# 3. Introduction

# Design

The Design phase describes how the system fulfils the user requirements. For the achievement of user requirement, we need create both logical and physical design. In this phase the systems design functions and operations are described in detail, including storyboard and screen layouts with annotations, business rules, process diagrams and other documentation.

For Health and Fitness Management System I have created Structural and Behavior Modelling diagram in the following way.

# 3.1 Structural Modelling

“A structure diagram is a conceptual modeling tool used to document the different structures that make up a system such as a database or an application.”

For Health and Fitness Management System, in the part of Structural Model, I have created Context Diagram and Class Diagram.

# 3.1.1 Class Diagram

# 3.2 Behavior Modelling

# 3.2.1 Activity Diagram

Activity Diagram is important behavioral diagram which is used to describe dynamic aspects of the system. For Health and Fitness Management System I have created the diagrams.

Notations used in Activity Diagram

Notation DescriptionUML Notation

**Activity**

Is used to represent a set of actions



**Action**

A task to be performed

Activity Diagram Notation - Action

**Control Flow**

Shows the sequence of execution

Activity Diagram Notation - Control Flow

**Object Flow**

Show the flow of an object from one activity (or action) to another activity (or action).

Activity Diagram Notation - Object Flow

**Initial Node**

Portrays the beginning of a set of actions or activities

Activity Diagram Notation - Initial Node

**Activity Final Node**

Stop all control flows and object flows in an activity (or action)

Activity Diagram Notation - Activity Final Node

**Object Node**

Represent an object that is connected to a set of Object Flows

Activity Diagram Notation - Object Node

**Decision Node**

Represent a test condition to ensure that the control flow or object flow only goes down one path

Activity Diagram Notation - Decision Node

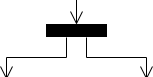
**Merge Node**

Bring back together different decision paths that were created using a decision-node.

Activity Diagram Notation - Merge Node

**Fork Node**

Split behavior into a set of parallel or concurrent flows of activities (or actions)



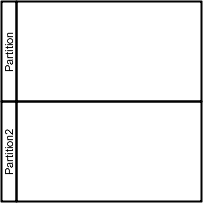
**Join Node**

Bring back together a set of parallel or concurrent flows of activities (or actions).

Activity Diagram Notation - Join Node

**Swimlane and Partition**

A way to group activities performed by the same actor on an activity diagram or to group activities in a single thread



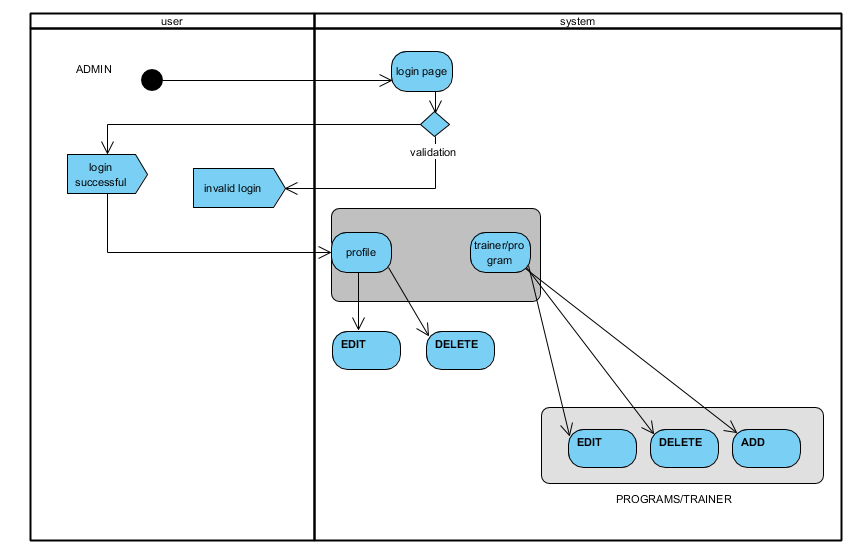


Fig: Admin login Activity Diagram

This shows the all the functions and the activity of the admin of the system. Admin login Activity Diagram shows the activity sequence of the admin in the system.

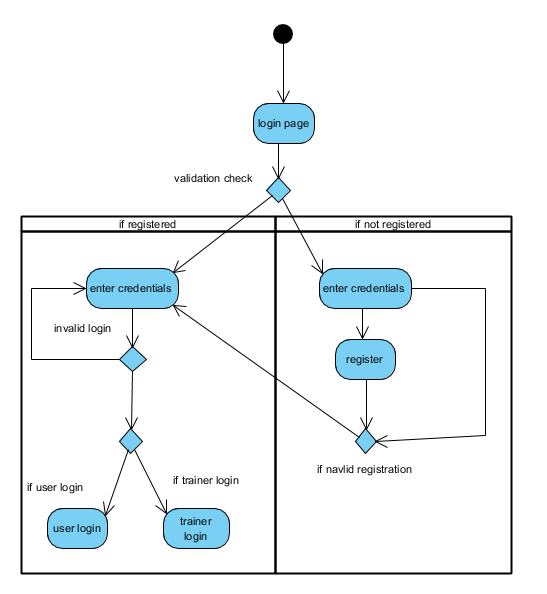


Fig: login activity diagram

This shows the overall activities involved during the login phase of the system in the Health and Fitness Management System.

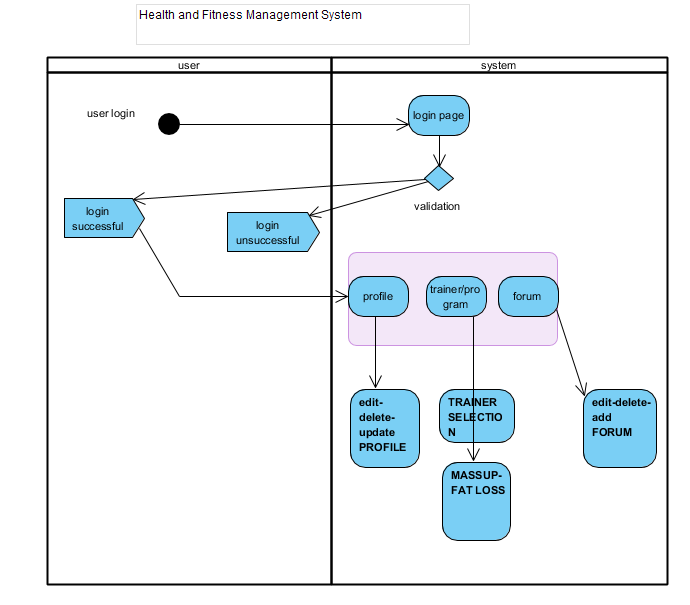


Fig: System Activity Diagram

System Activity Diagram show the overall functions of the Health and Fitness Management System.

# 3.2.2 Sequence Diagram

Sequence Diagrams are interaction diagrams that detail how operations are carried out. They capture the interaction between objects in the context of a collaboration.

Notations used in Sequence Diagram

Notation DescriptionVisual Representation

**Actor**

* An actor does not necessarily represent a specific physical entity but merely a particular role of some entity
* A person may play the role of several different actors and, conversely, a given actor may be played by multiple different person.

Actor

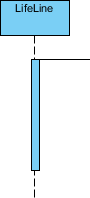
**Lifeline**

* A lifeline represents an individual participant in the Interaction.



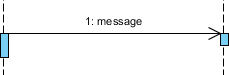
**Activations**

* A thin rectangle on a lifeline) represents the period during which an element is performing an operation.
* The top and the bottom of the of the rectangle are aligned with the initiation and the completion time respectively



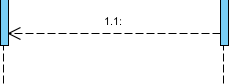
**Call Message**

* A message defines a particular communication between Lifelines of an Interaction.
* Call message is a kind of message that represents an invocation of operation of target lifeline.



**Return Message**

* A message defines a particular communication between Lifelines of an Interaction.
* Return message is a kind of message that represents the pass of information back to the caller of a corresponded former message.



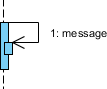
**Self Message**

* A message defines a particular communication between Lifelines of an Interaction.
* Self message is a kind of message that represents the invocation of message of the same lifeline.



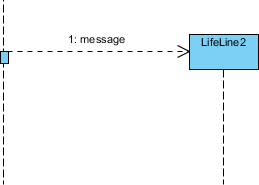
**Recursive Message**

* A message defines a particular communication between Lifelines of an Interaction.
* Recursive message is a kind of message that represents the invocation of message of the same lifeline. It's target points to an activation on top of the activation where the message was invoked from.



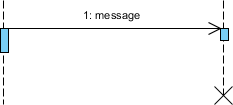
**Create Message**

* A message defines a particular communication between Lifelines of an Interaction.
* Create message is a kind of message that represents the instantiation of (target) lifeline.



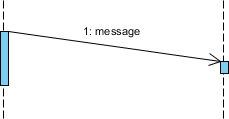
**Destroy Message**

* A message defines a particular communication between Lifelines of an Interaction.
* Destroy message is a kind of message that represents the request of destroying the lifecycle of target lifeline.



**Duration Message**

* A message defines a particular communication between Lifelines of an Interaction.
* Duration message shows the distance between two time instants for a message invocation.



**Note**

A note (comment) gives the ability to attach various remarks to elements. A comment carries no semantic force, but may contain information that is useful to a modeler.

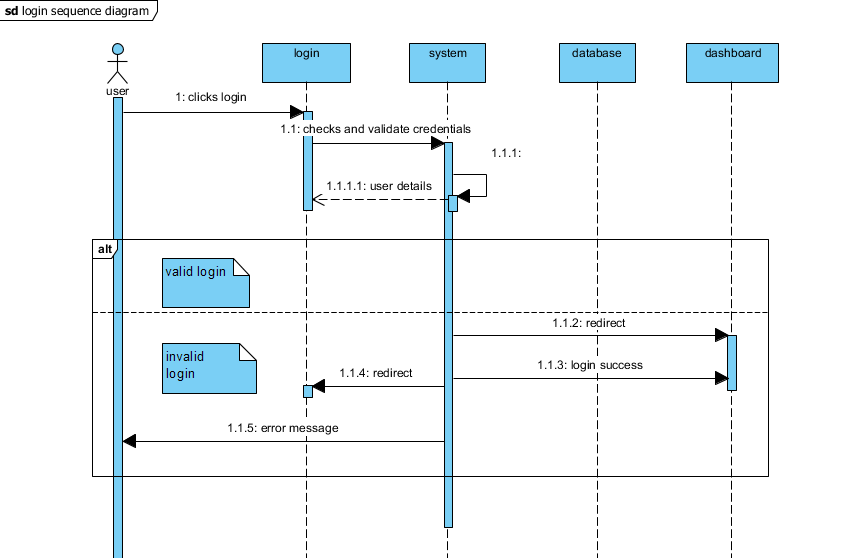


Fig: Login Sequence Diagram

This diagram shows the sequence of the actions involved during the login phase of the system.

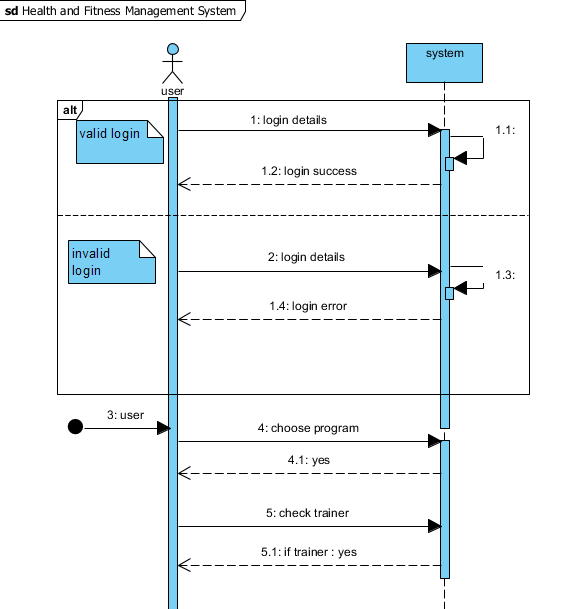


Fig: System Sequence Diagram

This the representation of the overall system sequence. This system sequence diagram shows the functionality sequence of the system.

# 3.3 Database Modelling

Data modeling is the process of creating a data model for the data to be stored in a Database. This data model is a conceptual representation of

Data objects

The associations between different data objects

The rules.

# 3.3.1 Data Dictionary

A data dictionary is a file or a set of files that contains a database's metadata. The data dictionary contains records about other objects in the database, such as data ownership, data relationships to other objects, and other data.

Here for Health and Fitness Management System I have created the following Data Dictionary.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Field Name** | **Data Type** | **Field Length** | **Constraint** | **Description** |
| admin\_id | Int | 10 | Primary key | Admin of the system |
| admin\_pass | Varchar | 20 | Not null | Login password for admin |
| user\_id | Int | 10 | Primary key | User |
| user\_name | Varchar | 20 | Not null | User detail |
| address | varchar | 20 | Nullable | User detail |
| user\_pass | Varchar | 20 | Not null | Login password for user |
| email | Varchar | 30 | Not null | User detail |
| height | Int | 5 | Not null | Required for selection of program |
| weight | Int | 5 | Not null | Required for selection of program |
| mass\_up | varchar | 25 | Not null | Available programs |
| fat\_loss | Varchar | 25 | Not null | Available programs |
| workout | Varchar | 25 | Not null | Available programs |
| diet | Varchar | 25 | Not null | Available programs |
| blog | Varchar | 20 | Not null | User can post articles |
| Trainer\_name | Varchar | 20 | Not null | Selected by the user |
| trainer\_id | int | 10 | Primary key | Trainer id is required |

**ADMIN**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Field Name** | **Data Type** | **Field Length** | **Constraint** | **Description** |
| admin\_uname | varchar | 20 | Primary key | Admin of the system |
| admin\_pass | Varchar | 20 | Not null | Login password for admin |
| del\_user |  |  |  |  |
| del\_forum |  |  |  |  |
| del\_trainer |  |  |  |  |
| add\_trainer |  |  |  |  |

**USER**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Field Name** | **Data Type** | **Field Length** | **Constraint** | **Description** |
| user\_uname | varchar | 20 | Primary key | user of the system |
| user\_pass | Varchar | 20 | Not null | Login password for users |

**SLIDER**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Field Name** | **Data Type** | **Field Length** | **Constraint** | **Description** |
| sider\_id | integer | 10 | Primary key | Id of slider |
| slider\_name | Varchar | 20 | Not null | Name of slider |
| slider\_img | varchar | 20 | Not null | Slider image |

**SER-REGISTER**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Field Name** | **Data Type** | **Field Length** | **Constraint** | **Description** |
| name | varchar | 20 | Not null | Name of user |
| address | varchar | 20 | Nullable | User detail |
| email | Varchar | 30 | Not null | User detail |
| height | Int | 5 | Not null | Required for selection of program |
| weight | Int | 5 | Not null | Required for selection of program |
| Phone num | int | 10 | Not null | User derail |
| program | string | 25 | Not null | User detail |
| user\_uname | string | 25 | Primary key | username |
| user\_pass | string | 25 | Not null | User password for login |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Field Name** | **Data Type** | **Field Length** | **Constraint** | **Description** |
| forum\_id | int | 10 | Primary key | Id of forum |
| forum\_title | Varchar | 50 | Not null | Title of forum |
| forum\_img | Varchar | 25 | Not null | Forum image |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Field Name** | **Data Type** | **Field Length** | **Constraint** | **Description** |
| diet\_id | int | 10 | Primary key | Id of diet |
| diet\_title | Varchar | 50 | Not null | Title of diet |
| diet\_img | Varchar | 25 | Not null | diet image |

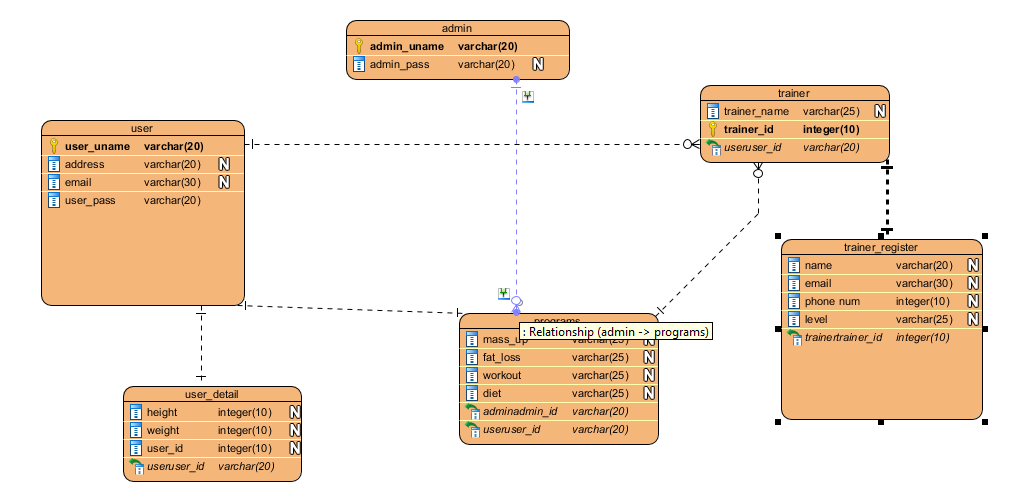
|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Field Name** | **Data Type** | **Field Length** | **Constraint** | **Description** |
| workout\_id | int | 10 | Primary key | Id of workout |
| workout \_title | Varchar | 50 | Not null | Title of workout |
| workout\_img | Varchar | 25 | Not null | workout image |

**TRAINER-REGISTER**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Field Name** | **Data Type** | **Field Length** | **Constraint** | **Description** |
| name | varchar | 20 | Not null | Name of trainer |
| email | Varchar | 30 | Not null | Trainer detail |
| Phone num | Int | 10 | Not null | Trainer detail |
| level | varchar | 50 | Not null | Trainer detail |
| trainer\_uname | varchar | 25 | Primary key | Username for trainer |
| trainer\_pass | varchar | 25 | Not null | Password for login |

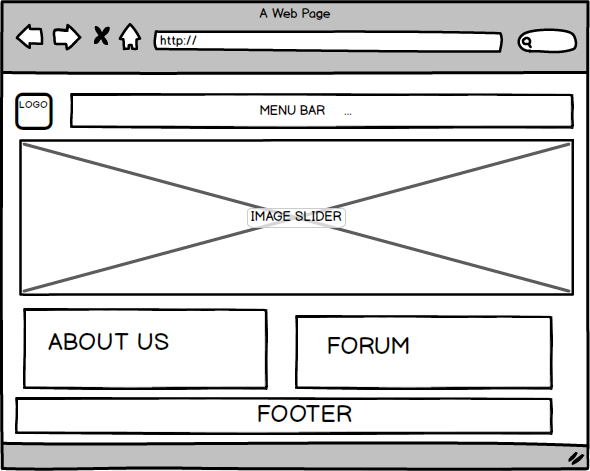
# 3.3.2 ER Diagram

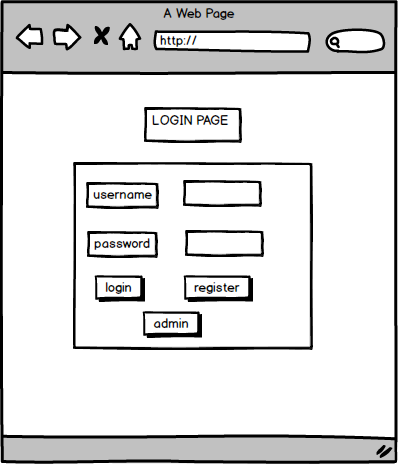
Here is the ER diagram I have created for Health and Fitness Management System

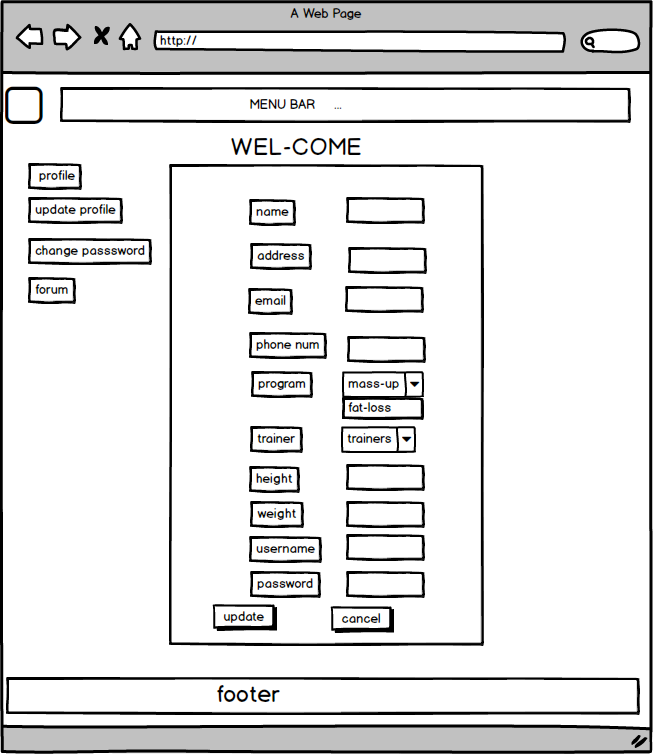
fig:ER Diagram

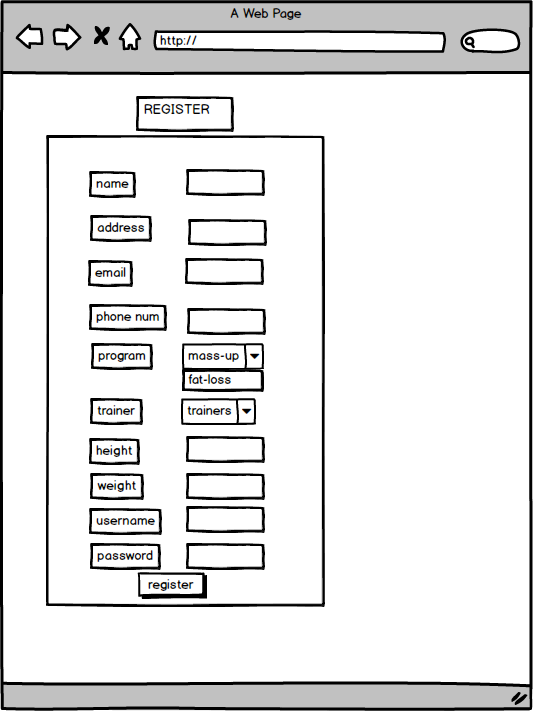
# 3.4 Prototype

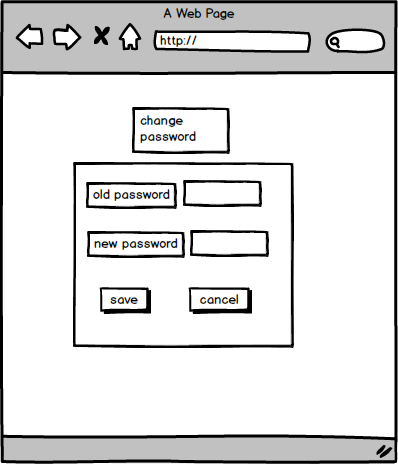
Here is the prototype I have created for Health and Fitness Management System

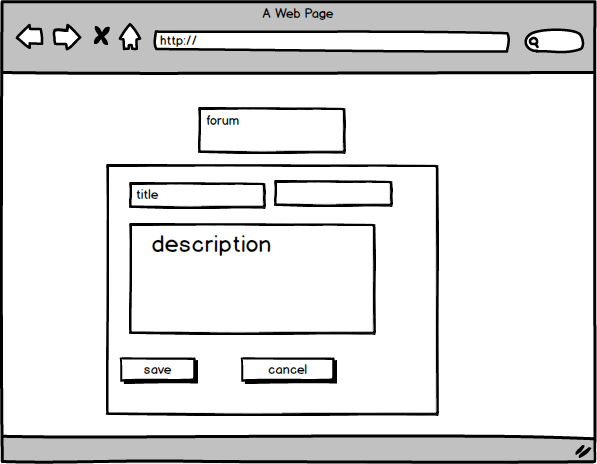


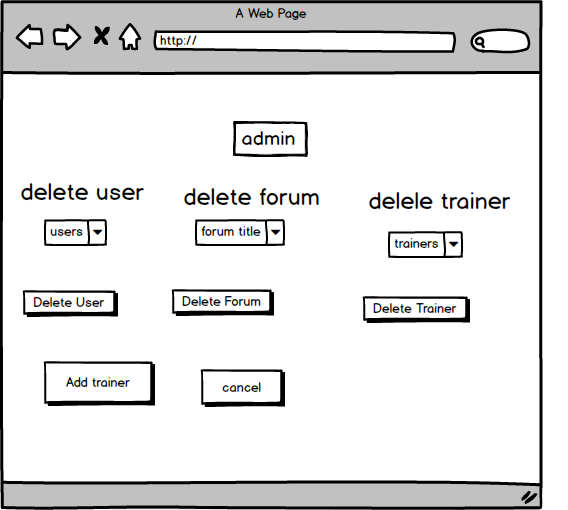


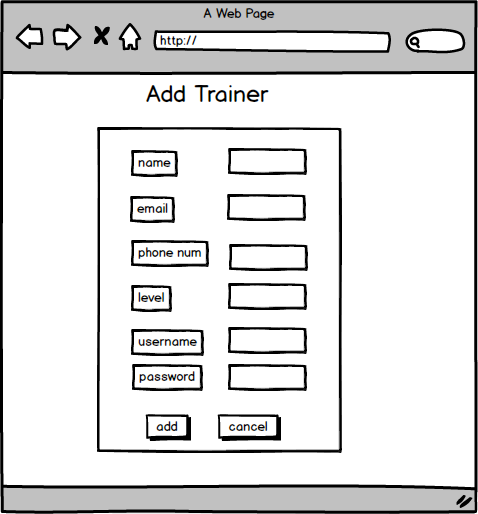


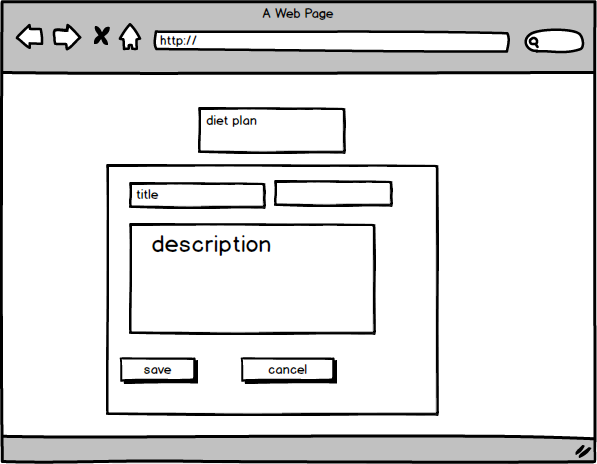
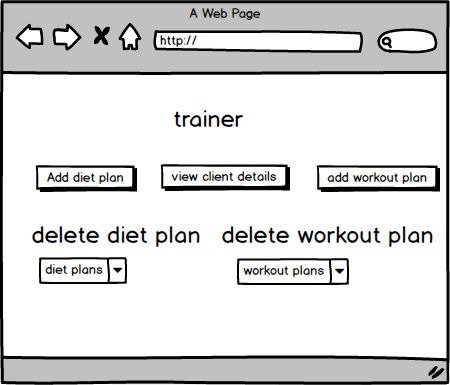


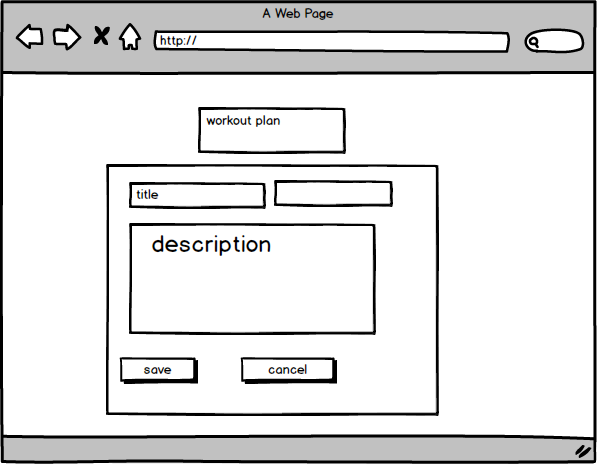


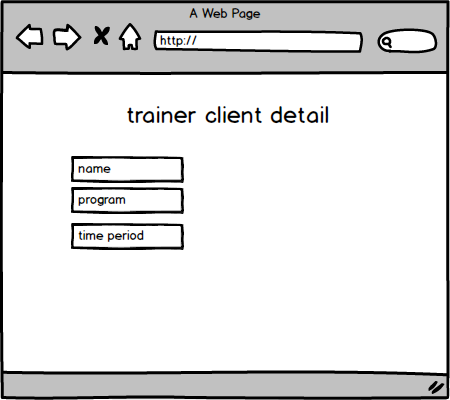












In this way I have created Prototype for the Health and Fitness Management System using Balsamiq.