



SWIM-Suite

A Swim Meet and Swimmer ManagementSoftware

Object Oriented Analysis Document

Submitted on: October 15th, 2010

Team Members (in alphabetical order):

AbhishekBindiganavile

Robert Greenberg

SedatOzer

Jay Takle

Contents

1Prioritization of Use Cases	3
2 Object Model.....	4
2.1Entity Objects for the Register use case	2
2.2Boundary Objects for the RegisterRace use case	2
2.3Control Objects for the RegisterRace use case.....	3
3Class Diagram	5
4Sequence Diagrams	2
4.1CreateMeet.....	2
4.2CreateProfile	2
4.3CreatRace.....	3
4.4SystemLogin	2
4.5DeleteMeet.....	2
4.6RaceRegister	3
5State Chart Diagrams	4
5.1State Chart for Object Race	2
5.2State Chart for Object Swimmer	2

1. Prioritization of Use Cases

Use Case	Description	Priority
Login	This use case checks whether the user is an organizer or a swimmer and display the relevant startup page accordingly.	5
CreateProfile	This use case allows the swimmer to create their profile which will be used to register the swimmer in eligible races.	4
Register	This use case allows the swimmer to register for a meet/race.	4
CreateMeet	This use case allows the organizer to create a new meet. Races can then be added to this meet.	4
ModifyMeet	This use case allows the organizer to modify an existing meet, this use case can also be extended to ModifyRace.	4
ViewStatistics	This use case allows the swimmer/organizer to view the results of present/past race.	3
SeedSwimmer	This use case allows the organizer to seed swimmers for the race.	2
DeleteMeet	This use case allows the organizer to delete a meet, in the event of non-participation or any other plausible reason.	1

2. Object Model – Data Dictionary

2.1 Entity Objects for the Register use case.

Entity Objects	Attributes and Associations	Definition
Swimmer	-name -id	A swimmer is the one who wishes to register for a race. The swimmer logs into the system with his name and unique ID and views a list of meets/races he/she is eligible for and proceeds to register for them.
Meet	-name -num_of_races -date -race_types	Meet is defined primarily by its name. Meet contains the number of races and the types of races. The meet object also specifies the date on which the meet is to be held.
Profile	-name -email -age -id -previous_results	Profile contains the profile details of the swimmer and is capable of listing the same upon request from the swimmer.
Database	-add -delete	The database primarily contains the meet, race, swimmer profile details in a database format and is capable of returning as and when a query is performed on it.

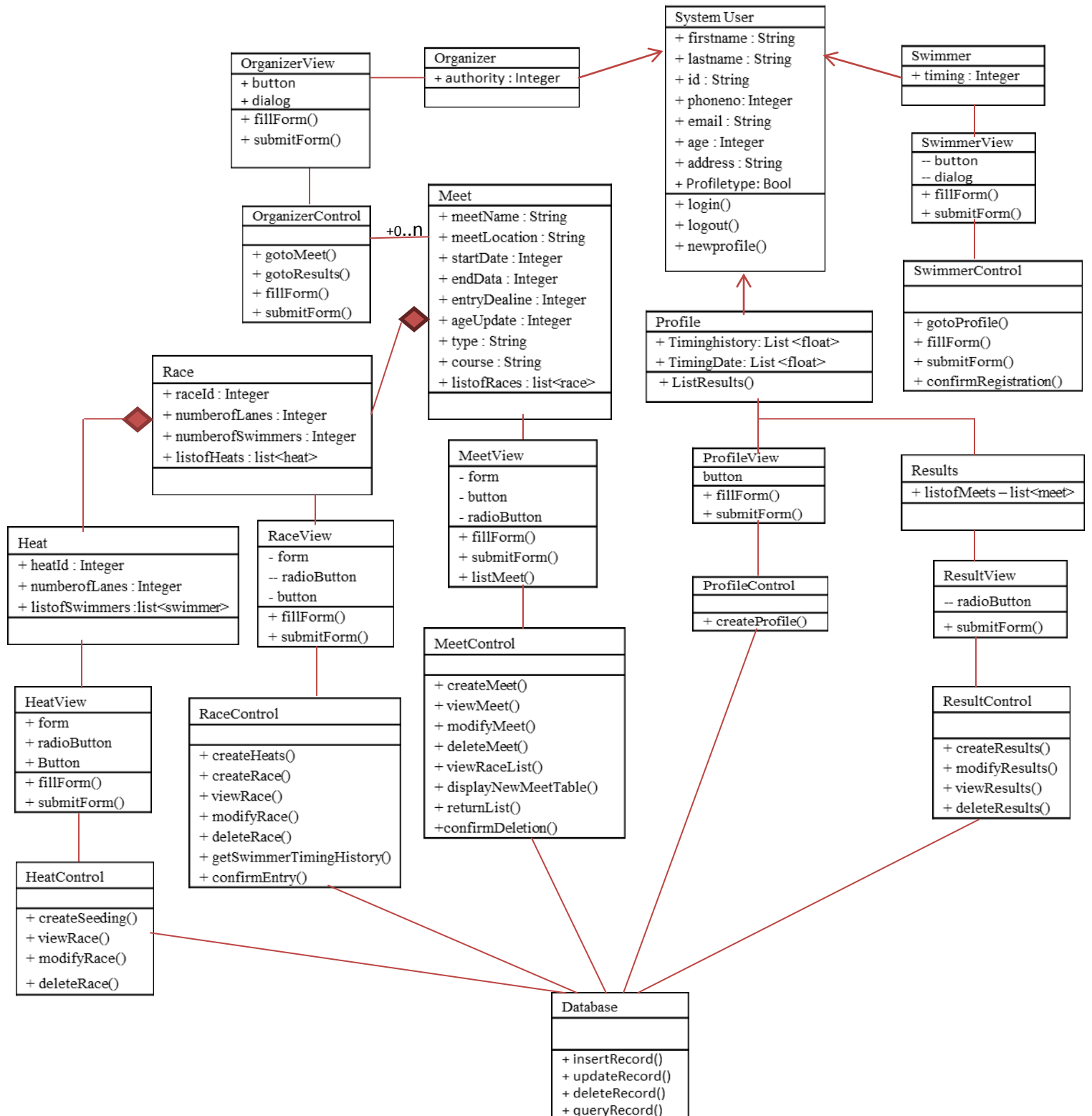
2.2 Boundary Objects for the RegisterRace Use case.

Boundary Objects	Description
LoginPage	The user enters their name and ID.
AccessDeniedPage	The system shows this page if there is no profile of the entered name and ID.
LoginButton AddRaceButton ViewRacesButton ViewResultsButton ProgressChartButton	The following buttons are implemented at various stages in the RegisterRace use case to initiate the RegisterRace object. ProgressChartButton is used to initiate, displaying the swimmers results in a graph.

2.1.3 Control Objects for Register Race Use Case .

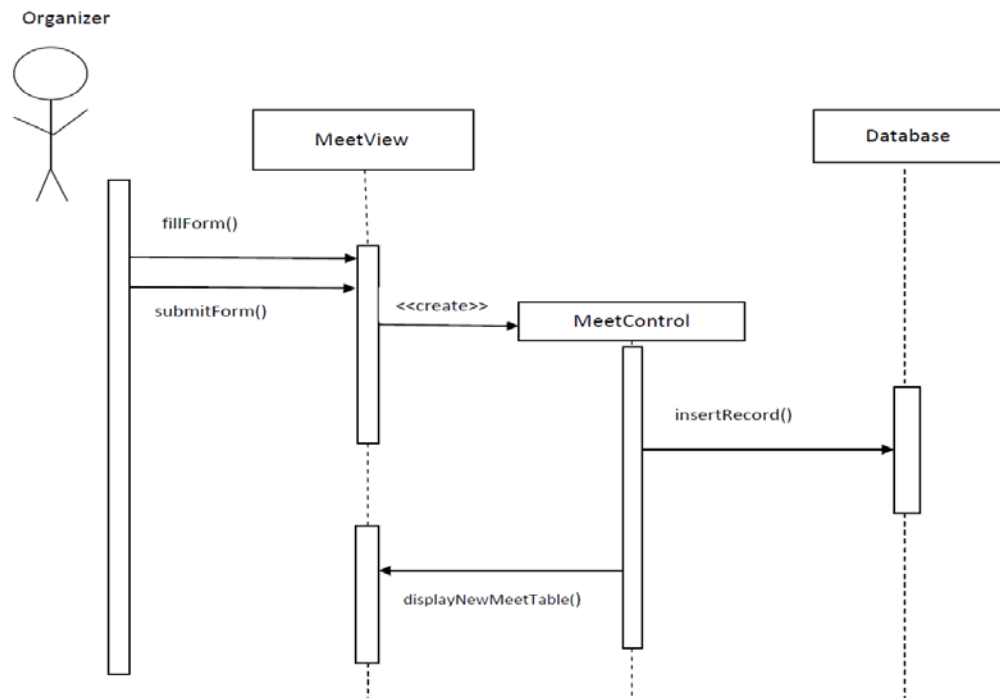
Control Object	Description
RegisterTerminal	RegisterTerminal is an object for the terminal that the swimmers use to log in and register for the races.
RaceControl	RaceControl object handles the functions to be performed by the system when the swimmer makes certain requests. RaceControl establishes connection to the user's view page by displaying the relevant forms and confirmation pages. RaceControl also establishes a connection to the database and stores the respective values from the forms into database tables.

4. Class Diagram

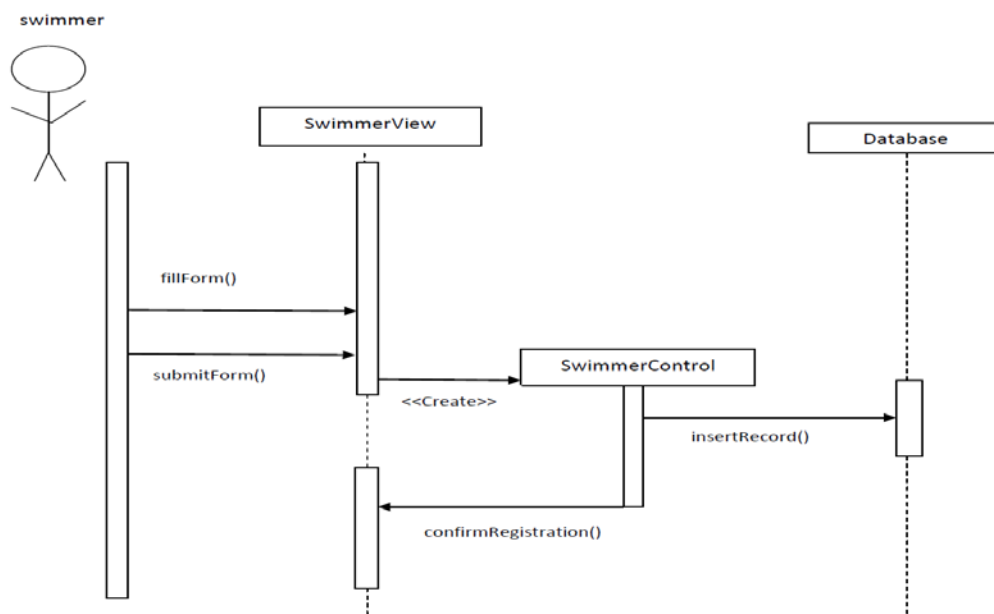


4. Sequence Diagrams:

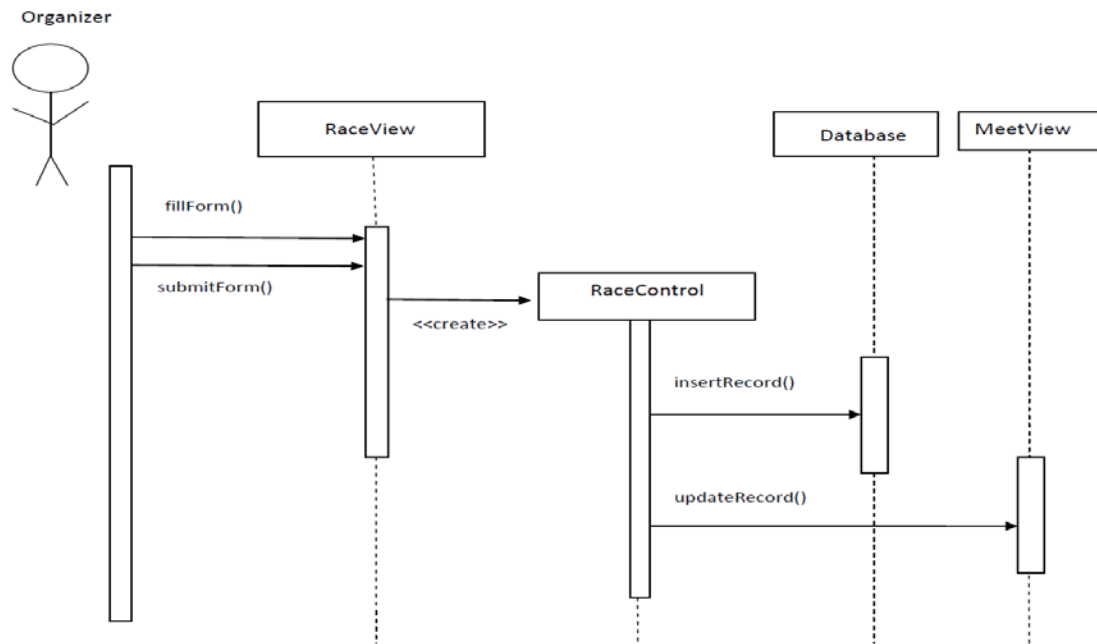
4.1 CreateMeet



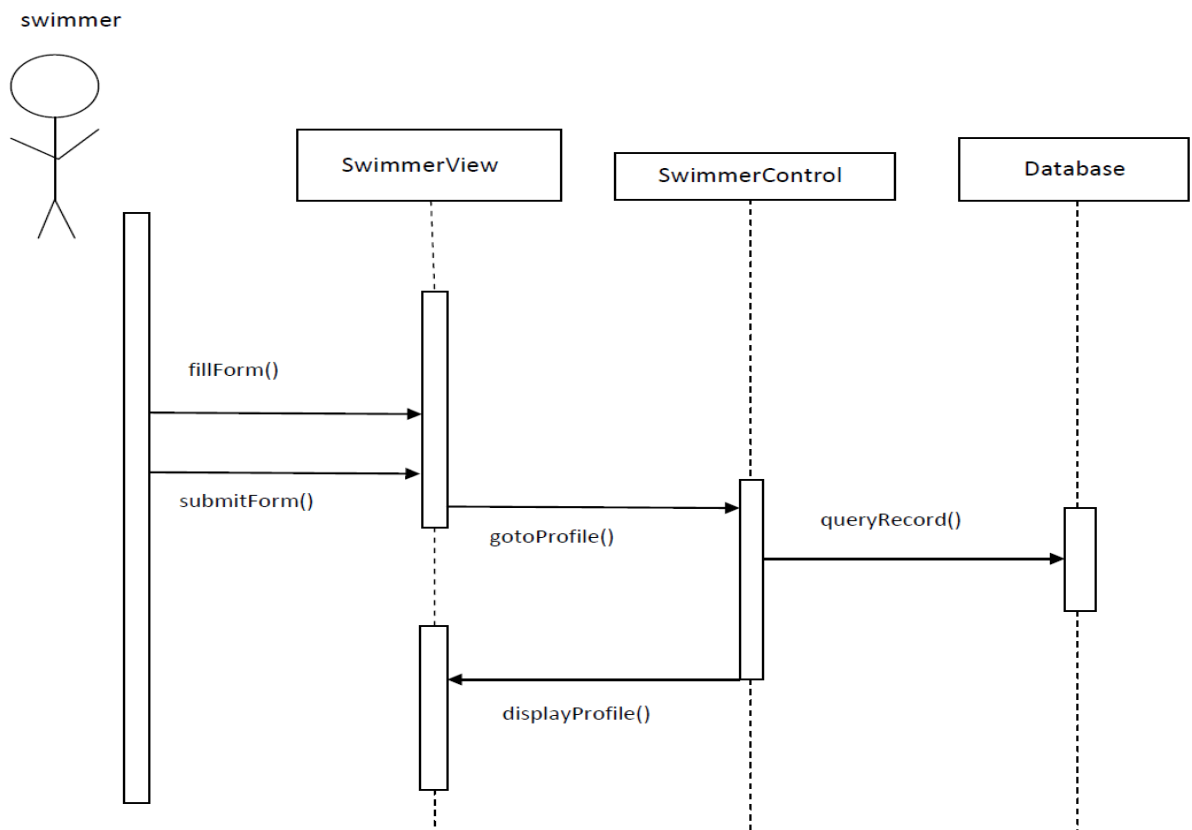
4.2 CreateProfile



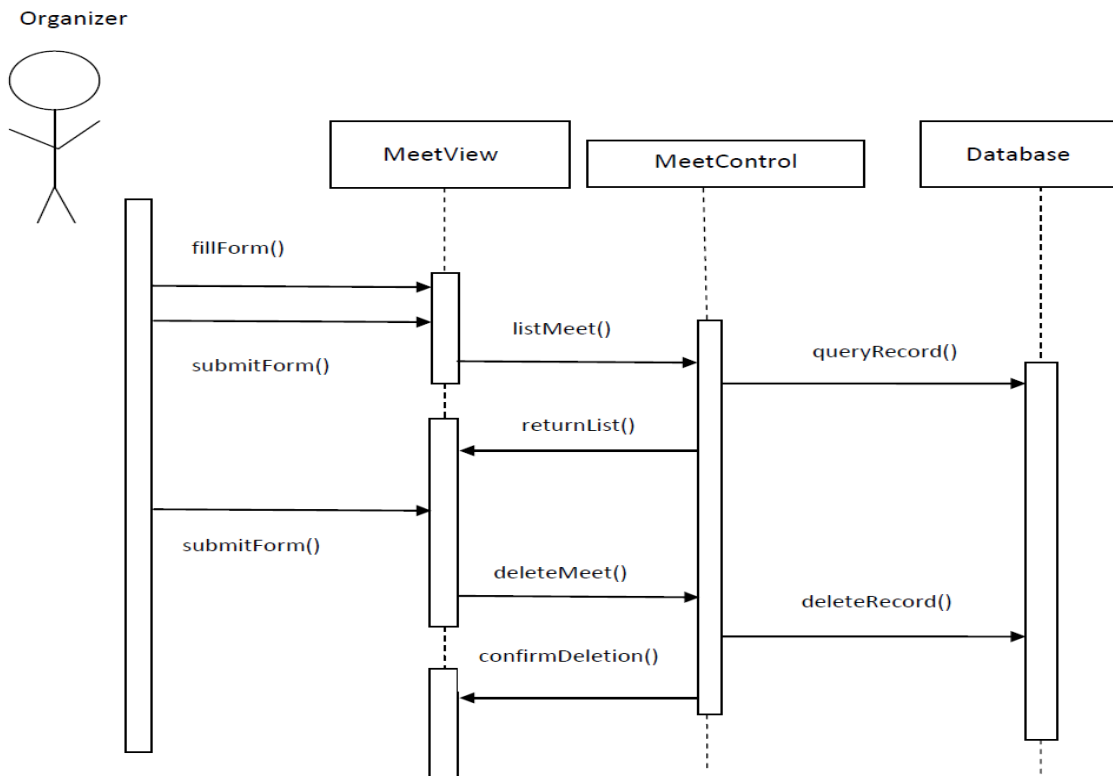
4.3 CreateRace



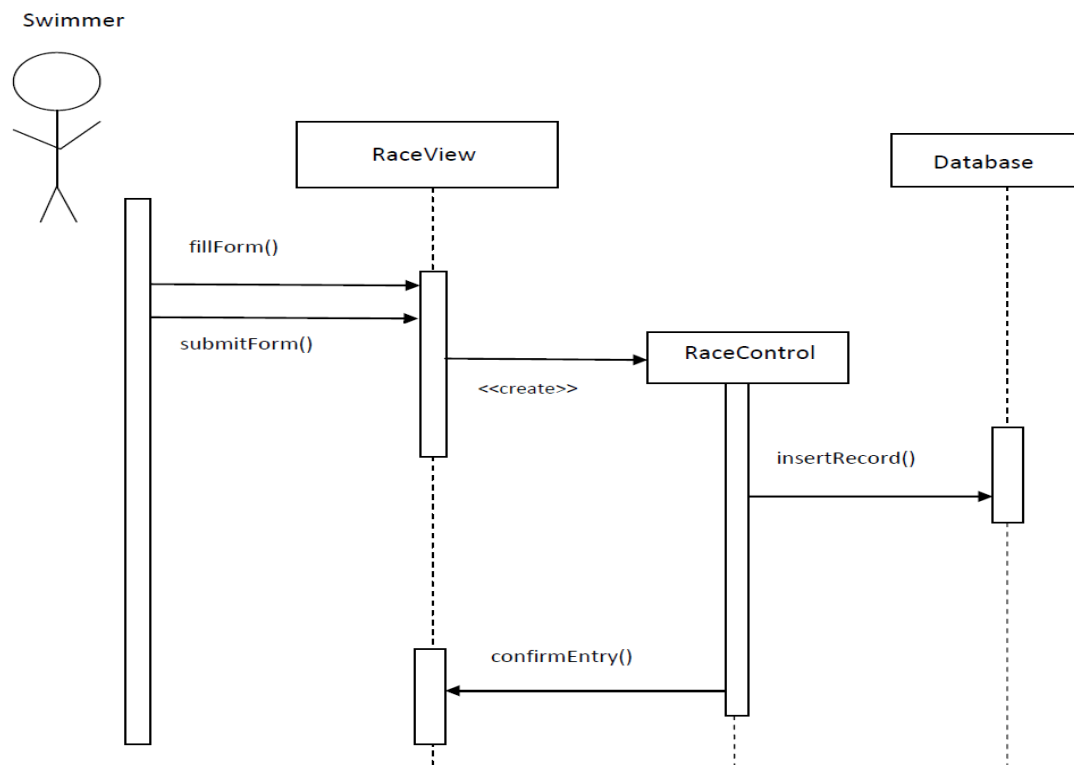
4.4 SystemLogin



4.5 DeleteMeet



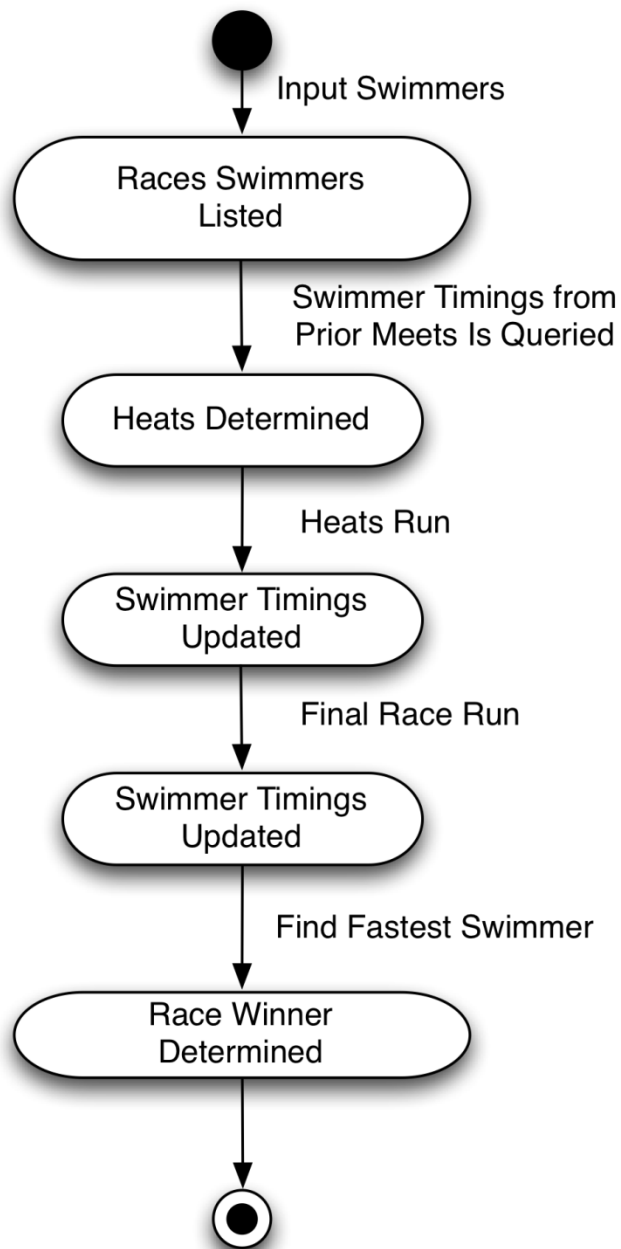
4.6 RaceRegister



5. State Chart Diagrams:

5.1 State Chart for Object Race

Race



5.2 State Chart for Object Swimmer

Swimmer

