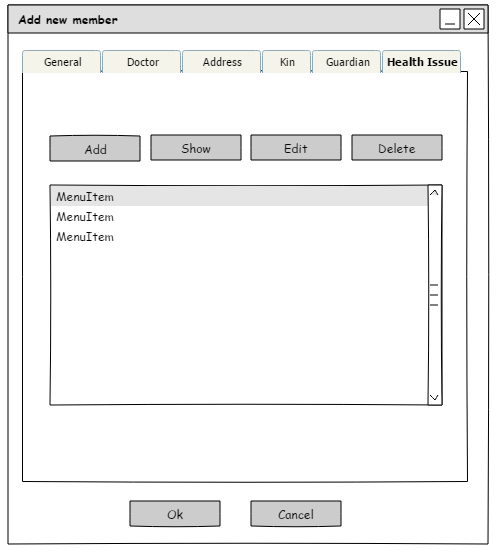
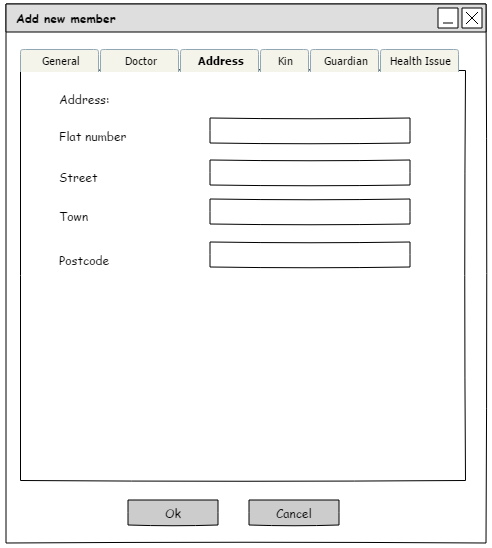
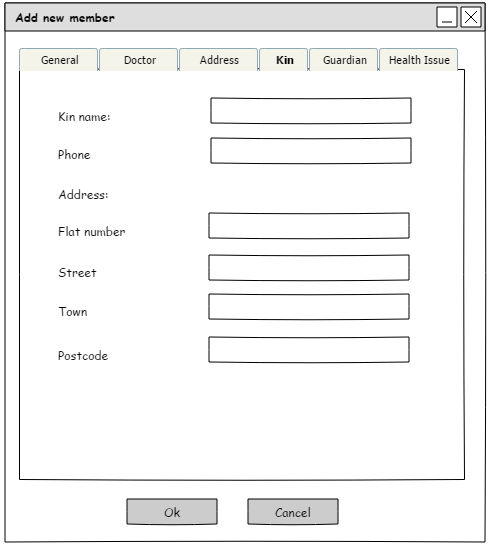
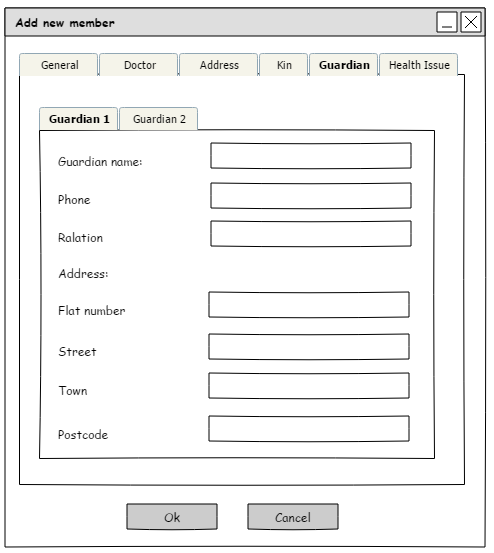
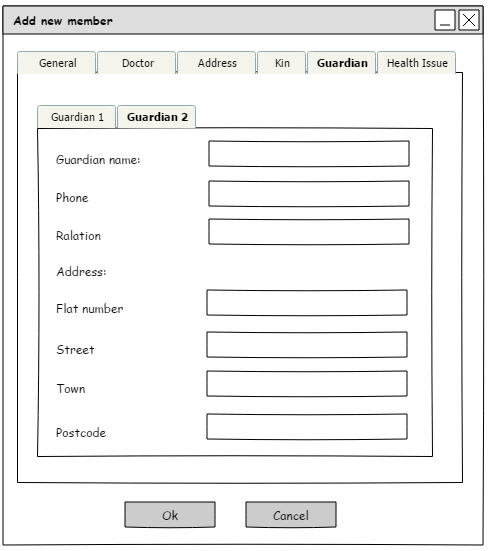
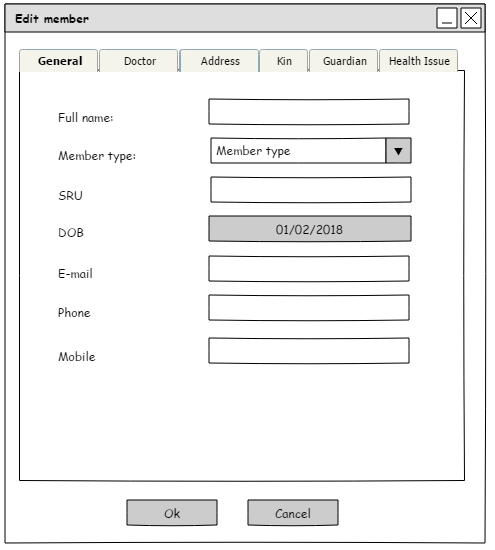
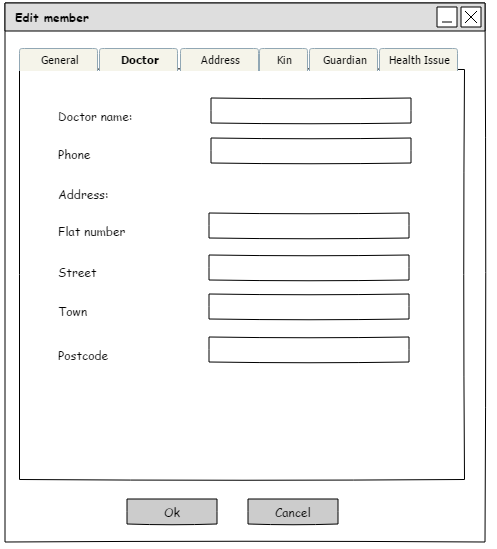
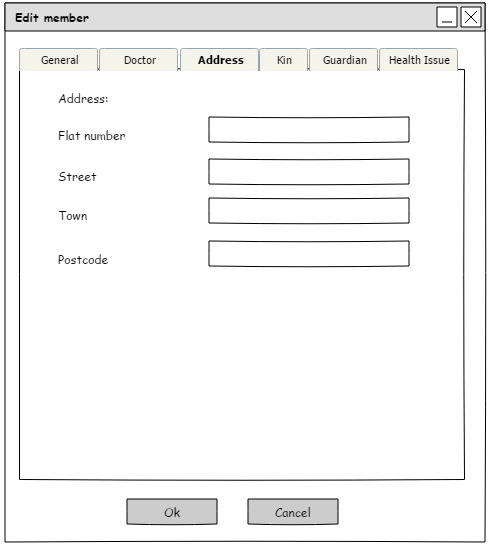
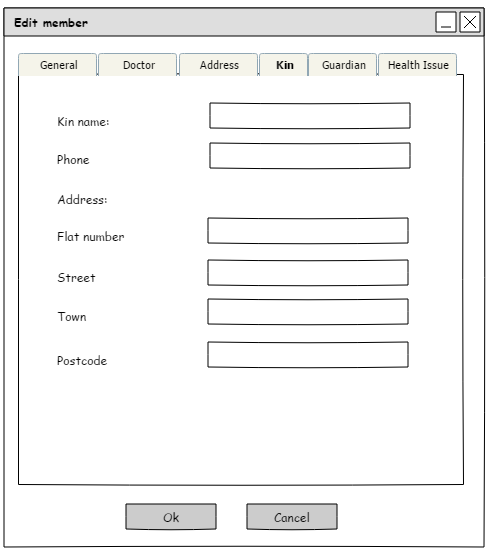
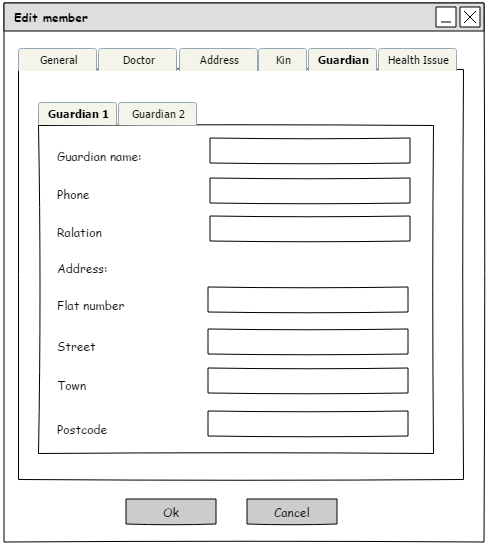
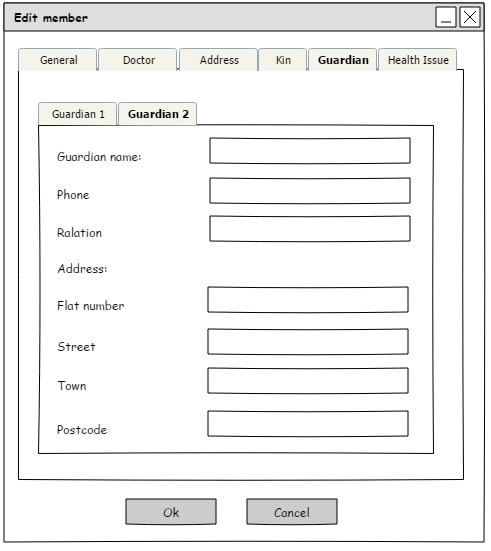
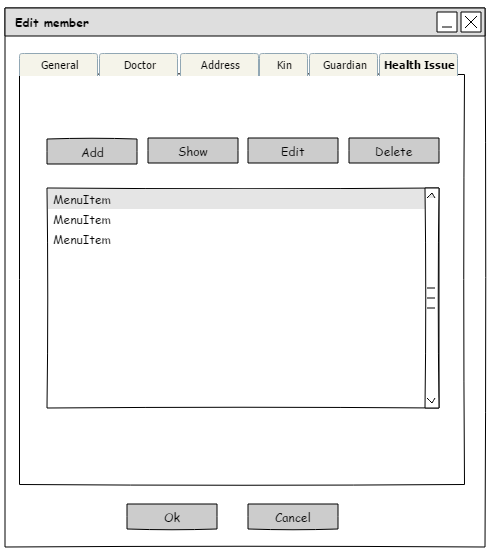
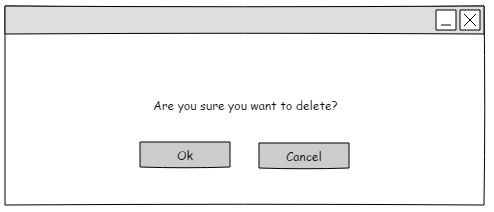
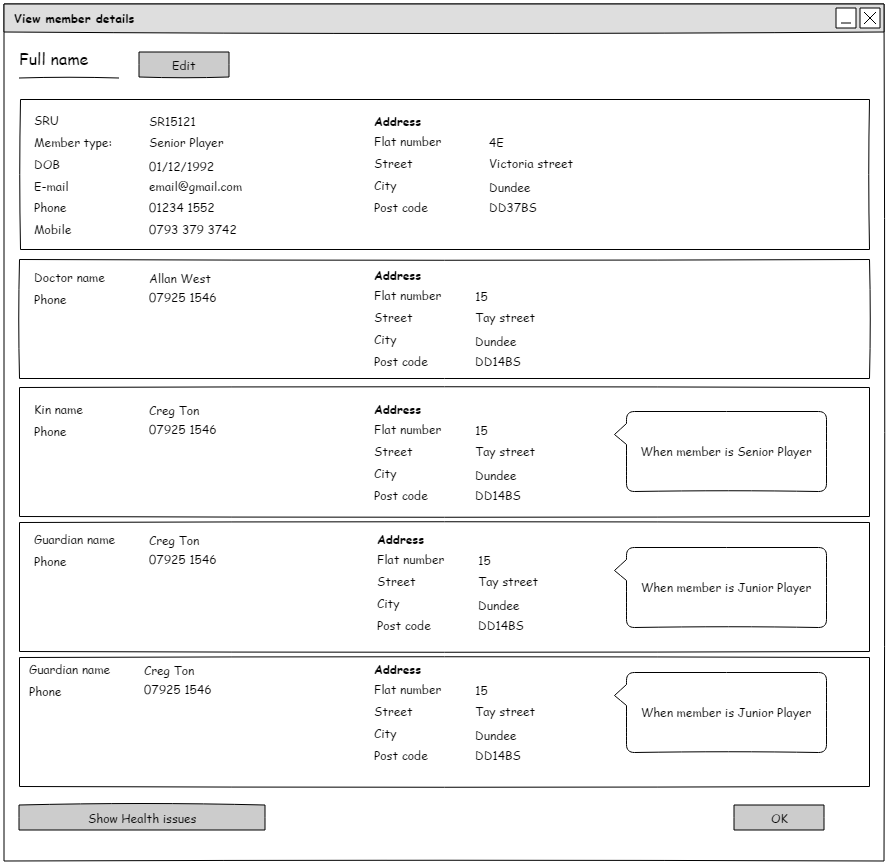
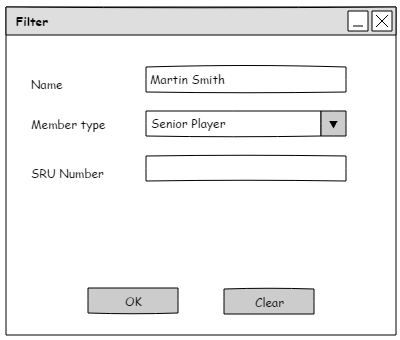
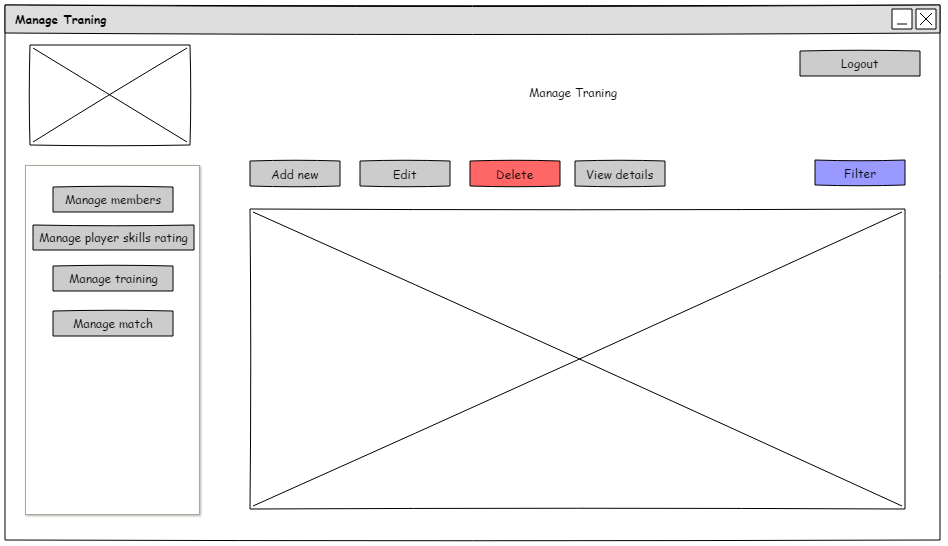
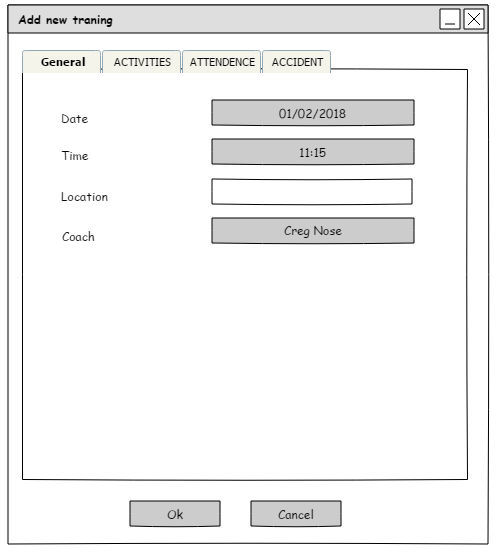
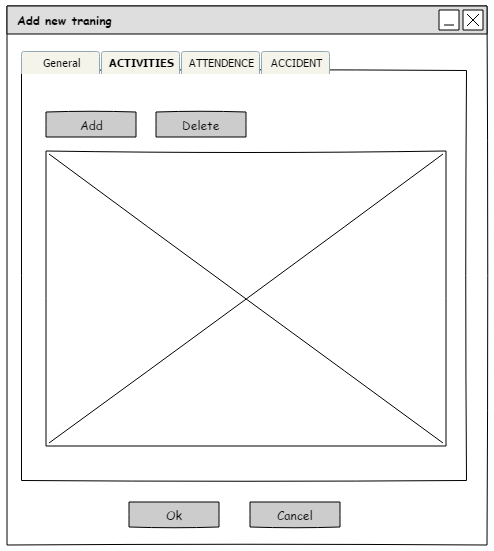
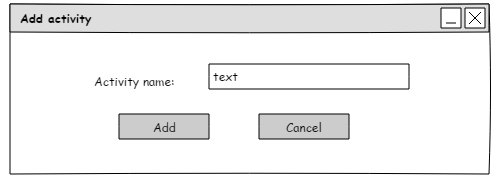
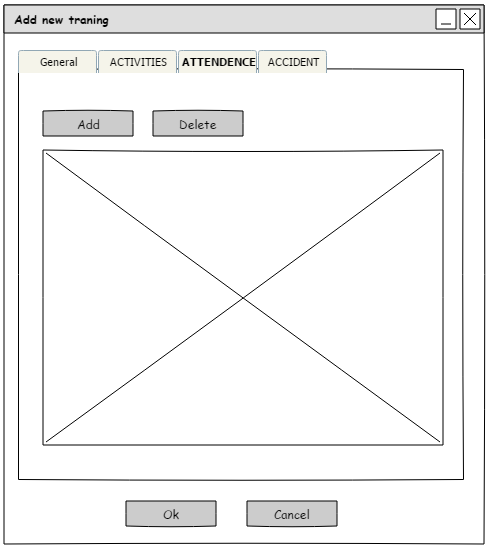
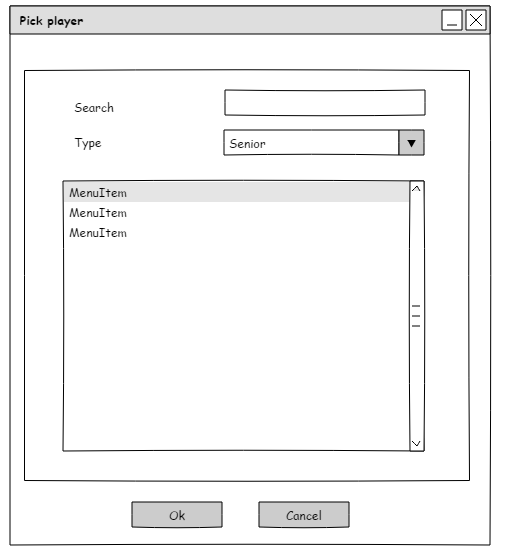
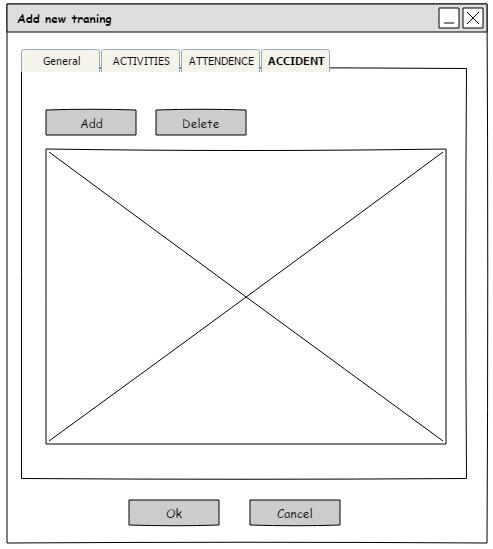
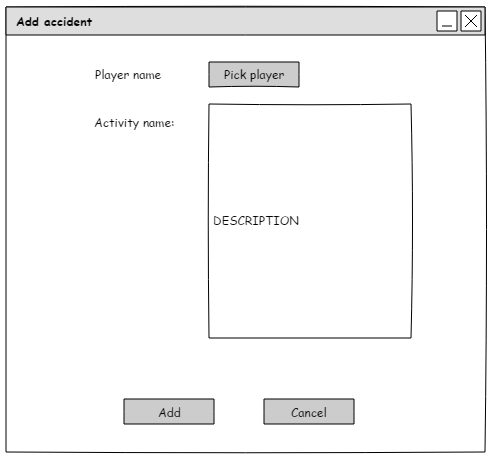
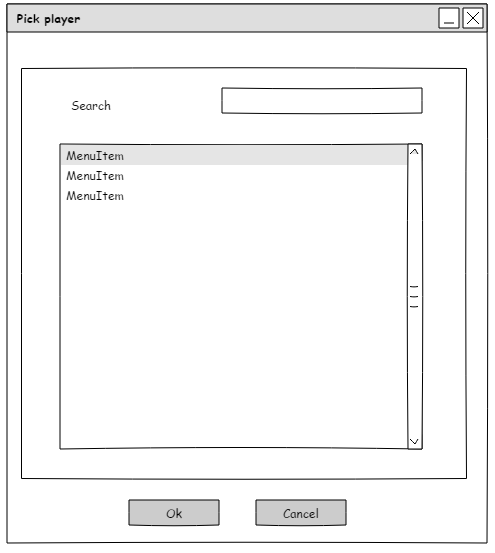
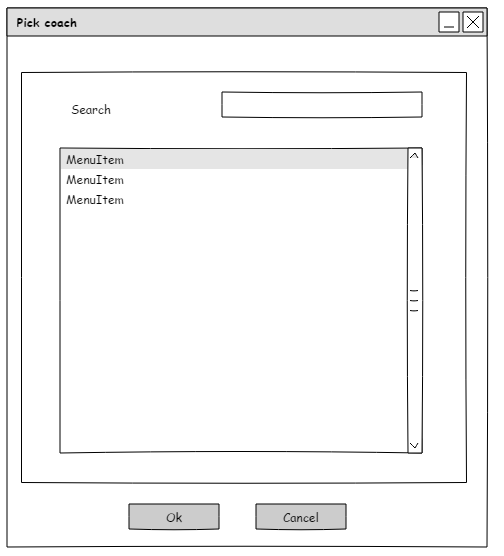
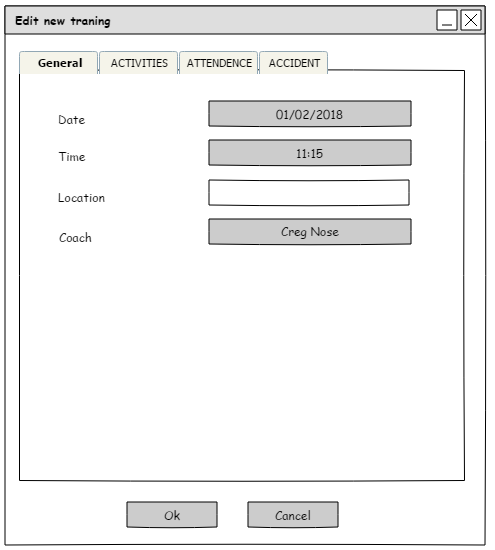
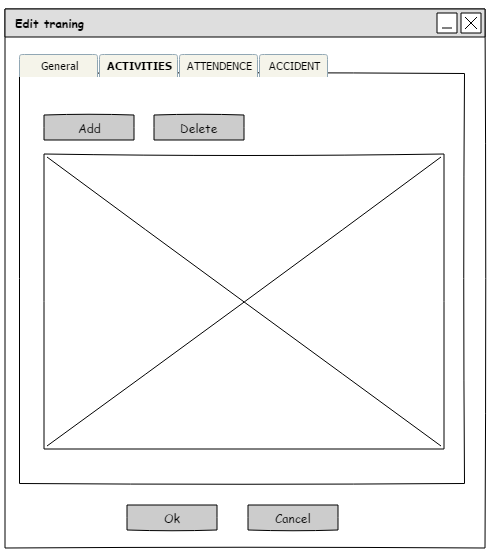
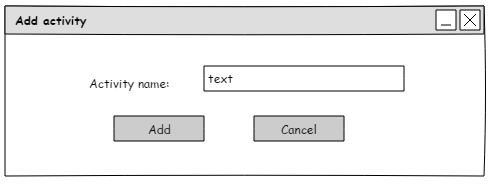
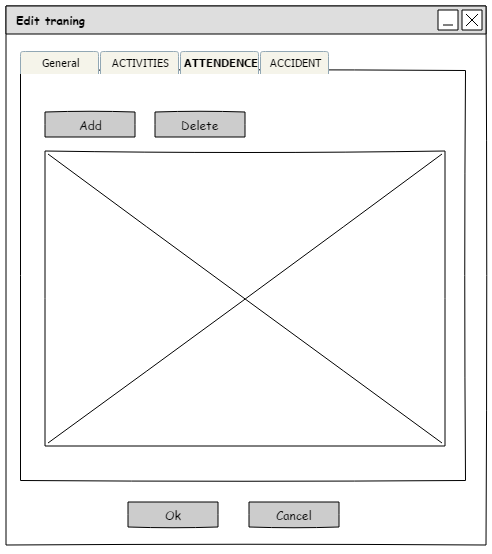
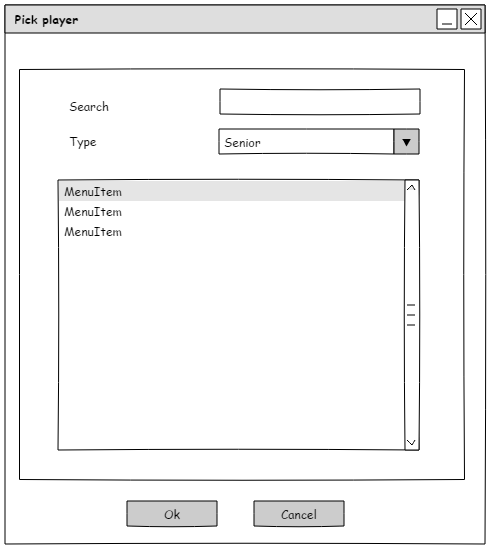
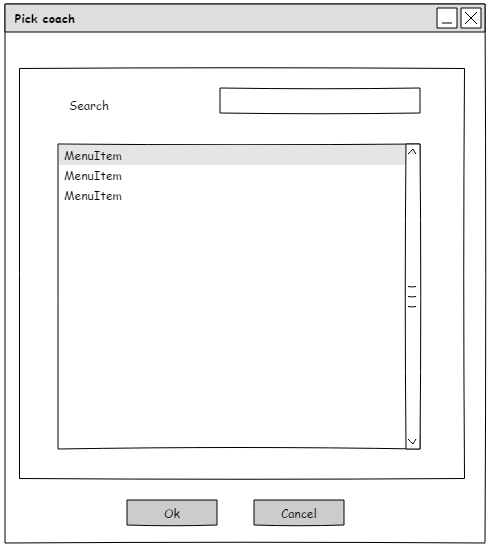
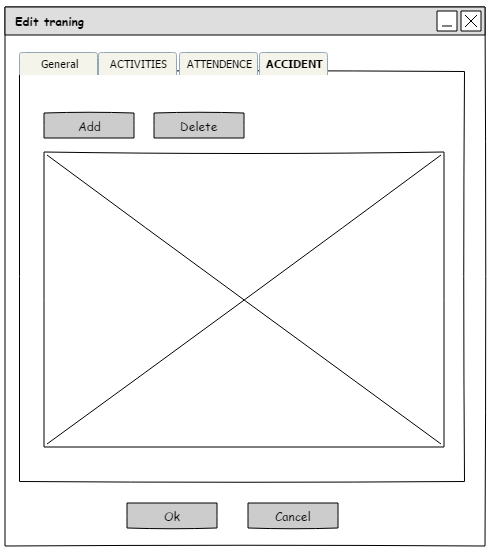
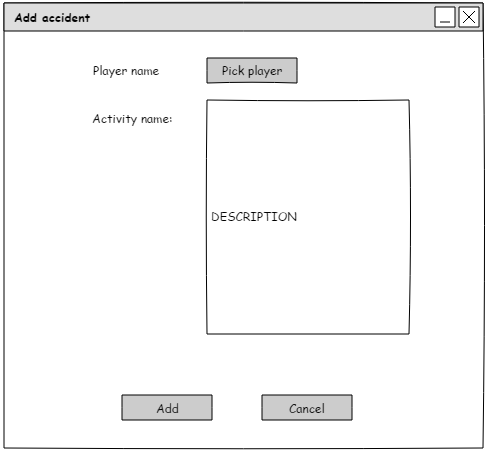
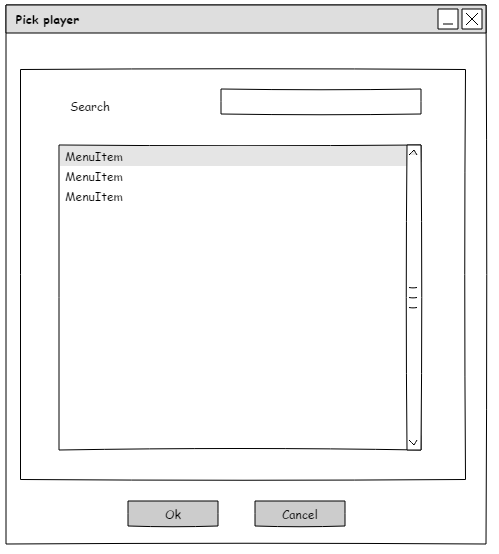
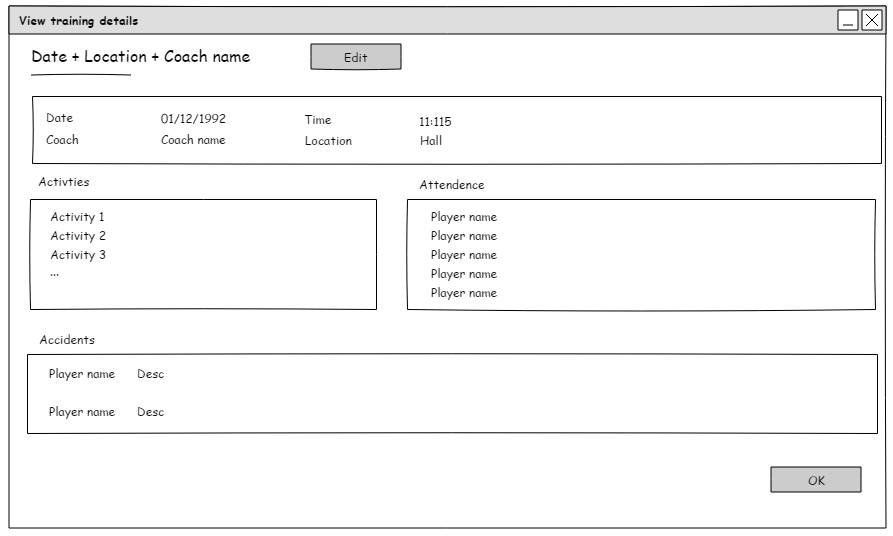
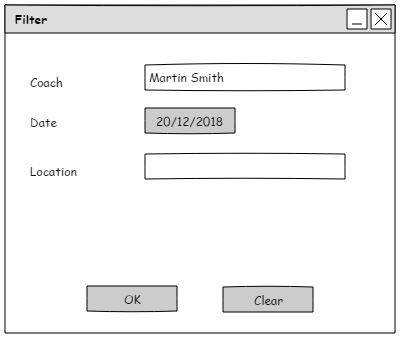
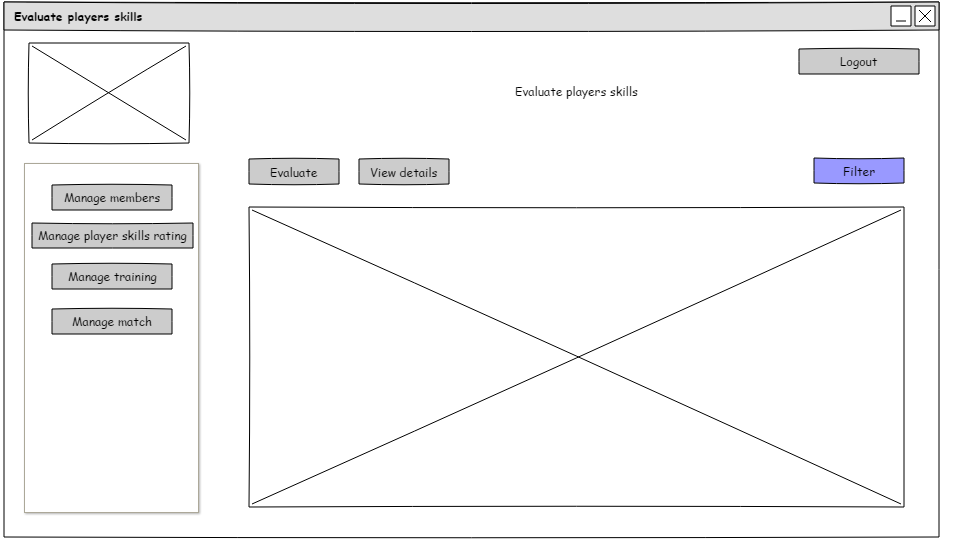
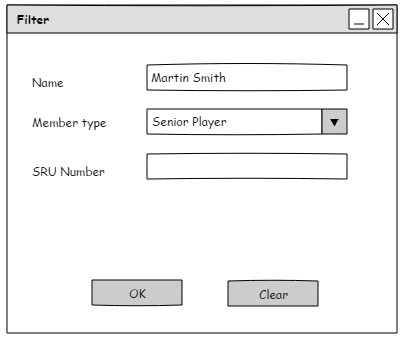
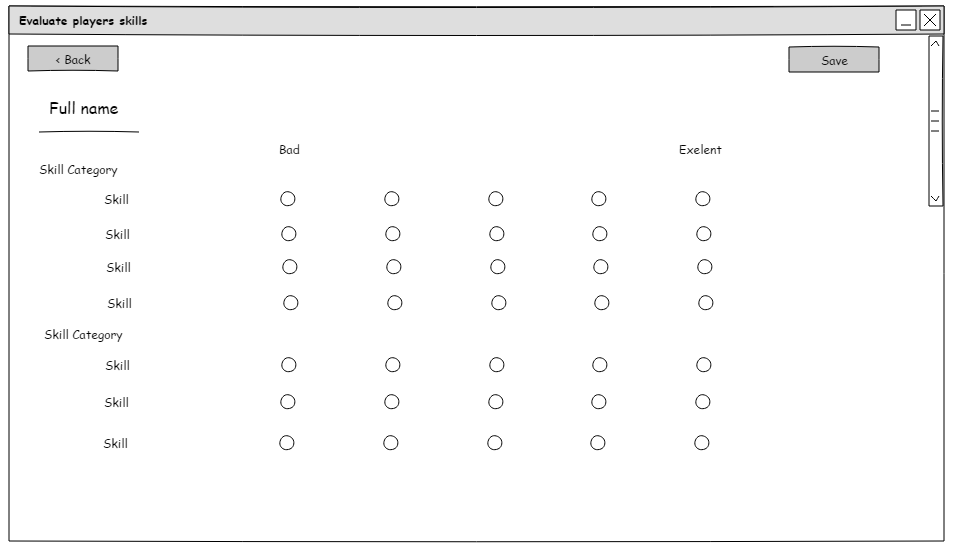
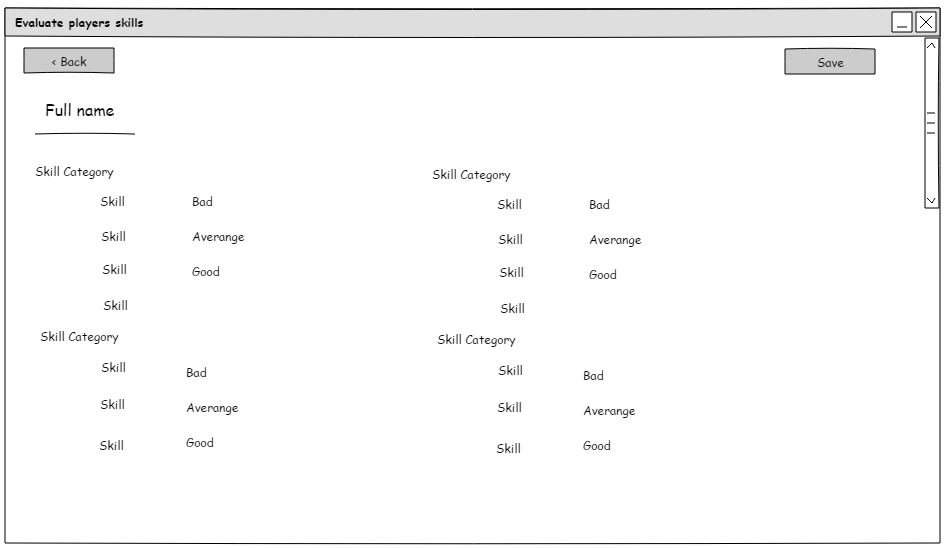
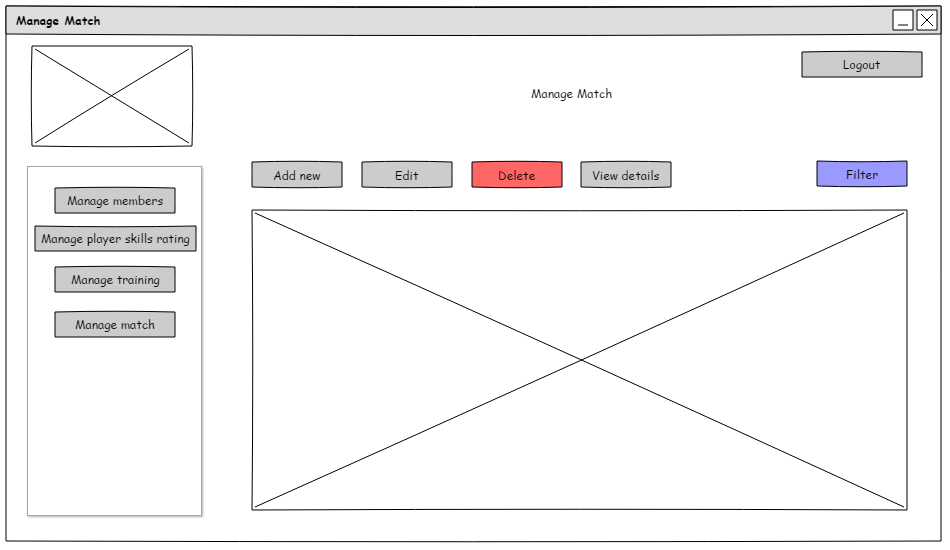
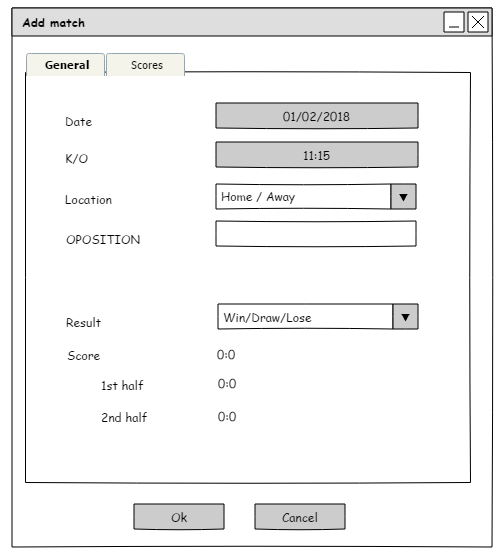
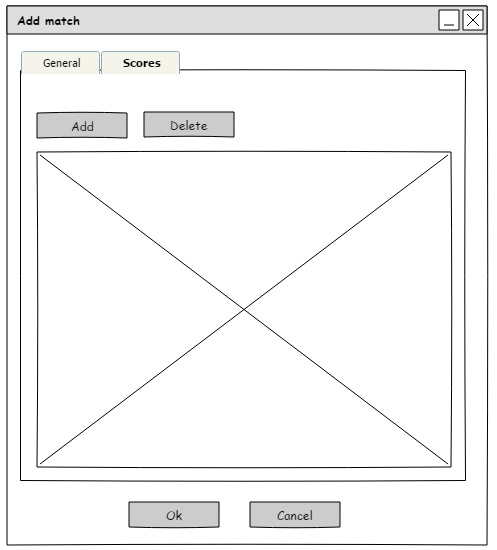
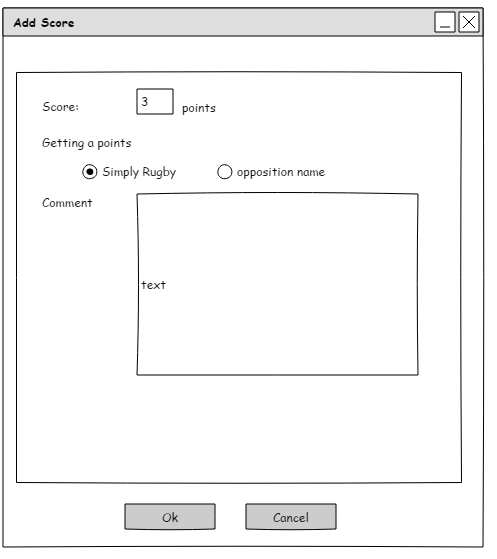
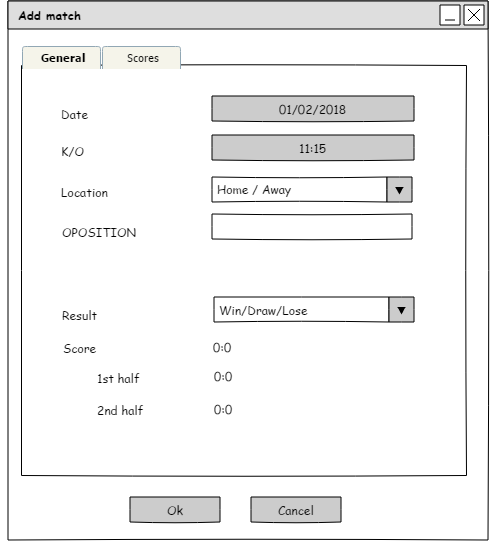
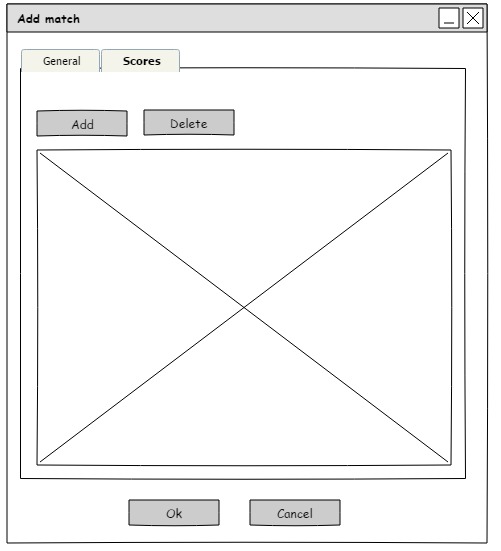
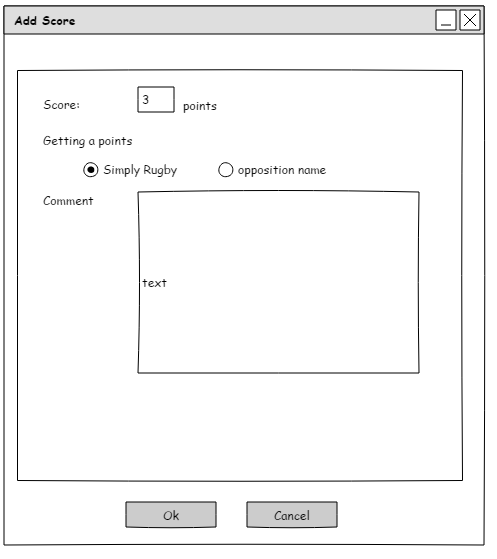
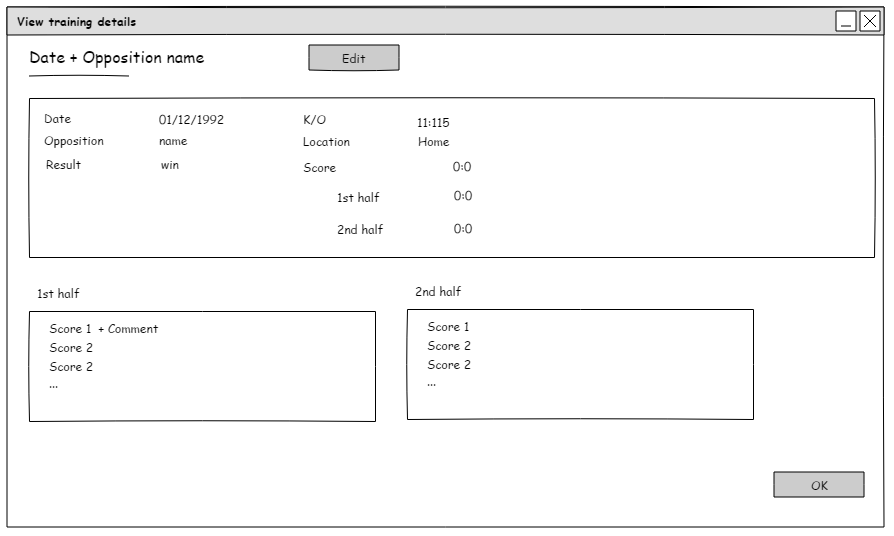
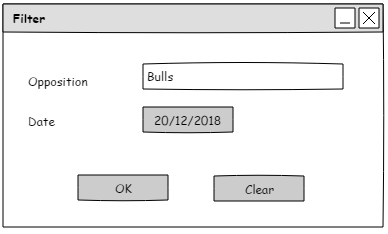
                   

1. View Model

This section presents an example graphic design of the application. The project assumes designing a minimalist user interface so that the data entered is not excessive, and the forms do not overwhelm the user with an excessive number of text fields in one window. The GUI must be easy to use and legible while providing the most important information.

* 1. Wireframe

A list of examples of window designs, presenting the arrangement of functionality in individual application windows. Some windows will be nested in other windows giving a consistent appearance of the application.

* + 1. Home page  
       
    2. Dashboard
    3. Manage members
    4. Add member  
       
    5. Add member: Doctor  
       
    6. Add member: Health Issues  
       
    7. Add member: Address  
       
    8. Add member: Kin  
       
    9. Add member: Guardian  
       
    10. Add member: Guardian (1)  
        
    11. Edit Member  
        
    12. Edit member: Doctor  
        
    13. Edit member: Address  
        
    14. Edit member: Kin  
        
    15. Edit member: Guardian  
        
    16. Edit member: Guardian (1)  
        
    17. Edit member: Health Issues  
        
    18. Member Delete Show Box  
        
    19. View Member  
        
    20. View Member: Health Issues  
        
    21. Filter Members  
        
    22. Manage training  
        
    23. Add Training  
        
    24. Add training activities  
        
    25. Add training: add activity  
        
    26. Add training: attendance  
        
    27. Add training: pick player  
        
    28. Add training: accident  
        
    29. Add training: add accident  
        
    30. Add accident: pick player  
        
    31. Add training pick coach  
        
    32. Edit Training  
        
    33. Edit training activities  
        
    34. Edit training: add activity  
        
    35. Edit training attendance  
        
    36. Edit training: pick player  
        
    37. Edit training: pick coach  
        
    38. Edit training: accident  
        
    39. Edit training: add accident  
        
    40. Edit accident: pick player  
        
    41. Training view details  
        
    42. Training filter  
        
    43. Manage player’s skills  
        
    44. Filter Members Evaluate  
        
    45. Evaluation  
        
    46. Evaluation Show  
        
    47. Manage match  
        
    48. Add match  
        
    49. Add match: Scores  
        
    50. Add: Scores: add  
        
    51. Edit match  
        
    52. Edit match: Scores  
        
    53. Edit: Scores: add  
        
    54. View match  
        
    55. Match filter  
        

1. References

Coacha. (2018, 09). *Classes and Automatically Generated Registers*. Retrieved from Coacha: https://www.coacha.co.uk/Features/Classes-and-Sessions

IEEE. (2018, 09 12). *Practice for Software Requirements*. Retrieved from Computer science and engineering: http://www.cse.msu.edu/~cse870/IEEEXplore-SRS-template.pdf

Rugby, S. (2018, 09 12). Retrieved from Official site of Scottish Rugby: http://www.scottishrugby.org/

SportEasy. (2018, 09). *Features*. Retrieved from SportEasy: https://www.sporteasy.net/en/presentation/