

Florida International University
School of Computing and Information Sciences

CIS 4911 - Senior Capstone Project
Software Engineering Focus

Final Deliverable

Project Title: Contest Registration System 1.0

Team Members:

Eduardo Guerra
Wayne Curling

Product Owner(s):

Kip Irvine

Mentor(s):

N/A

Instructor: Masoud Sadjadi

The MIT License (MIT)
Copyright (c) <2016> <copyright holders>

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

Abstract

The Ultimate Software Academy for Computer Science Education holds programming competitions for high school student teams. However with the competitions growing every year the current process of organization such a large event has become inefficient and time consuming for both project organizers and coaches of teams. This project set out to create an interactive website that makes it easy for coaches and contest administrators to collaborate on the registration process.

This document describes the approach taken to complete the requirements set by the project owner. The Registration System v1.0 is an interactive website to allow for a coach and administrator to have centralized source of information about the programming competition. The application makes it easy for coaches to register and manage their teams and allows administrators to monitor the registration information and send notifications to teams.

Table of Contents

INTRODUCTION

CURRENT SYSTEM

PURPOSE OF NEW SYSTEM

USER STORIES

IMPLEMENTED USER STORIES

PENDING USER STORIES

PROJECT PLAN

HARDWARE AND SOFTWARE RESOURCES

SPRINTS PLAN

Sprint 1

Sprint 2

Sprint 3

Sprint 4

Sprint 5

Sprint 6

Sprint 7

SYSTEM DESIGN

ARCHITECTURAL PATTERNS

SYSTEM AND SUBSYSTEM DECOMPOSITION

DEPLOYMENT DIAGRAM

DESIGN PATTERNS

SYSTEM VALIDATION

GLOSSARY

APPENDIX

APPENDIX A - UML DIAGRAMS

Static UML Diagrams

Dynamic UML Diagrams

APPENDIX B - USER INTERFACE DESIGN

APPENDIX C - SPRINT REVIEW REPORTS

APPENDIX D - SPRINT RETROSPECTIVE REPORTS

REFERENCES

INTRODUCTION

In this chapter will detail the current system in place for the organization of high school programming competitions. It will also explain the new system that was to created to remove the inefficiencies of the current system. The User Stories chapter will describe the user stories that were created from our initial requirements document and which of those were implemented. The Project Plan chapter will describe the hardware and software requirements for the system. It will also provide an explanation of each sprint and its completed user stories and tasks. The System Design chapter will provide visual representations of the system design and patterns that are used. The System Validation chapter will describe the testing approach taken during and after development. Any technology specific terms will be defined in the Glossary chapter. UML Diagrams, UI design, and Sprint Retrospective/Review reports will be given in the Appendix.

Current System

The programming competition put on by Ultimate Software and Florida International University has the bulk of its work structured around the contest organizers. Currently the programing competition website gives information about current and previous competitions and the ability for coaches so sign up there team for an upcoming competition. The teams are then placed in a database and will be extracted in excel format for the competition organizer/administrator to review.

The current limitations on the system are based on the fact that once a coach submits his team for the competition, their interaction with the system is over. If any changes need to be made to the teams the coach needs to contact the administrator to make updates on his behalf, and the administrator must then search through an excel sheet for a particular team to make the updates. With over 100 students and 12 schools from around Florida competing in the 2016 competition, it can get very overwhelming for an administrator to manage.

Purpose of New System

Registration System v1.0 is an interactive website for coaches and administrators to collaborate in the creation and updating competition information. In the newly introduced system, a multiple role system has been added that allows for a coach and an administrator to have separate views of information that they can interact with. A coach can now create an account within the system for easier access to information that pertains to their role. Originally a coach would sign up a team on a form, then submit it to the administrator for review. In the new system a coach will still be able to create a team and add participants, but it will also give a detailed view of all the teams a coach has created. A coach can also modify any information about the participants he has added to a team.

The administrator also has the ability to log into the system, but has access to more detailed information about the teams participating in the competitions. The administrator now gets a easy to navigate view of all schools/coaches participating, and is now able to search for a particular school/coach and get that information in a tabular format. The administrator also now has the ability to remove schools from a competition at the click of a button instead of having to remove them through excel. The administrator can now take all the information about a contest(participants, schools, coaches) and archive them for a later viewing with the same ease as the other features.

USER STORIES

The Contest Registration System used the agile process of Scrum for software development. This chapter contains a comprehensive list of all the user stories gathered from the product owner. They are separated into implemented user stories that were completed and pending user stories to be carried over into the next version of the system.

Implemented User Stories

User Story #678 - Modify Teams as Coach

As a coach, I would like to be able to view, select, modify or delete a single team.

User Story #676 - Create Account for Coaches(Required)

As a coach, I would like to be able to create accounts, so I may register teams for the competition.

User Story #687 - Create Teams as Coach(Required)

As a coach, I would like to create a team, so I may register my participants for the contests.

User Story #688 - Login for Coach/Admin(Required)

As a coach/admin, I would like to be able to sign in, so I can access my customized pages.

User Story #679 - Add a Participant(Required)

As a coach, I would like to be able to add a new participant, so I may register my participants for the contests.

User Story #677 - Modify Account(Required)

As a coach, I would like to be able to modify my account, so that I may edit my information.

User Story #681 - View Teams as Administrator(Required)

As an administrator, I would like to view all registered teams, so that I may see all teams participating in contests.

User Story #685 -View List of Schools as Administrator(Required)

As an administrator, I would like to view and sort the list of all schools, so I may know all participating schools and coaches.

User Story #694 - View Active Contests as Administrator(Required)

As an administrator, I would like to view all active contests, so I may manage them.

User Story #695 - View Stored Contests as Administrator(Required)

As an administrator, I would like to see all stored contests, so that I may review the information of previous contests.

User Story #696 - Create Contest(Required)

As an administrator, I would like to create contests, so that coaches and their teams may register for them.

User Story #684 - View Teams as Administrator(Required)

As an administrator, I would like to view the list of all team member, so I may manage the participants of the contests.

User Story #683 - View Other Administrators(Required)

As an administrator, I would like to view a list of all administrators, so I may manage all super-users of the system.

User Story #718 - Delete Administrators(Required)

As an administrator, I would like to be able to delete other administrators, so I may manage all super-users of the system.

User Story #680 - Create an Administrator Account(Required)

As an administrator, I would like to be able to create an administrator account, so I add another person to help with management of contests.

User Story #686 - Store Contest List(Required)

As an administrator, I would like to store all registration data for a contest, so that the participation data can later be reviewed.

User Story #709 - Remove Coach (Discarded)

As an administrator, I would like to delete coaches, so I may remove them from the website once they are no longer participating.

User Story #721 - Search a Team as Administrator(Required)

As an Administrator, I would like to search for a team, so I may see the specifics of that team.

User Story #727 - Delete Schools(Required)

As an administrator, I would like to remove schools, so I may better manage the users on the system.

Pending User Stories

User Story #682 - Send Email Notifications as Admin

As an administrator, I would like to view the list of all coaches so that I may send them an email.

User Story #689 - Recover Password

As a coach, I would like to be able to recover my password, so I may continue logging in to the site.

PROJECT PLAN

The Contest Registration System was the first iteration of this project, so the first two sprints were dedicated to the setup and design of the system. The coach and administrator side of the functionality were developed concurrently by Wayne Curling and Eduardo Guerra respectively, starting on the third sprint. There were changes made to both ongoing stories and the original requirements, which were handled according to Agile methodology.

Hardware and Software Resources

Operating System: Linux - Requested by Product Owner.

Server: Apache Tomcat - Requested by Product Owner

Database: MySQL - Requested by Product Owner

Application Logic: Java/JSP - User familiarity with the language.

Sprints Plan

Sprint 1

(01/16/2016 - 01/29/2016)

Tasks

- Concept review
- User Story Creation
- Server Setup
- Database Setup

Sprint 2

(01/30/2016 - 02/12/2016)

Tasks

- System Prototyping
- Front End Development
- Tomcat Setup

Sprint 3

(02/13/2016 - 2/26/2016)

User Story #678 - Modify Teams as Coach***Tasks***

- Documentation
- Create page to display teams
- Create form to modify team information
- Create logic to modify team information in database.

Acceptance Criteria

- Deletion of the team must result in the members of the team having being deleted.
- A team can have no more than 5 members.
- Only participants created by the coach can be added to the team.

Modeling

Appendix A - Figure #2 - User Story #678 Use Case Diagram

Appendix A - Figure #2 - User Story #678 Sequence Diagram

Appendix A - Figure #2 - User Story #678 Class Diagram

User Story #676 - Create Account for Coach***Tasks***

- Documentation
- Create form for registration data input
- Create logic to processing coach information and insertion into database.

Acceptance Criteria

- Coach must be able to create an account with the first name, last name, email, password, and school name.
- Validation: First name, last name, email, password, and school name must be required, and the system must prompt the coach with a specific message if any field is left empty.
- Security: Input fields must be protected against SQL injection.

Modeling

Appendix A - Figure #1 - User Story #676 Use Case Diagram

Appendix A - Figure #1 - User Story #676 Sequence Diagram

Appendix A - Figure #1 - User Story #676 Class Diagram

User Story #687 - Create Team as Coach***Tasks***

- Documentation
- Create form to enter team information.
- Create logic to process entered team information and database insertion.

Acceptance Criteria

- Coach must be able to create a team with an unique team name.
- Validation: Team name is required and the system must prompt the coach with a specific message if the field is left empty.
- Security: Input fields must be protected against SQL injection.

Modeling

Appendix A - Figure #3 - User Story #687 Use Case Diagram

Appendix A - Figure #3 - User Story #687 Sequence Diagram

Appendix A - Figure #3 - User Story #687 Class Diagram

User Story #688 - Login for Coach/Admin***Tasks***

- Documentation
- Create page with login form
- Create logic to determine whether user is administrator or coach.
- Create logic to create session and session parameters.

Acceptance Criteria

- Coach/admin must be able to login with the email and password of a registered account.
- Validation: Email and password are required.
- Security: All fields must be protected against SQL injection.

Modeling

Appendix A - Figure #4 - User Story #688 Use Case Diagram

Appendix A - Figure #4 - User Story #688 Sequence Diagram

Appendix A - Figure #4 - User Story #688 Class Diagram

Sprint 4

(02/27/2016 - 03/11/2016)

User Story #679 - Add a Participant as Coach***Tasks***

- Documentation
- Create form to add a new participant
- Create logic to process participant information and database insertion.

Acceptance Criteria

- Created participant must be linked to the coach that created it.
- Participants must have a first name, last name, and email.
- Validation: All input fields are required.
- Security: All fields must prevent SQL injection

Modeling

Appendix A - Figure #5 - User Story #679 Use Case Diagram

Appendix A - Figure #5 - User Story #679 Sequence Diagram

Appendix A - Figure #5 - User Story #679 Class Diagram

User Story #677 - Modify Account as Coach***Tasks***

- Documentation
- Create page to display current information for coach.
- Create form to allow information editing.
- Create logic to process changes

Acceptance Criteria

- Coach must be able to modify their first name, last name, email, password and school name.
- Coach should only be able modify the teams he/she created.
- Validation: Required fields will prompt the coach for correct input if validation is not met.
- Security: All Input fields must be protected against SQL injection.

Modeling

Appendix A - Figure #6 - User Story #677 Use Case Diagram

Appendix A - Figure #6 - User Story #677 Sequence Diagram

Appendix A - Figure #6 - User Story #677 Class Diagram

User Story #681 - View Teams as Administrator

Tasks

- Documentation
- Create page to display team information

Acceptance Criteria

- Administrator must be able to view all registered teams.

Modeling

Appendix A - Figure #7 - User Story #681 Use Case Diagram

Appendix A - Figure #7 - User Story #681 Sequence Diagram

Appendix A - Figure #7 - User Story #681 Class Diagram

User Story #685 - View List of Schools as Administrators

Tasks

- Documentation
- Create page to display school information

Acceptance Criteria

- Each record must display the school name, coach name, email and list of team names.

Modeling

Appendix A - Figure #8 - User Story #685 Use Case Diagram

Appendix A - Figure #8 - User Story #685 Sequence Diagram

Appendix A - Figure #8 - User Story #685 Class Diagram

User Story #694 - View Active Contests as Administrator

Tasks

- Documentation
- Create page to display active contests information

Acceptance Criteria

- The name, time, date, location, creator, and description of all active contest must be visible.

Modeling

Appendix A - Figure #9 - User Story #694 Use Case Diagram

Appendix A - Figure #9 - User Story #694 Sequence Diagram

Appendix A - Figure #9 - User Story #694 Class Diagram

User Story #695 - View Stored Contests as Administrator***Tasks***

- Documentation
- Create page to view stored contest information

Acceptance Criteria

- The name, time, date, location, creator, and description of all stored contest must be visible.

Modeling

Appendix A - Figure #10 - User Story #695 Use Case Diagram

Appendix A - Figure #10 - User Story #695 Sequence Diagram

Appendix A - Figure #10 - User Story #695 Class Diagram

User Story #696 - Create Contest***Tasks***

- Documentation
- Create logic to switch contest from active to inactive.

Acceptance Criteria

- A contest is created with the name, date, time, location, and description entered by the administrator
- All contests will be active by default.
- Validation: Name, date, time, location, and description are required.
- Security: All input fields must be protected against SQL injection.

Modeling

Appendix A - Figure #11 - User Story #696 Use Case Diagram

Appendix A - Figure #11 - User Story #696 Sequence Diagram

Appendix A - Figure #11 - User Story #696 Class Diagram

Sprint 5

(03/19/2016 - 04/01/2016)

User Story #684 - View Teams as Administrator***Tasks***

- Documentation
- Create a page to display all team information

Acceptance Criteria

- All records must show a team member's name, email, t-shirt size and team name.
- All records must be able to be sorted in ascending order.

Modeling

Appendix A - Figure #13 - User Story #684 Use Case Diagram

Appendix A - Figure #13 - User Story #684 Sequence Diagram

Appendix A - Figure #13 - User Story #684 Class Diagram

User Story #683 - View Other Administrators***Tasks***

- Documentation
- Create a page to display all administrator information.

Acceptance Criteria

- All records must show the administrator's
- All records must be sorted in ascending order.

Modeling

Appendix A - Figure #12 - User Story #683 Use Case Diagram

Appendix A - Figure #12 - User Story #683 Sequence Diagram

Appendix A - Figure #12 - User Story #683 Class Diagram

User Story #718 - Delete Administrators***Tasks***

- Documentation
- Create logic to remove administrator from database.

Acceptance Criteria

- Administrator is removed from the database

Modeling

Appendix A - Figure #14 - User Story #718 Use Case Diagram

Appendix A - Figure #14 - User Story #718 Sequence Diagram

Appendix A - Figure #14 - User Story #718 Class Diagram

User Story #680 - Create an Administrator Account***Tasks***

- Documentation
- Create form to allow administrator to create a new administrator
- Create logic to process administrator information and database insertion.

Acceptance Criteria

- Administrator must be able to create an account with first name, last name, email, and password.
- Validation: All input fields are required.
- Security: Input fields must be protected against SQL injection.

Modeling

Appendix A - Figure #15 - User Story #680 Use Case Diagram

Appendix A - Figure #15 - User Story #680 Sequence Diagram

Appendix A - Figure #15 - User Story #680 Class Diagram

User Story #686 - Store Contest***Tasks***

- Documentation
- Create logic to set a contest from active to stored.

Acceptance Criteria

- Data, time, event creator, and participants must be stored in the database and displayed.

Modeling

Appendix A - Figure #16 - User Story #686 Use Case Diagram

Appendix A - Figure #16 - User Story #686 Sequence Diagram

Appendix A - Figure #16 - User Story #686 Class Diagram

User Story #709 - Remove Coach (Discarded)***Tasks***

- Documentation
- Create logic to remove coach from database.

Acceptance Criteria

- The selected coach is removed from the database.

Modeling

Appendix A - Figure #17 - User Story #709 Use Case Diagram

Appendix A - Figure #17 - User Story #709 Sequence Diagram

Appendix A - Figure #17 - User Story #709 Class Diagram

Sprint 6

(04/02/2016 - 04/15/2016)

User Story #721 - Search a Team as Administrator***Tasks***

- Documentation
- Create logic to filter team data based on user input.

Acceptance Criteria

- Team/Teams are displayed alongside coach and school name.
- Validation: The system must prompt the coach with a specific message if any field is left empty.
- Security: Input fields must be protected against SQL injection.

Modeling

Appendix A - Figure #18 - User Story #721 Use Case Diagram

Appendix A - Figure #18 - User Story #721 Sequence Diagram

Appendix A - Figure #18 - User Story #721 Class Diagram

User Story #727 - Delete Schools***Tasks***

- Documentation
- Create logic for cascading delete of schools and related teams, coaches, and participants.

Acceptance Criteria

- The selected school and related teams, coaches, and participants are deleted.

Modeling

Appendix A - Figure #19 - User Story #727 Use Case Diagram

Appendix A - Figure #19 - User Story #727 Sequence Diagram

Appendix A - Figure #19 - User Story #727 Class Diagram

Sprint 7

(04/16/2016 - 04/29/2016)

User Story #682 - Send Email Notifications***Tasks***

- Documentation
- Create a page to select coaches to receive the mail.
- Create logic to send email to selected recipients.

Acceptance Criteria

- Selected recipients receive the email.

Modeling

Discarded since story was put back to the product backlog.

SYSTEM DESIGN

This chapter includes the architectural pattern used to develop the system on top of. The system and subsystem decomposition will be described through the use of a UML package diagram and a Deployment diagram. This chapter will also explain the design patterns used during the development of each user story.

Architectural Patterns

The Contest Registration System uses a 3-Tier architecture. The Contest Registration System queries a lot of information from the database and the 3-tier architecture allows for a protection any potential misuse of the system due to the interface never making direct calls to the database. This architecture was perfect for a group project because since the tiers aren't connected it allows for different people to be working in a different tier at the same time. Most importantly if one tier fails no data is lost which is extremely important when dealing with user information.

System and Subsystem Decomposition

The Contest Registration System is composed of two subsystems, one for administrator and one for coach, which are the two actors of the system. The coach subsystem consists of creating and editing teams for competitions, and editing of personal information, for both participants and coaches. The admin subsystem consists of the management of all users in the system, including teams, coaches, participants, and administrators, as well as creation and storage of contests. The two subsystems are mostly independent, except for participants. The administrator's view of all participants depends on whether coaches have registered for the system and created teams and participants.

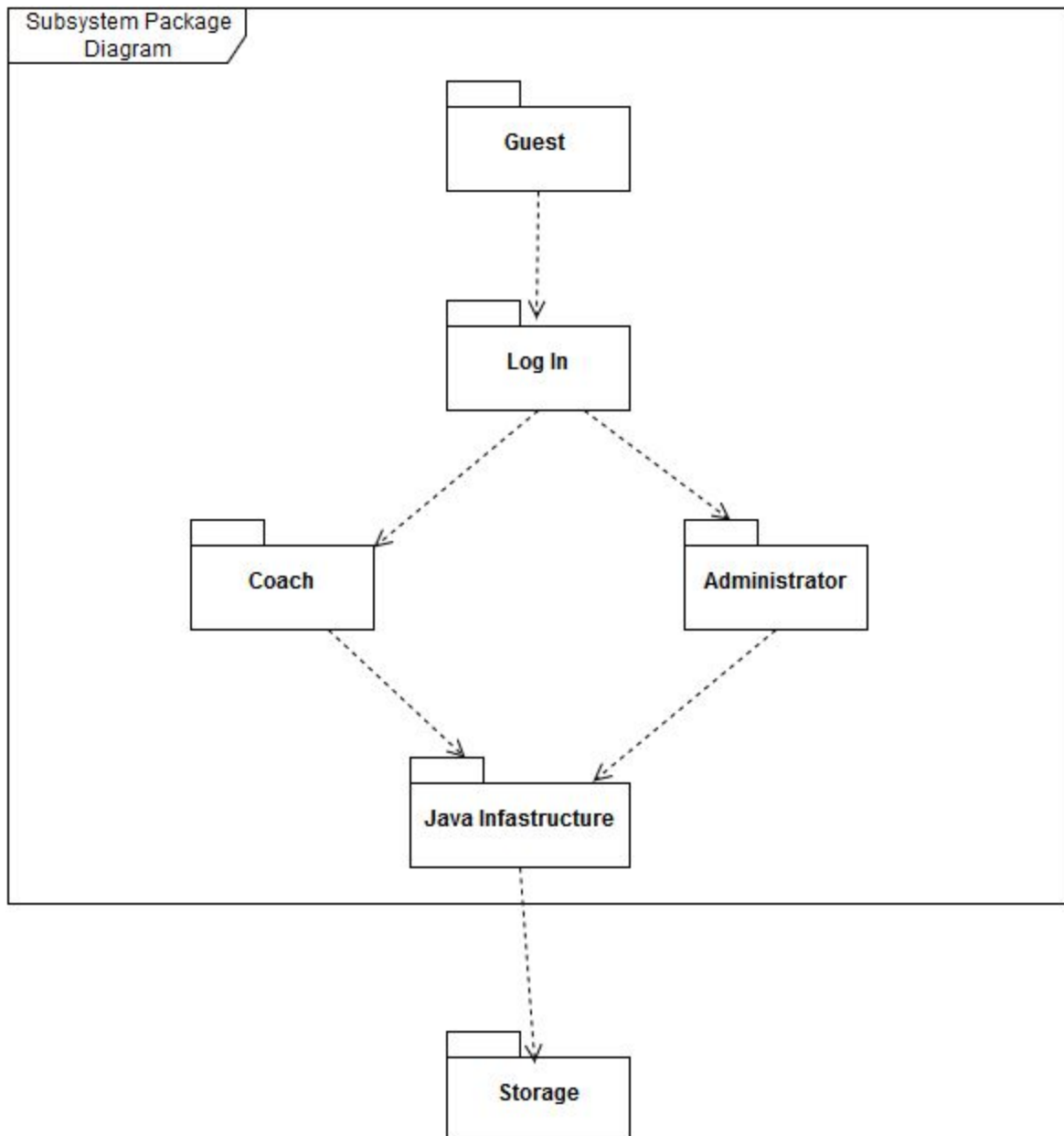
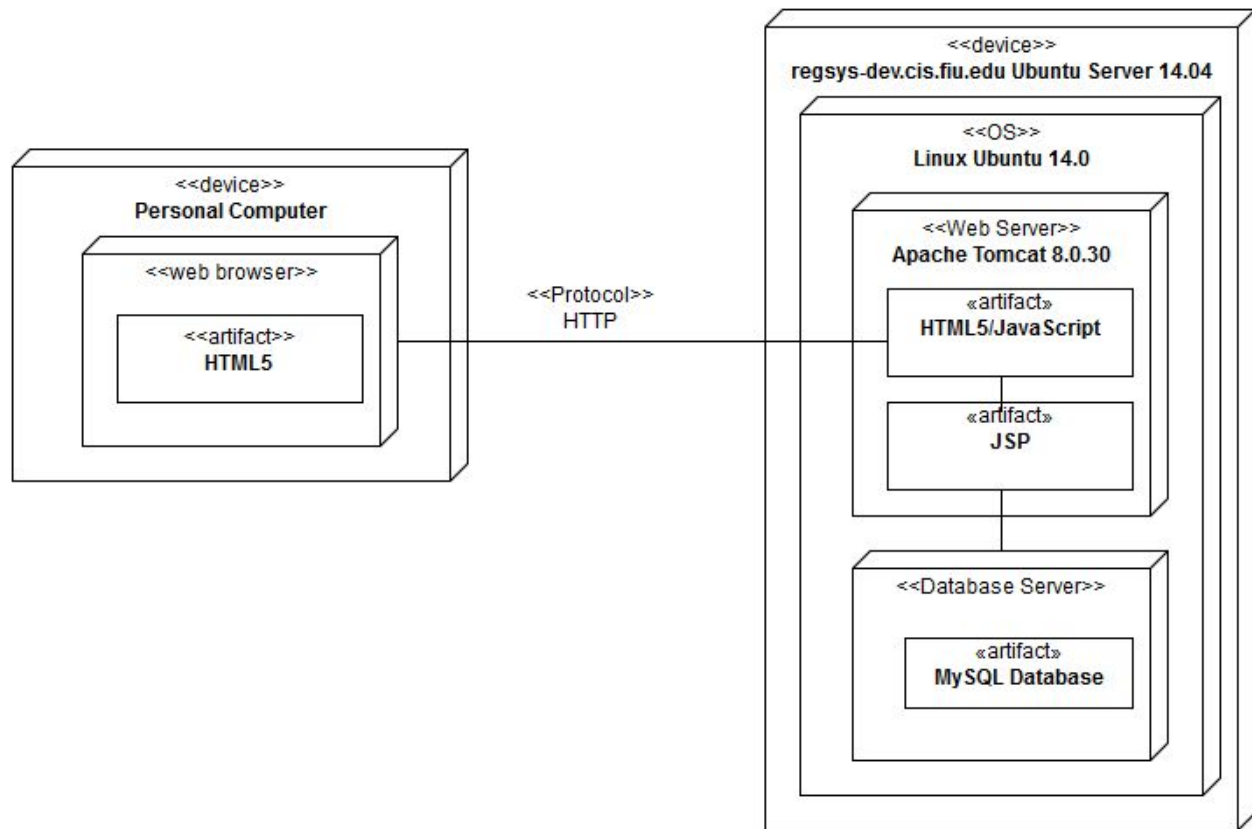


Figure #1 - Package Diagram

Deployment Diagram



Deployment Diagram

Design Patterns

The design pattern used throughout development of the system was the Model View Controller pattern. We chose this pattern for similar reasons to the 3-tier architecture. The MVC pattern is great for group projects because of the ability for different members of the project to be simultaneously be working on the project without worrying about interdependency issues. The pattern also allows for multiple user interfaces using the same business logic to allow for future development on phones, tablets, etc.

Identify the design patterns used and briefly explained why they were selected. You should not provide a definition of the pattern here, but rather explain its pertinence.

SYSTEM VALIDATION

This chapter will introduce all of the test cases done during the development of the system. Unit and Subsystem tests were done with satisfactory results. System testing was also done manually using the Big Bang method. Due to time constraints the Big Bang method was the only viable method to complete the amount of testing needed.

User Story #688 - Login for Coach/Admin

System Tests

- IT688_01: To test the successful sequence for an administrator to log in.
- IT688_02: To test the successful sequence for a coach to log in.
- IT688_03: To test the failed sequence for an administrator to log in.
- IT688_04: To test the failed sequence for a coach to log in.

Subsystem Tests

- UC688_01: To test a failed login to RegSys due to empty fields
- UC688_02: To test a failed login to RegSys due to invalid credentials.
- UC688_03: To test a failed login to RegSys due to invalid password.
- UC688_04: To test a successful login to RegSys as a coach.
- UC688_05: To test a successful login to RegSys as an admin

User Story #676 - Create account for coach

System Tests

- IT676_01: To test the successful sequence for a coach to create an account.
- IT676_02: To test the failed sequence for a coach to create an account

Subsystem Tests

- UC688_01: To test a failed registration attempt on RegSys with empty fields.
- UC688_02: To test a failed registration attempt on RegSys with an invalid email.
- UC688_03: To test a failed registration attempt on RegSys with non-matching passwords.
- UC688_04: To test a successful registration attempt on RegSys with valid input.

User Story #681 - View Participants as Administrators

System Tests

- IT681_01: To test the successful sequence for an administrator to view participants
- IT681_02: To test the failed sequence for an administrator to view participants.

Subsystem Tests

- UC681_01: To test the display of all current participants.
- UC681_02: To test the display of all current participants by adding a new participant.
- UC681_03: To test the display of all current participants by removing an existing participant.

User Story #685 - View List of Schools as Administrators**System Tests**

- IT685_01: To test the successful sequence for an administrator to view schools.
- IT685_02: To test the failed sequence for an administrator to view schools.

Subsystem Tests

- UC685_01: To test the display of all current schools.
- UC685_02: To test the display of all current schools by adding a new school.
- UC685_03: To test the display of all current school by removing an existing school.

User Story #694 - View Active Contests as Administrators**System Tests**

- IT694_01: To test the successful sequence for an administrator to view active contests.
- IT694_02: To test the failed sequence for an administrator to view active contests.

Subsystem Tests

- UC694_01: To test the display of all active contests.
- UC694_02: To test the display of all current active contests by adding a new active contest.
- UC694_03: To test the display of all current active contests by removing an active contest.

User Story #695 - View Stored Contests as Administrators**System Tests**

- IT695_01: To test the successful sequence for an administrator to view stored contests.
- IT695_02: To test the failed sequence for an administrator to view stored contests.

Subsystem Tests

- UC695_01: To test the display of all stored contests.
- UC695_02: To test the display of all current stored contests by adding a new stored contest.
- UC695_03: To test the display of all current stored contests by removing a stored contest.

User Story #696 - Create Contest

System Tests

- IT696_01: To test the successful sequence for an administrator to create a contest.
- IT696_02: To test the failed sequence for an administrator to create a contest.

Subsystem Tests

- UC696_01: To test a failed contest creation on RegSys with an invalid date.
- UC696_02: To test a successful contest creation on RegSys with a valid date.

User Story #680 - Create an Administrator Account

System Tests

- IT680_01: To test the successful sequence for an administrator to create an administrator account.
- IT680_02: To test the failed sequence for an administrator to create an administrator account.

Subsystem Tests

- UC680_01: To test a failed admin creation on RegSys with empty fields.
- UC680_02: To test a failed admin creation on RegSys with an invalid email.
- UC680_03: To test a failed admin creation on RegSys with non-matching passwords
- UC680_04: To test a successful admin creation on RegSys with valid input.

User Story #686 - Store Contest Data

System Tests

- IT686_01: To test the successful sequence for an administrator to store a contest.
- IT686_02: To test the failed sequence for an administrator to store a contest.

Subsystem Tests

- UC686_01: To test the storing function of RegSys by setting a contest to inactive.
- UC686_02: To test the storing function of RegSys by calling the store function twice.

User Story #709 - Remove Coach

System Tests

- IT709_01: To test the successful sequence for an administrator to remove a coach
- IT709_02: To test the failed sequence for an administrator to remove a coach.

Subsystem Tests

- UC709_01: To test the removal of coaches by pressing the delete button.
- UC709_02: To test the removal of coaches by attempting to delete a nonexistent coach.

User Story #727 - Remove School

System Tests

- IT727_01: To test the successful sequence for an administrator to log in and remove a school.
- IT727_02: To test the failed sequence for an administrator to log in and remove a school.

Subsystem Tests

- UC727_01: To test the successful deletion of a school that exists.
- UC727_02: To test the failed deletion of a school that does not exist.

User Story #678 - Coach Modifies

System Tests

- IT678_01: To test the successful modification of a participant on a team
- IT678_02: To test the successful modification of a participant on a team
- IT678_03: To test the successful modification of a participant on a team
- IT678_04: To test the failed modification of a participant on a team
- IT678_05: To test the failed modification of a participant on a team
- IT678_06: To test the failed modification of a participant on a team

Subsystem Tests

- UC678_01: To test the successful sequence for a coach to log in, create a team and modify that team's participants.
- UC678_02: To test the successful sequence for a coach to log in, create a team and modify that team's participants.
- UC678_03: To test the successful sequence for a coach to log in, create a team and modify that team's participants.
- UC678_04: To test the failed sequence for a coach to log in, create a team and modify that team's participants.
- UC678_05: To test the failed sequence for a coach to log in, create a team and modify that team's participants.
- UC678_06: To test the failed sequence for a coach to log in, create a team and modify that team's participants.

User Story #687 - Coach Creates a Team

System Tests

- IT687_01: To test the successful sequence for a coach to log in and create a team
- IT687_02: To test the successful sequence for a coach to log in and create a team
- IT687_03: To test the successful sequence for a coach to log in and create a team
- IT687_04: To test the failed sequence for a coach to log in and create a team
- IT687_05: To test the failed sequence for a coach to log in and create a team
- IT687_06: To test the failed sequence for a coach to log in and create a team

Subsystem Tests

- UC687_01: To test the successful creation of a team.
- UC687_02: To test the successful creation of a team.
- UC687_03: To test the successful creation of a team.
- UC687_04: To test the failed creation of a team.
- UC687_05: To test the failed creation of a team.
- UC687_06: To test the failed creation of a team.

User Story #679 - Coach Adds Participant to Team

System Tests

- IT679_01: To test the successful sequence for a coach to log in, create a team and adding a participant.
- IT679_02: To test the successful sequence for a coach to log in, create a team and adding a participant.

- IT679_03: To test the successful sequence for a coach to log in, create a team and adding a participant.
- IT679_04: To test the failed sequence for a coach to log in, create a team and adding a participant.
- IT679_05: To test the failed sequence for a coach to log in, create a team and adding a participant.
- IT679_06: To test the failed sequence for a coach to log in, create a team and adding a participant.

Subsystem Tests

- UC679_01: To test the successful addition of a participant to a team.
- UC679_02: To test the successful addition of a participant to a team.
- UC679_03: To test the successful addition of a participant to a team.
- UC679_04: To test the failed addition of a participant to a team.
- UC679_05: To test the failed addition of a participant to a team.
- UC679_06: To test the failed addition of a participant to a team.

User Story #677 - Coach Modifies Account

System Tests

- IT677_01: To test the successful sequence for a coach to log in and modify his account.
- IT677_02: To test the successful sequence for a coach to log in and modify his account
- IT677_03: To test the successful sequence for a coach to log in and modify his account
- IT677_04: To test the failed sequence for a coach to log in and modify his account
- IT677_05: To test the failed sequence for a coach to log in and modify his account
- IT677_06: To test the failed sequence for a coach to log in and modify his account

Subsystem Tests

- UC677_01: To test the successful modification of a coach's account.
- UC677_02: To test the successful modification of a coach's account.
- UC677_03: To test the successful modification of a coach's account.
- UC677_04: To test the failed modification of a coach's account.
- UC677_05: To test the failed modification of a coach's account.
- UC677_06: To test the failed modification of a coach's account.

User Story #684 - Administrator Views Teams

System Tests

- IT684_01: To test the successful sequence for an administrator to log in and view teams
- IT684_02: To test the successful sequence for an administrator to log in and view teams
- IT684_03: To test the successful sequence for an administrator to log in and view teams
- IT684_04: To test the failed sequence for an administrator to log in and view teams
- IT684_05: To test the failed sequence for an administrator to log in and view teams
- IT684_06: To test the failed sequence for an administrator to log in and view teams

Subsystem Tests

- UC684_01: To test the successful display of teams.
- UC684_02: To test the successful display of teams.
- UC684_03: To test the successful display of teams.
- UC684_04: To test the failed display of teams.
- UC684_05: To test the failed display of teams.
- UC684_06: To test the failed display of teams.

User Story #683 - Administrator Views Administrators

System Tests

- IT683_01: To test the successful sequence for an administrator to log in and view other administrators.
- IT683_02: To test the successful sequence for an administrator to log in and view other administrators.
- IT683_03: To test the successful sequence for an administrator to log in and view other administrators.
- IT683_04: To test the failed sequence for an administrator to log in and view other administrators.
- IT683_05: To test the failed sequence for an administrator to log in and view other administrators.

Subsystem Tests

- UC683_01: To test the successful display of administrators.
- UC683_02: To test the successful display of administrators.
- UC683_03: To test the successful display of administrators.
- UC683_04: To test the failed display of administrators.
- UC683_05: To test the failed display of administrators.
- UC683_06: To test the failed display of administrators.

User Story #718 - Administrator Deletes Administrator

System Tests

- IT718_01: To test the successful sequence for an administrator to log in, create and administrator and delete that administrator.
- IT718_02: To test the successful sequence for an administrator to log in, create and administrator and delete that administrator.
- IT718_03: To test the successful sequence for an administrator to log in, create and administrator and delete that administrator.
- IT718_04: To test the failed sequence for an administrator to log in, create and administrator and delete that administrator.
- IT718_05: To test the failed sequence for an administrator to log in, create and administrator and delete that administrator.
- IT718_06: To test the failed sequence for an administrator to log in, create and administrator and delete that administrator.

Subsystem Tests

- UC718_01: To test the successful deletion of an administrator.
- UC718_02: To test the successful deletion of an administrator.
- UC718_03: To test the successful deletion of an administrator.
- UC718_04: To test the failed deletion of an administrator.
- UC718_05: To test the failed deletion of an administrator.
- UC718_06: To test the failed deletion of an administrator.

User Story #721 - Administrator Searches for a Team

System Tests

- IT721_01: To test the successful sequence for an administrator to log in and search for a team
- IT721_02: To test the successful sequence for an administrator to log in and search for a team
- IT721_03: To test the successful sequence for an administrator to log in and search for a team
- IT721_04: To test the failed sequence for an administrator to log in and search for a team
- IT721_05: To test the failed sequence for an administrator to log in and search for a team
- IT721_06: To test the failed sequence for an administrator to log in and search for a team

Subsystem Tests

- UC721_01: To test the successful search for a team that exists

- UC721_02: To test the successful search for a team that exists
- UC721_03: To test the successful search for a team that exists
- UC721_04: To test the failed search for a team that exists
- UC721_05: To test the failed search for a team that exists
- UC721_06: To test the failed search for a team that exists

GLOSSARY

Active Contest: A competition that is currently open for registration.

Stored Contest: A competition that has already occurred.

Participant: A high school student that is a member of a team.

APPENDIX

Appendix A - UML Diagrams

Static UML Diagrams

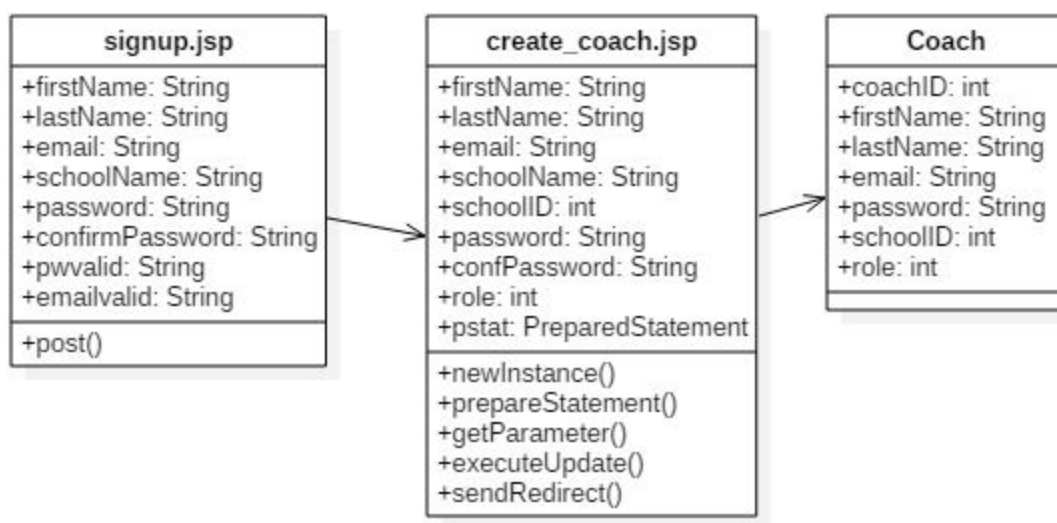


Figure #1 - User Story #676 Class Diagram

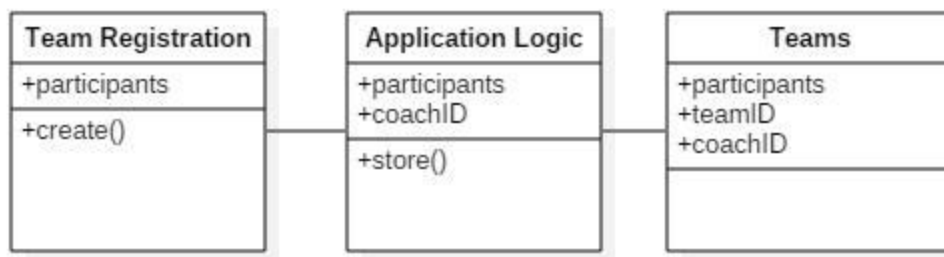


Figure #2 - User Story #678 Class Diagram

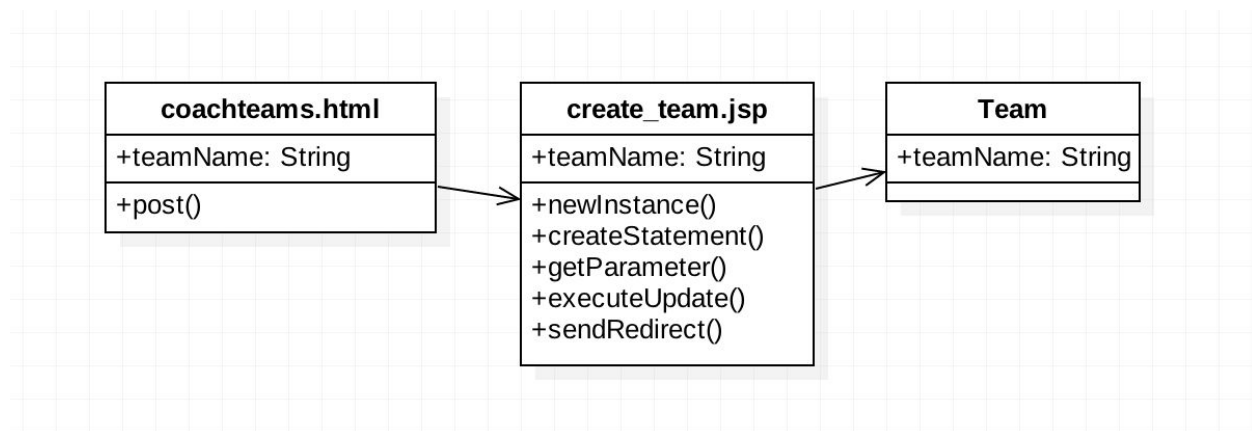


Figure #3 - User Story #687 Class Diagram

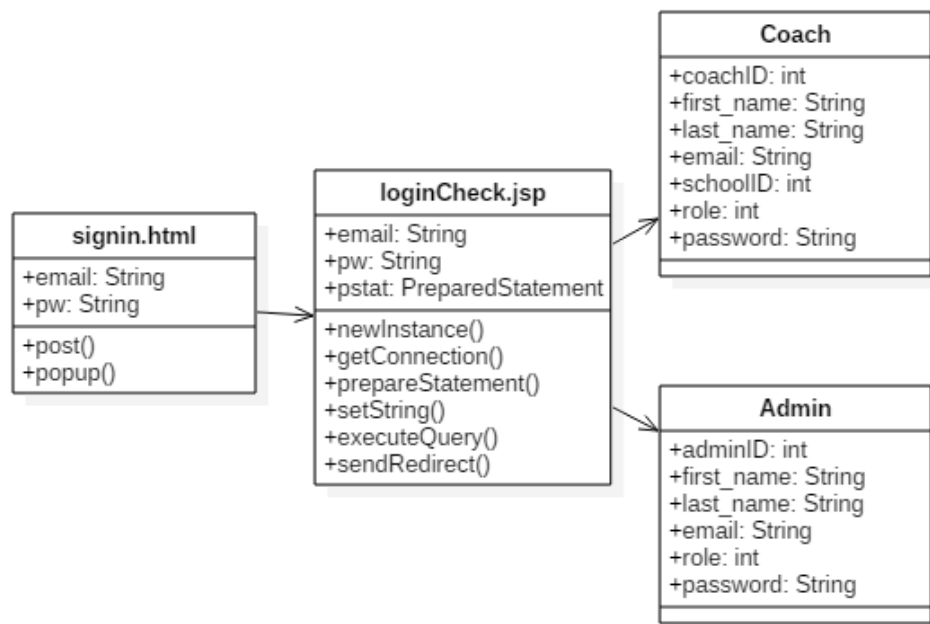


Figure #4 - User Story #688 Class Diagram

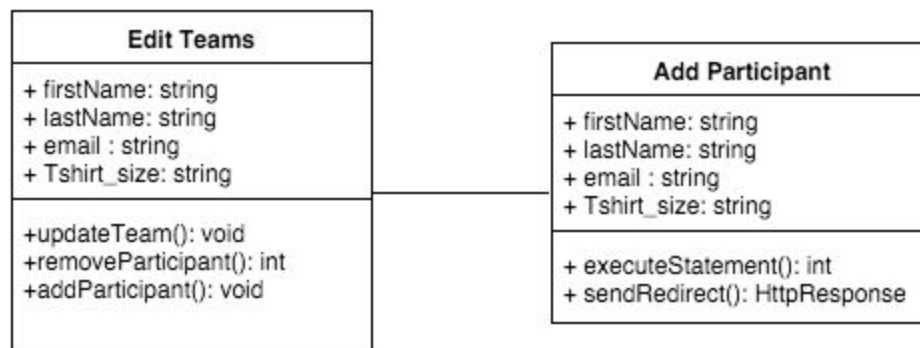


Figure #5 - User Story #679 Class Diagram

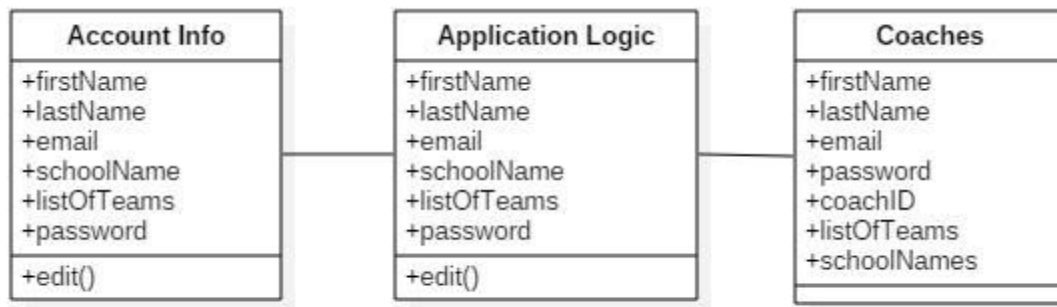


Figure #6 - User Story #677 Class Diagram

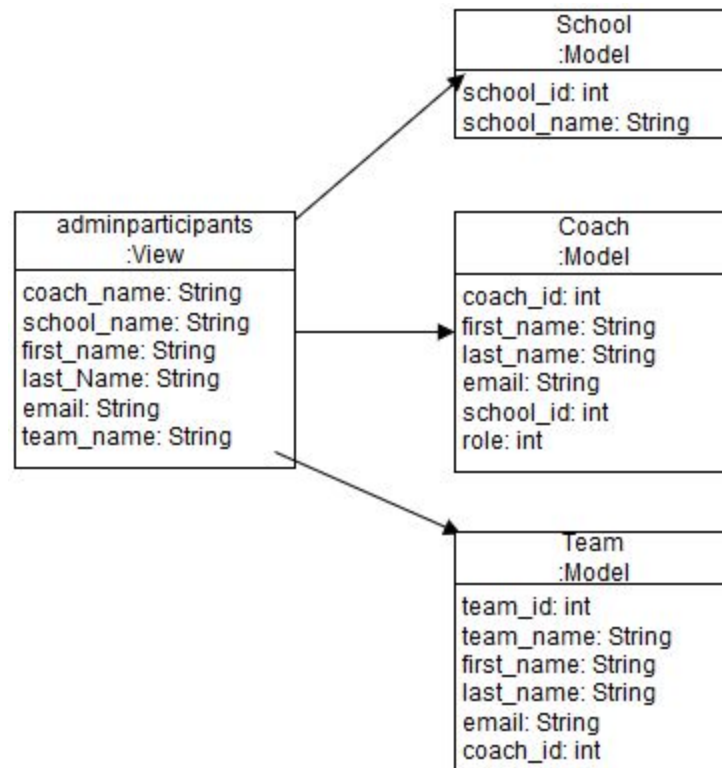


Figure #7 - User Story #681 Class Diagram

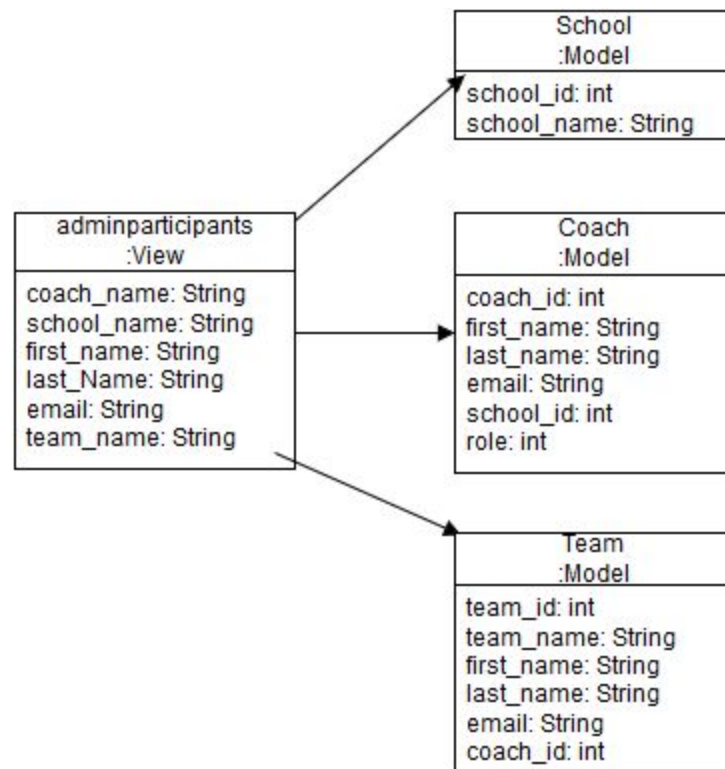


Figure #8 - User Story #685 Class Diagram

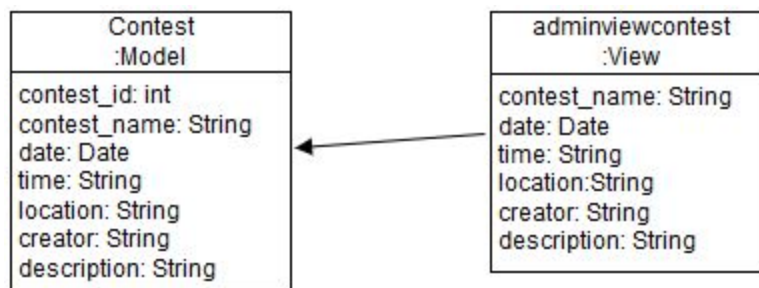


Figure #9 - User Story #694 Class Diagram

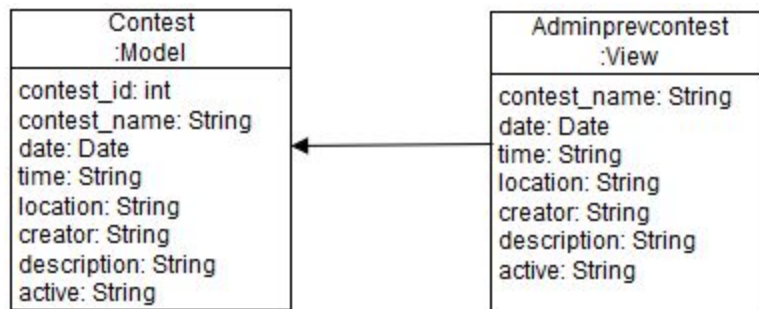


Figure #10 - User Story #695 Class Diagram

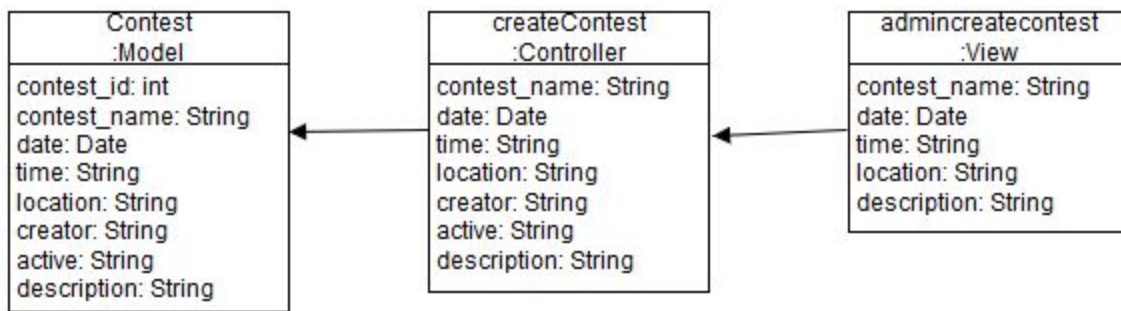


Figure #11 - User Story #696 Class Diagram

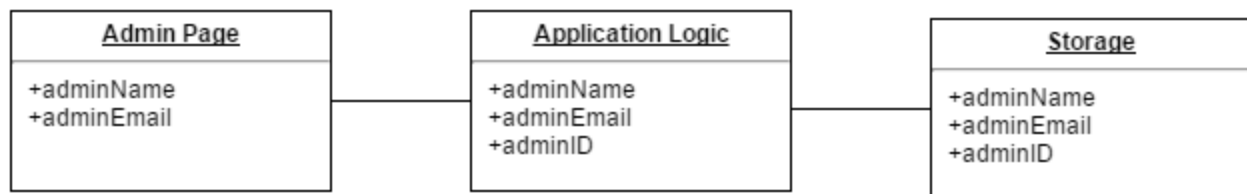


Figure #12 - User Story #683 Class Diagram

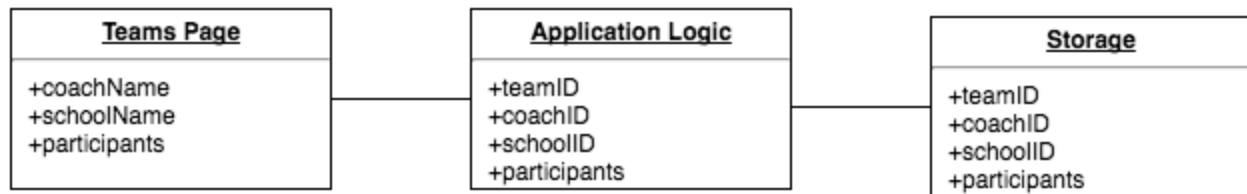


Figure #13 - User Story #684 Class Diagram



Figure #14 - User Story #718 Class Diagram

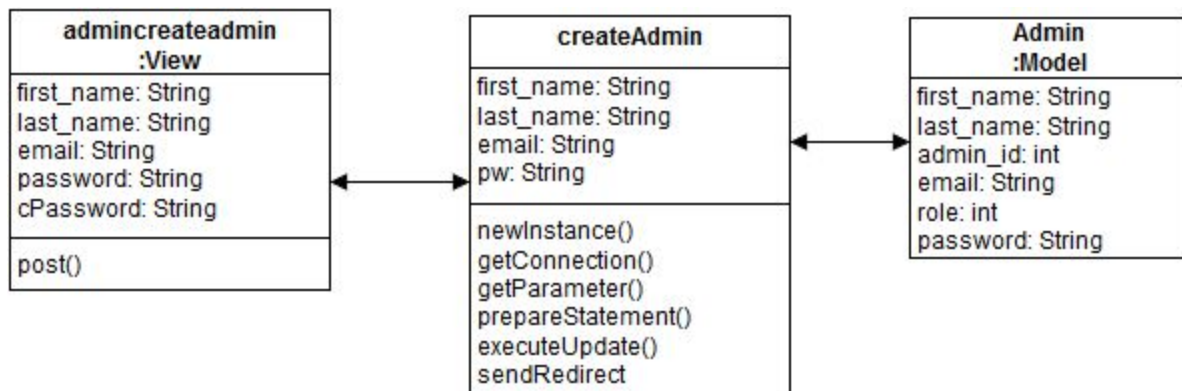


Figure #15 - User Story #680 Class Diagram

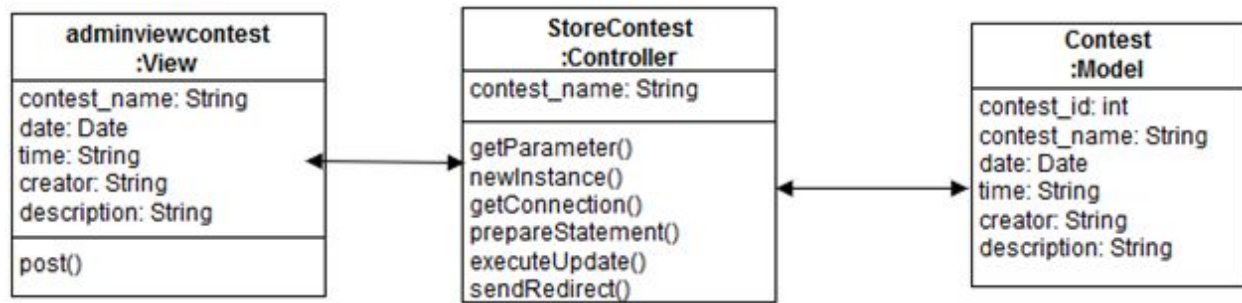


Figure #16 - User Story #686 Class Diagram

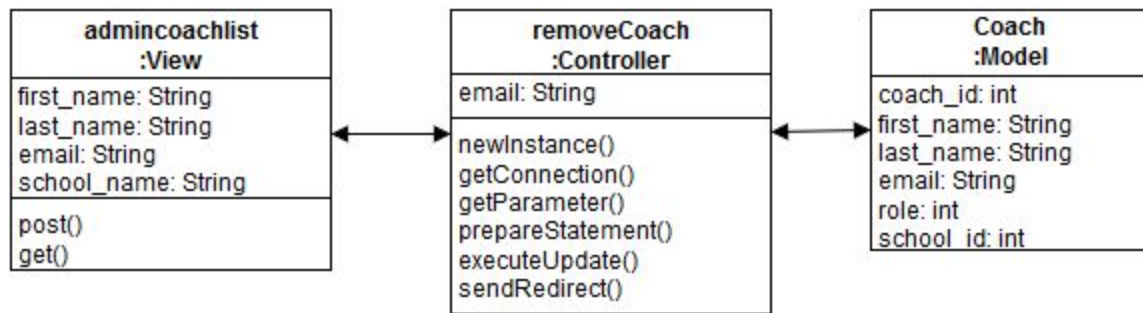


Figure #17 - User Story #709 Class Diagram

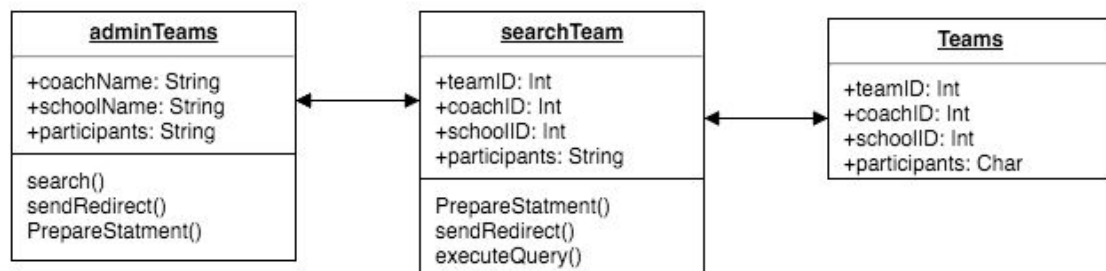


Figure #18 - User Story #721 Class Diagram

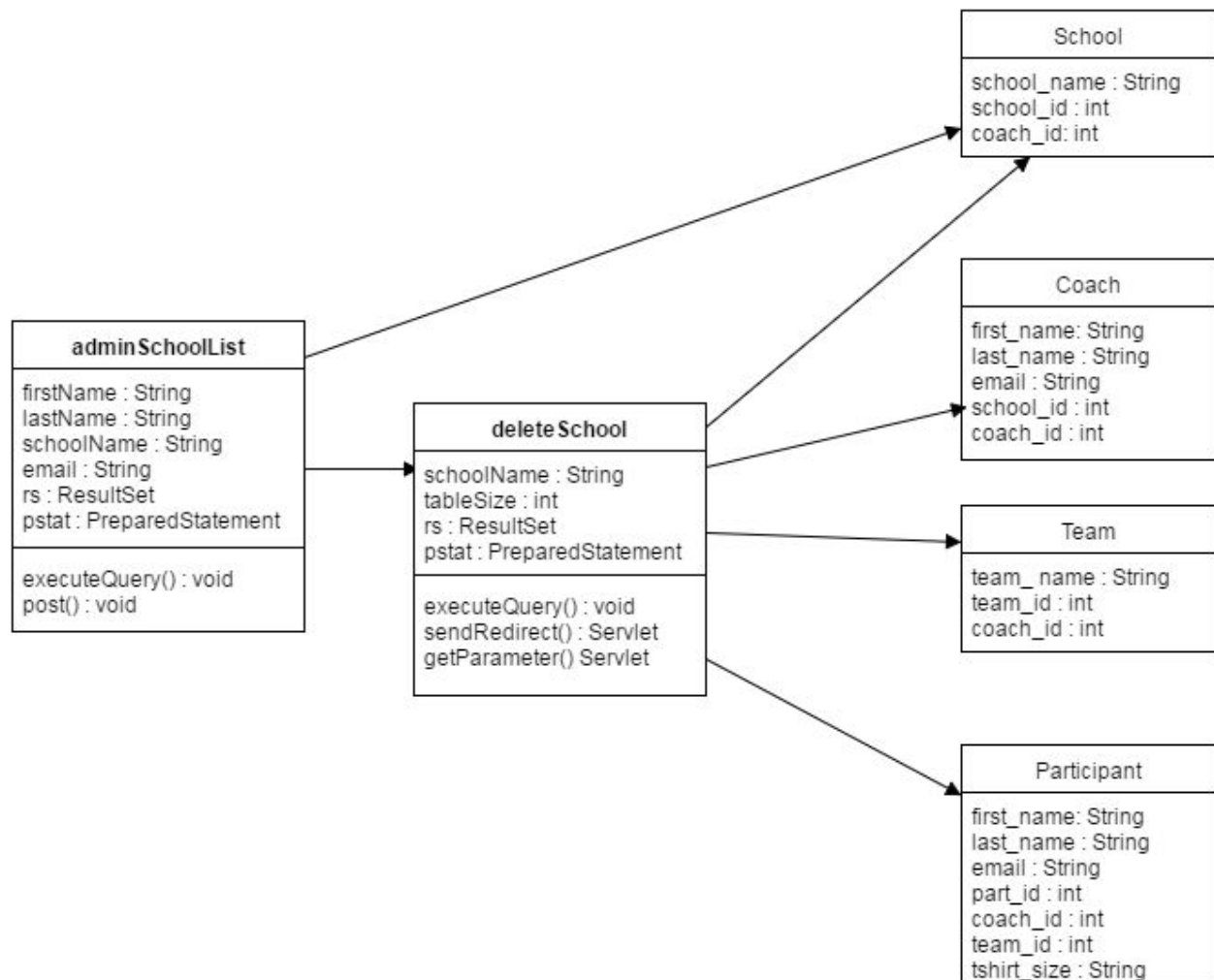


Figure #19 - User Story #727 Class Diagram

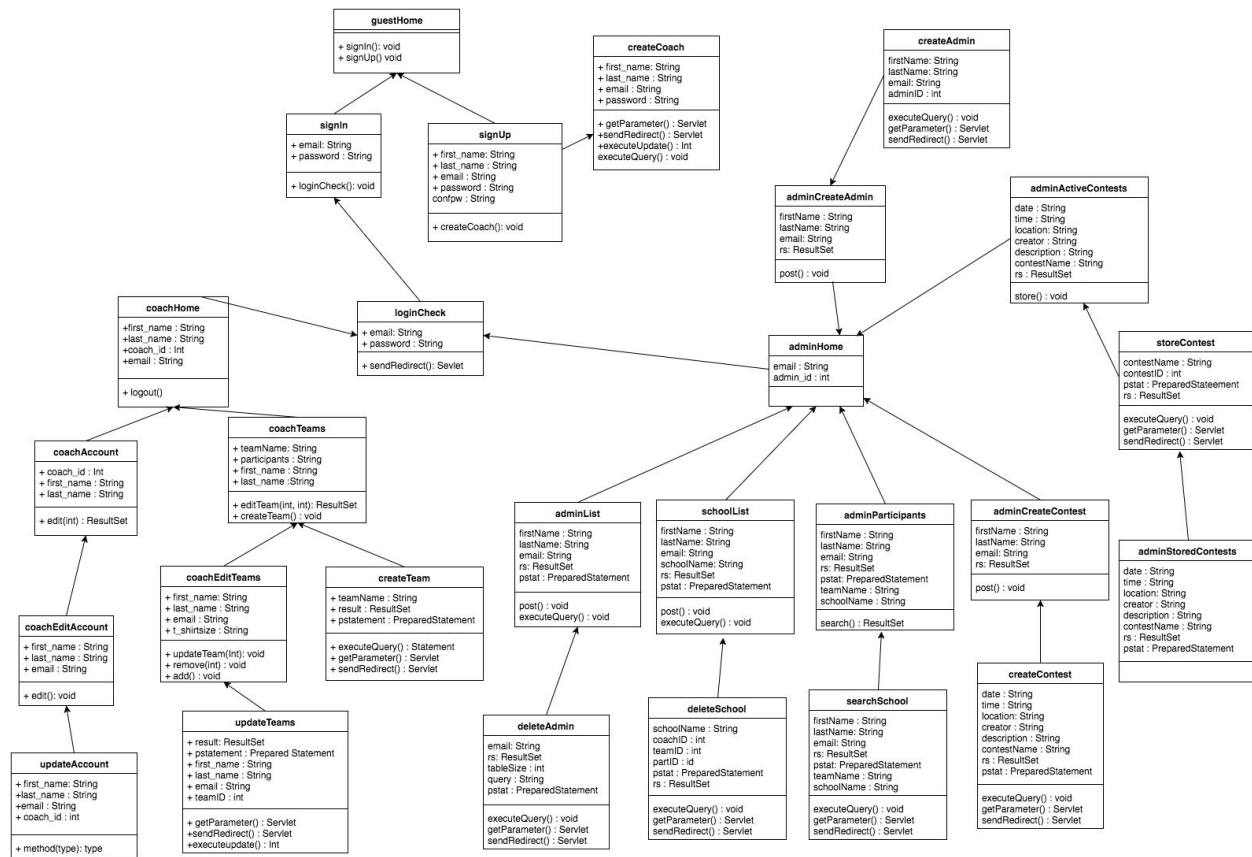


Figure #20 - Object Diagram

Dynamic UML Diagrams

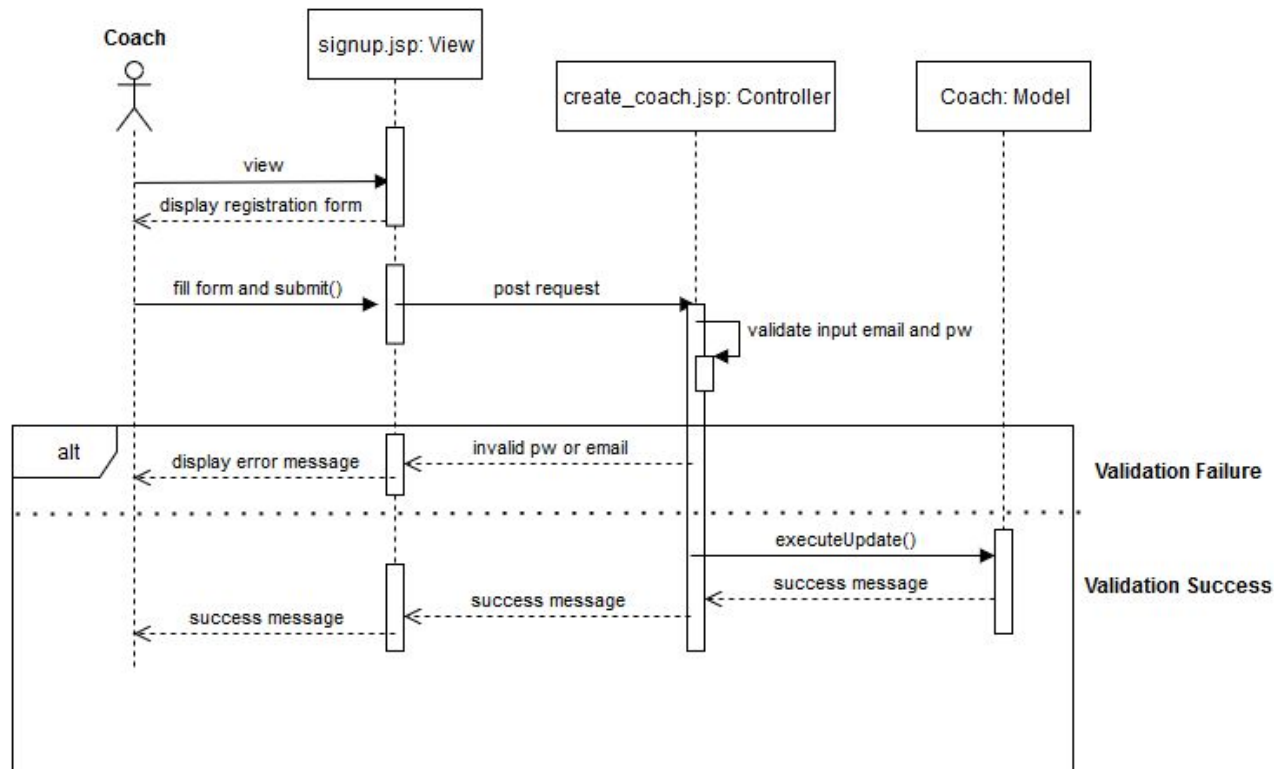


Figure #1 - User Story #676 Sequence Diagram

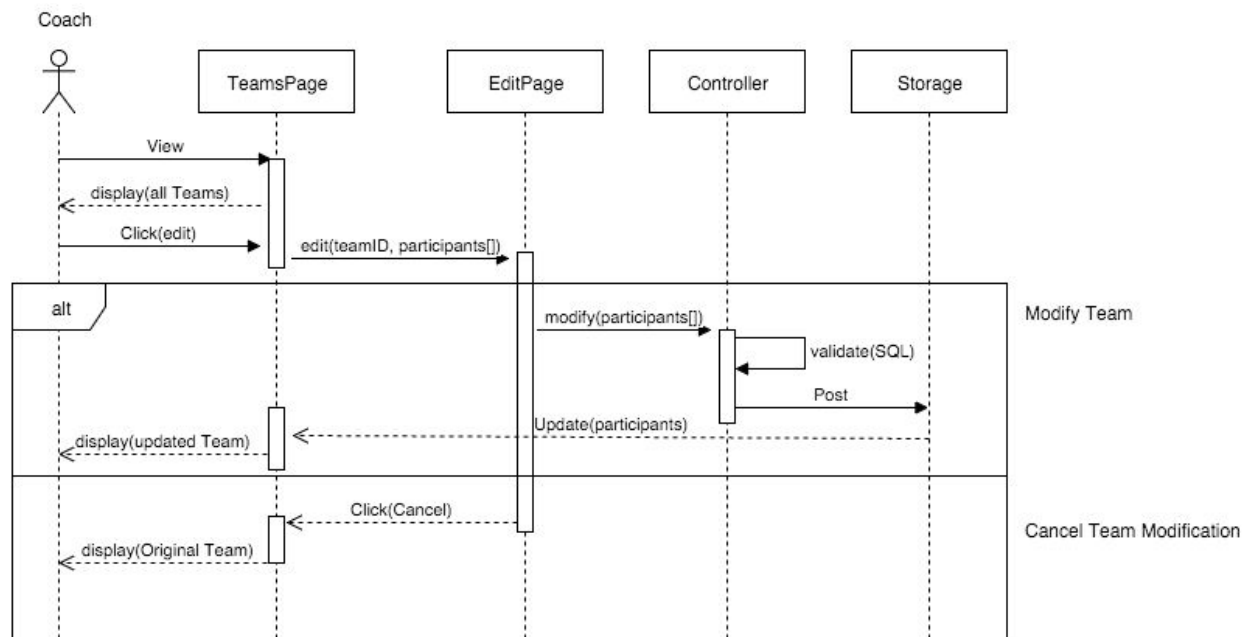


Figure #2 - User Story #678 Sequence Diagram

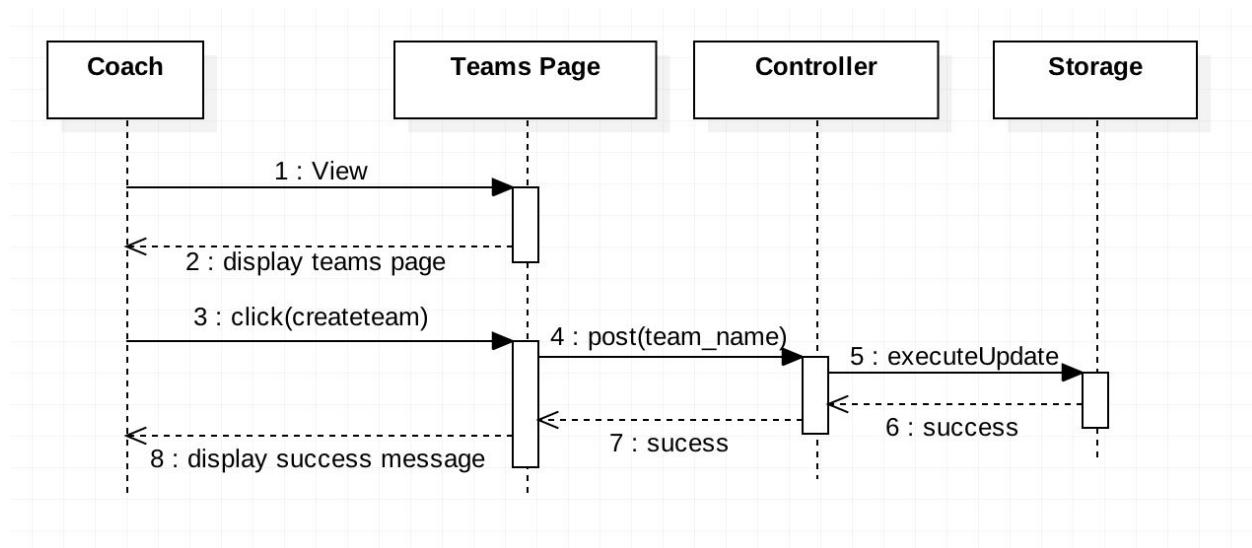


Figure #3 - User Story #687 Sequence Diagram

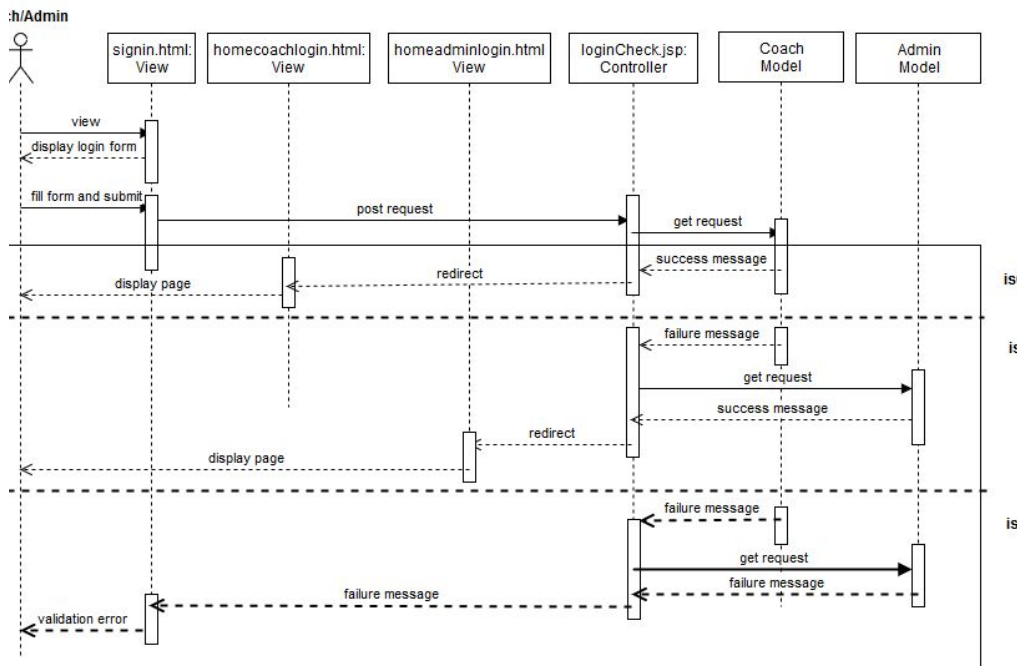


Figure #4 - User Story #688 Sequence Diagram

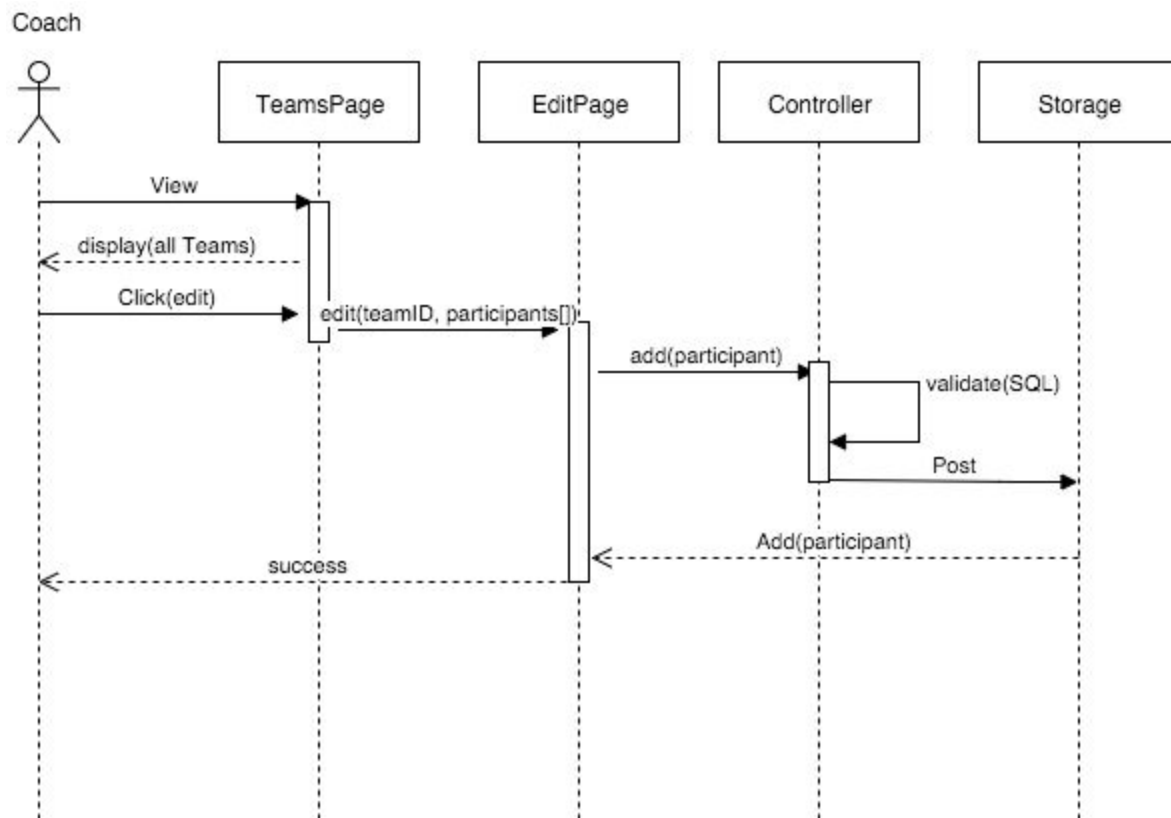


Figure #5 - User Story #679 Sequence Diagram

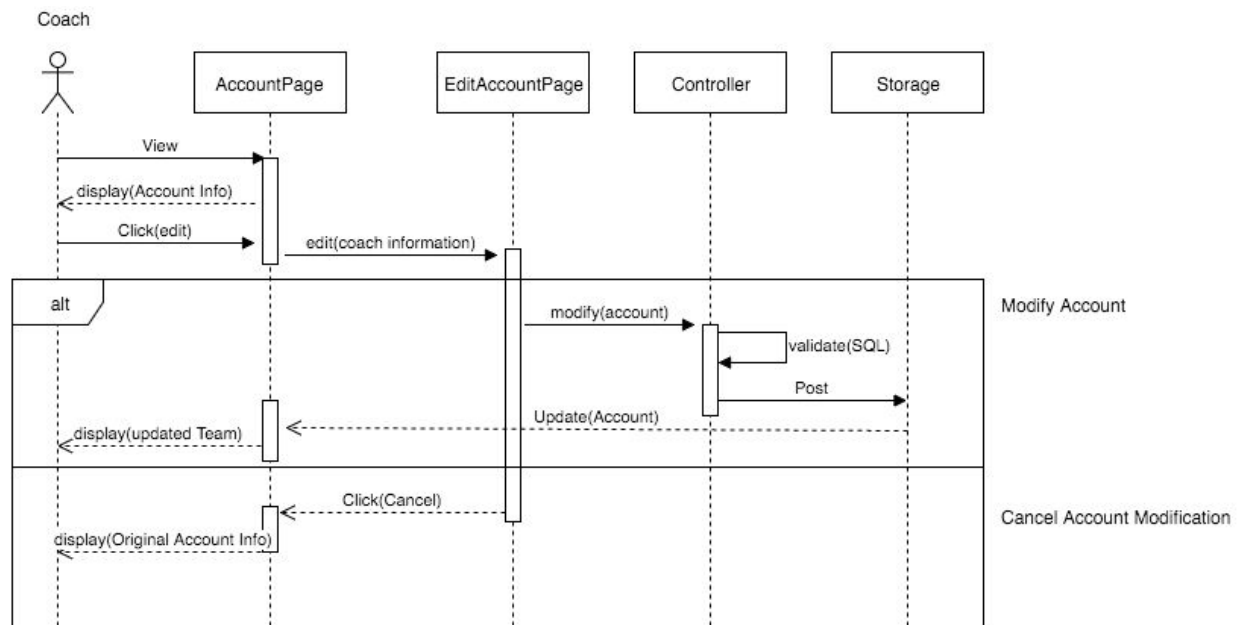


Figure #6 - User Story #677 Sequence Diagram

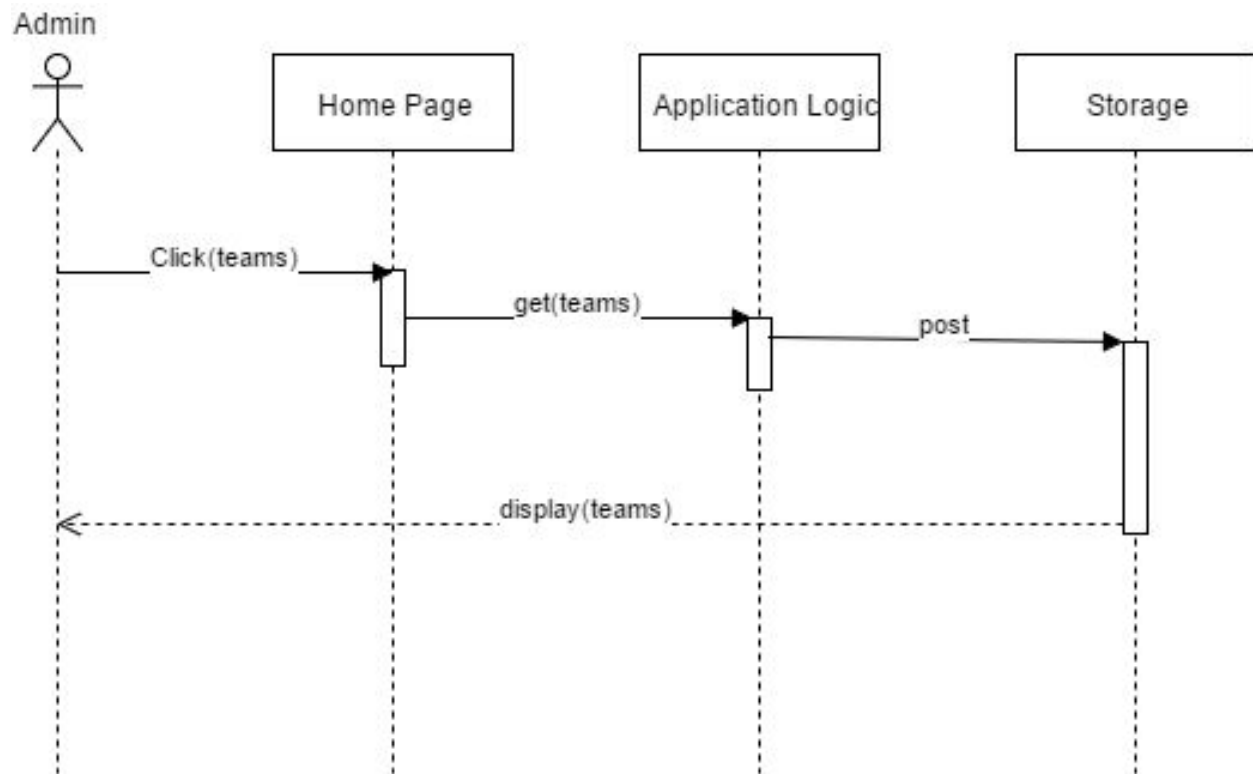


Figure #7 - User Story #681 Sequence Diagram

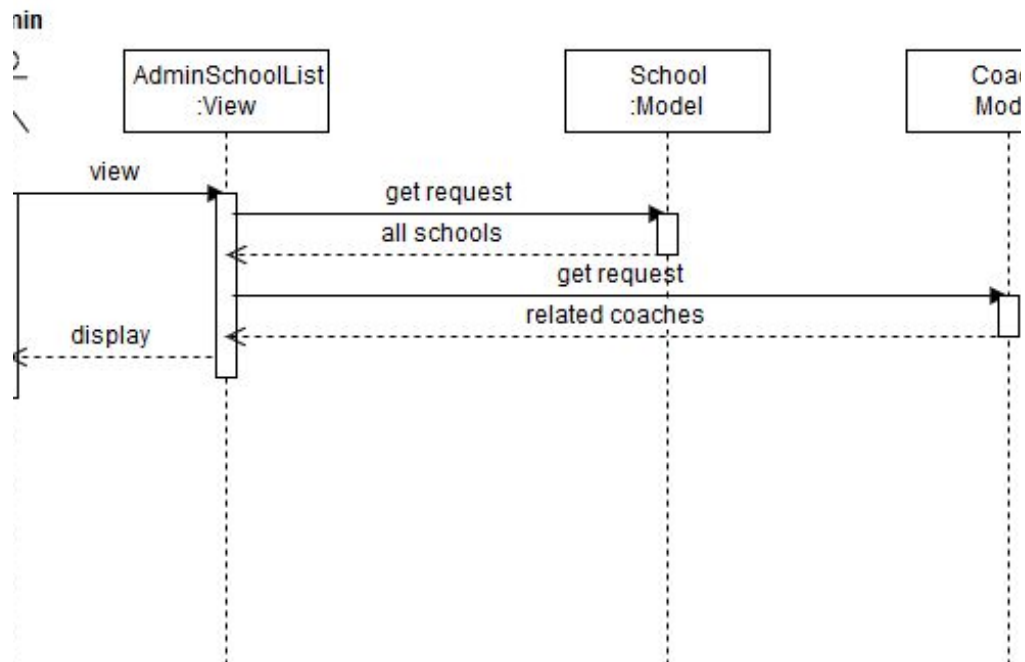


Figure #8 - User Story #685 Sequence Diagram

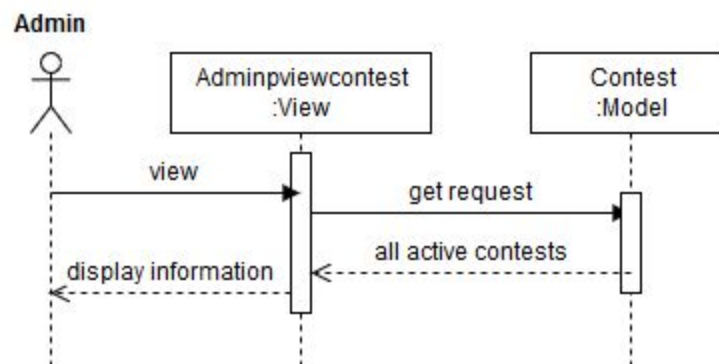


Figure #9 - User Story #694 Sequence Diagram

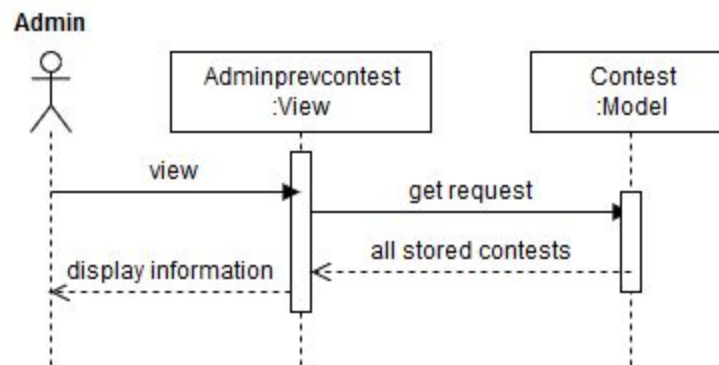


Figure #10 - User Story #695 Sequence Diagram

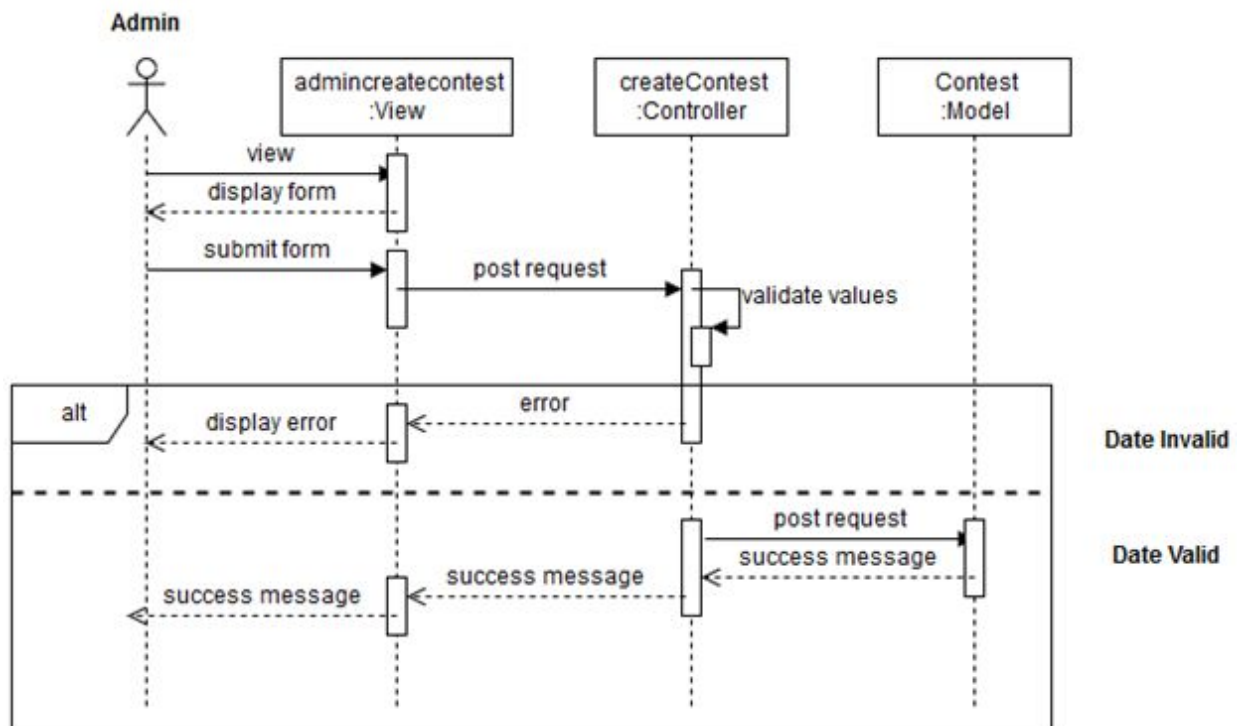


Figure #11 - User Story #696 Sequence Diagram

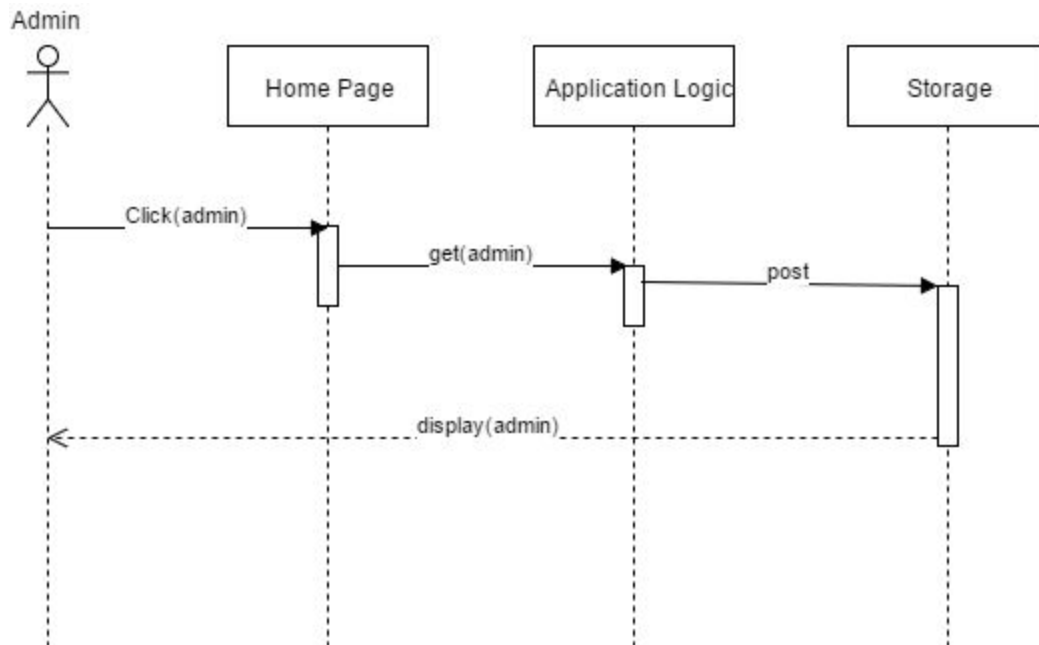


Figure #12 - User Story #683 Sequence Diagram

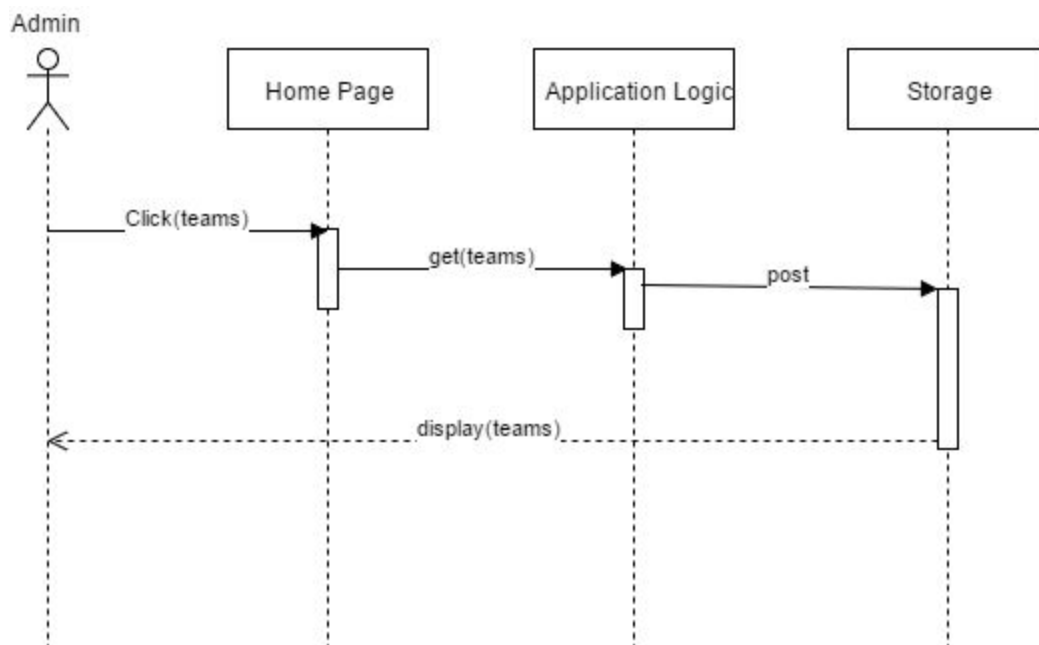


Figure #13 - User Story #684 Sequence Diagram

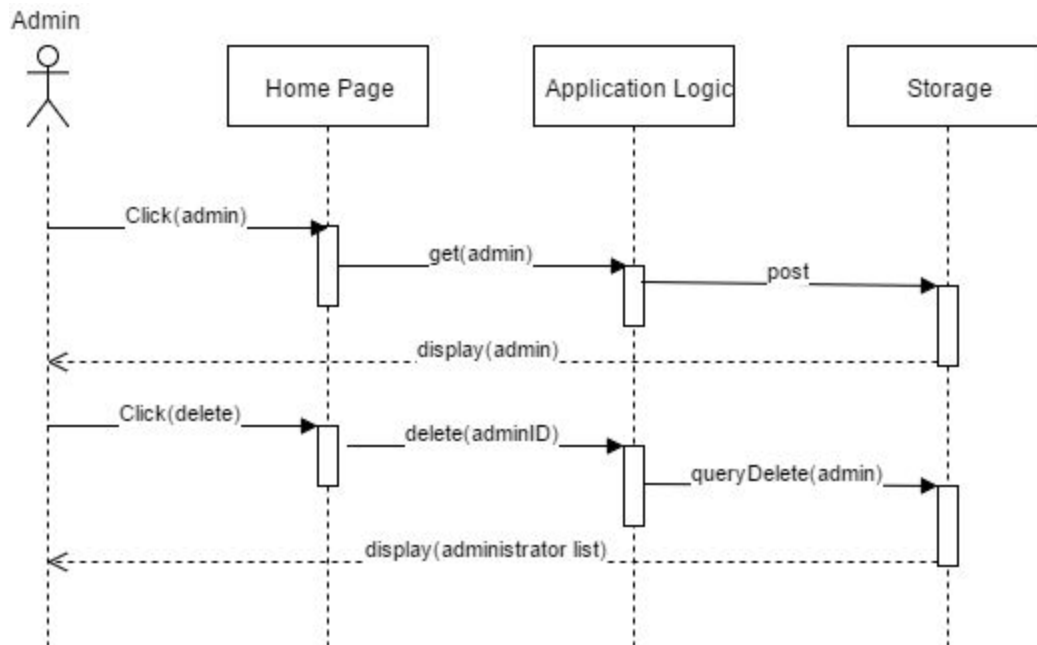


Figure #14 - User Story #718 Sequence Diagram

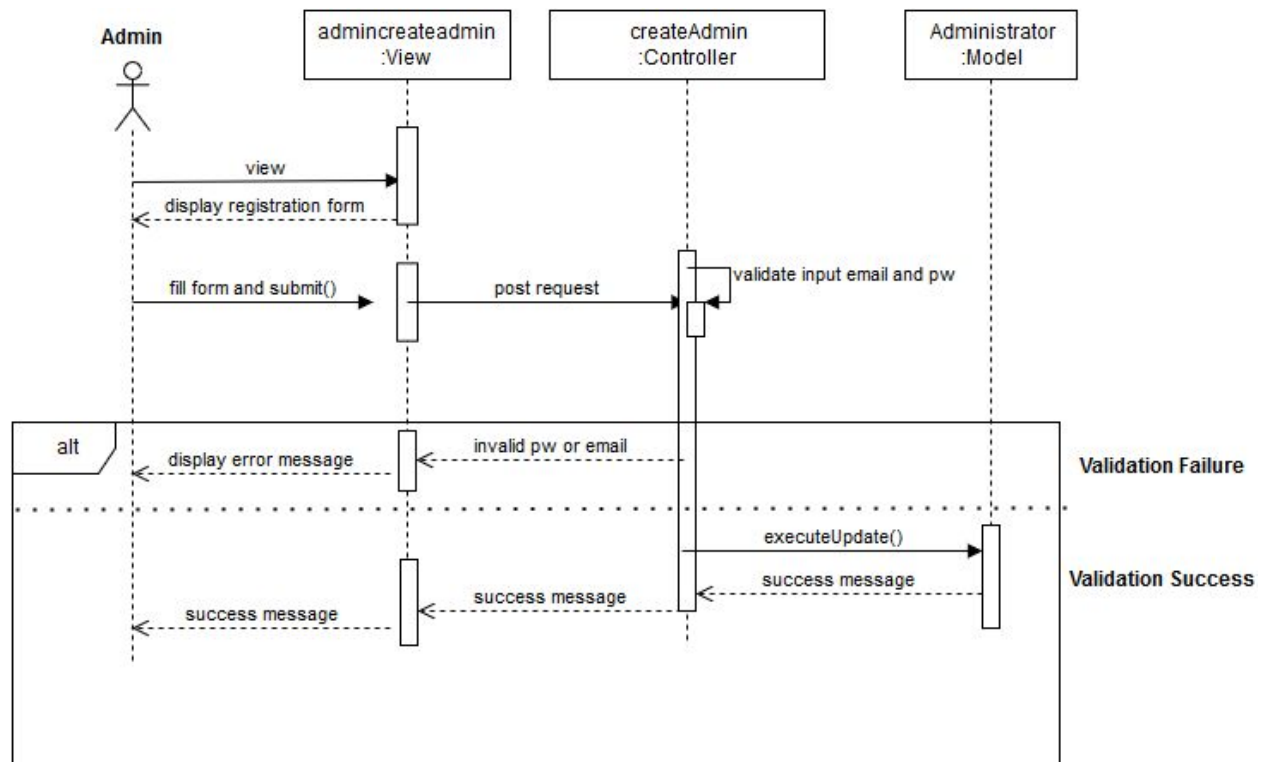


Figure #15 - User Story #680 Sequence Diagram

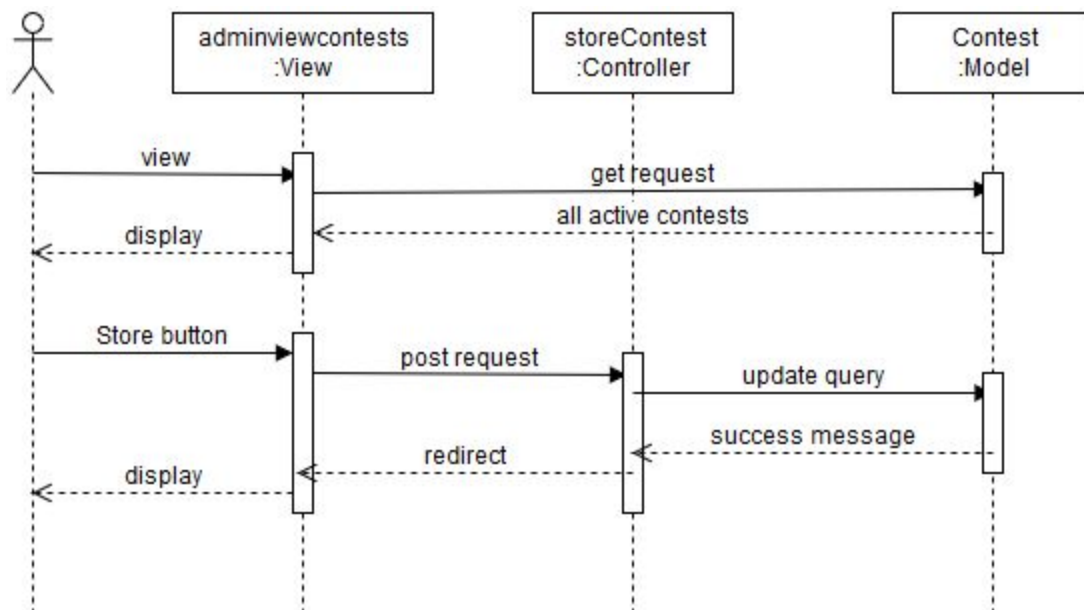


Figure #16 - User Story #686 Sequence Diagram

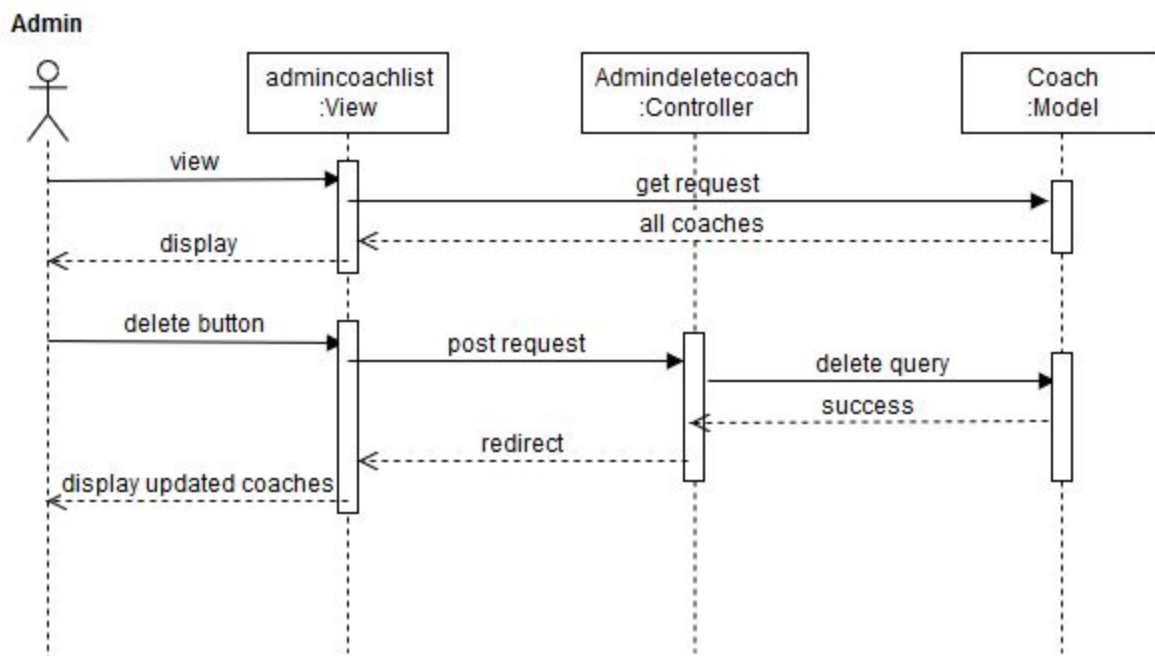


Figure #17 - User Story #709 Sequence Diagram

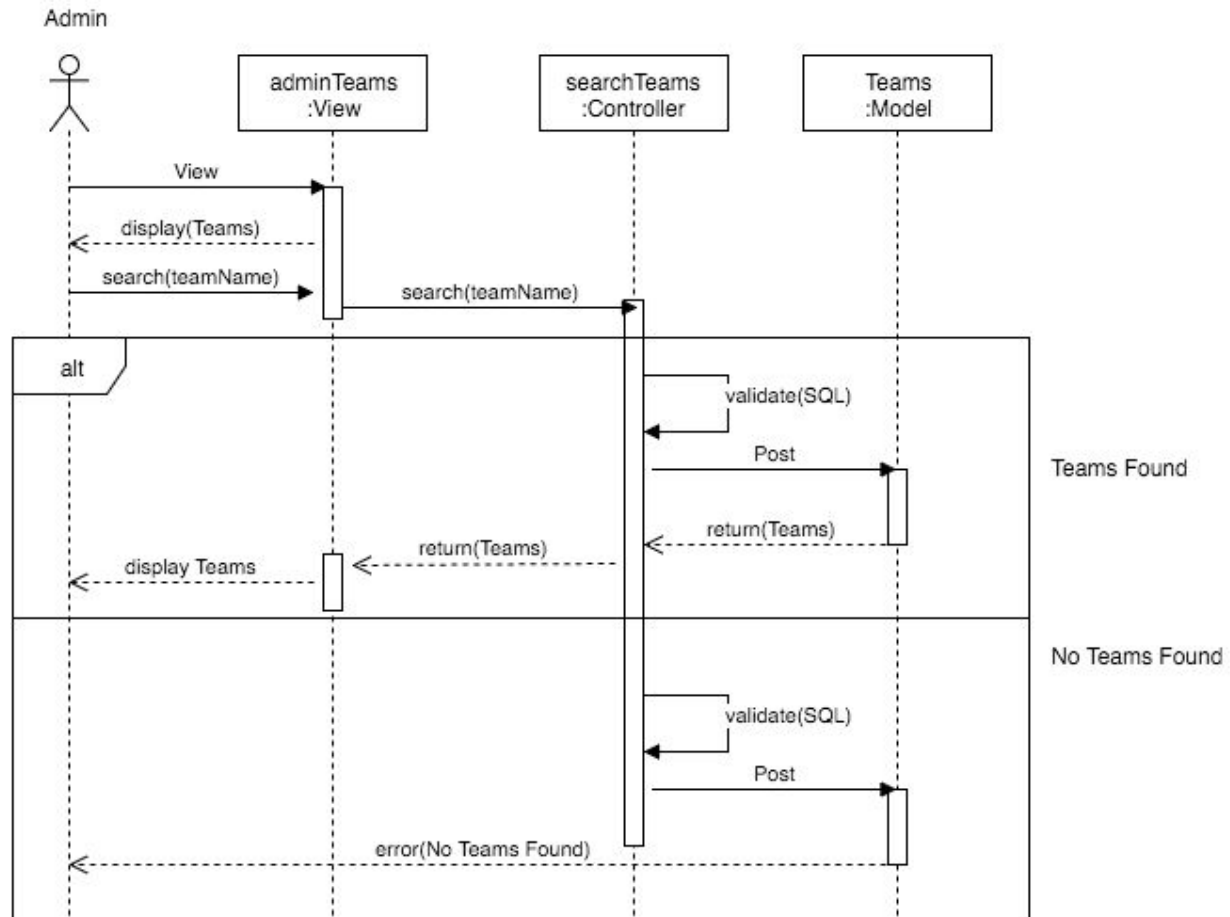


Figure #18 - User Story #721 Sequence Diagram

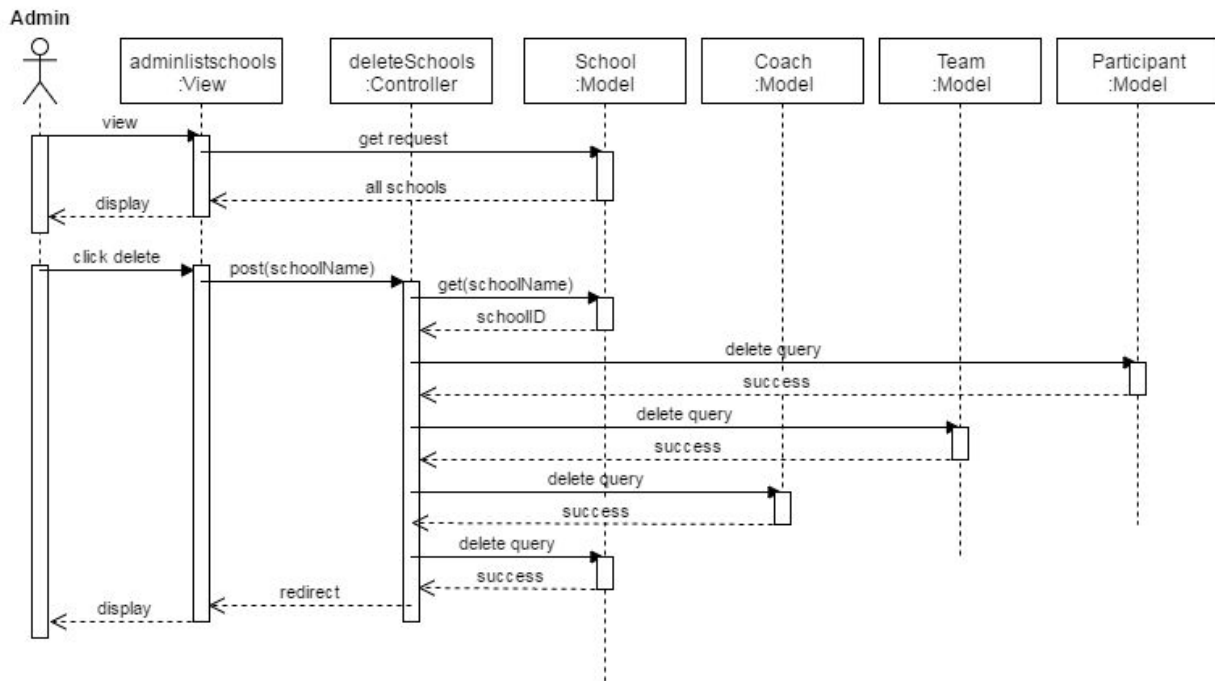


Figure #19 - User Story #727 Sequence Diagram

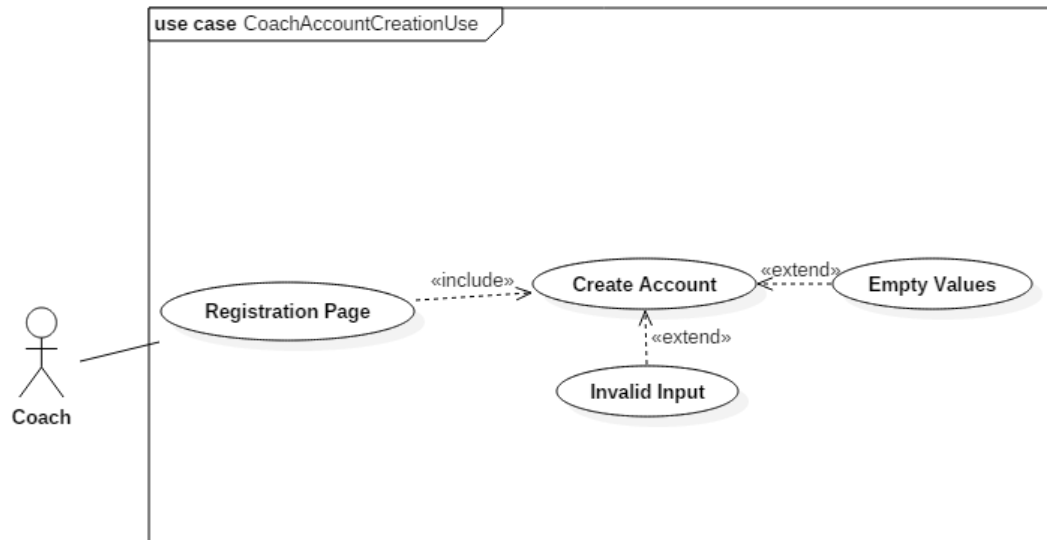


Figure #1 - User Story #676 Use Case Diagram

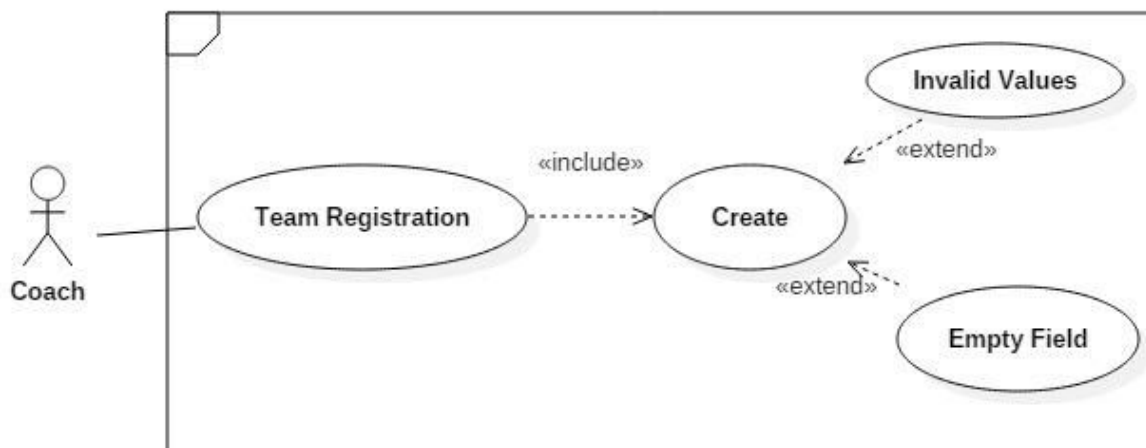


Figure #2 - User Story #678 Use Case Diagram

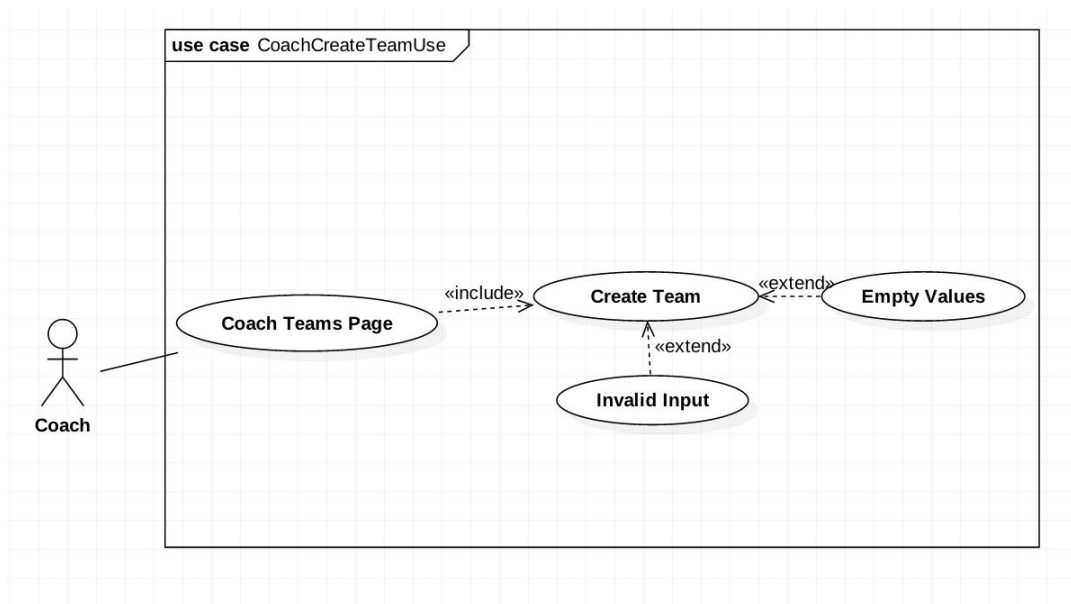


Figure #3 - User Story #687 Use Case Diagram

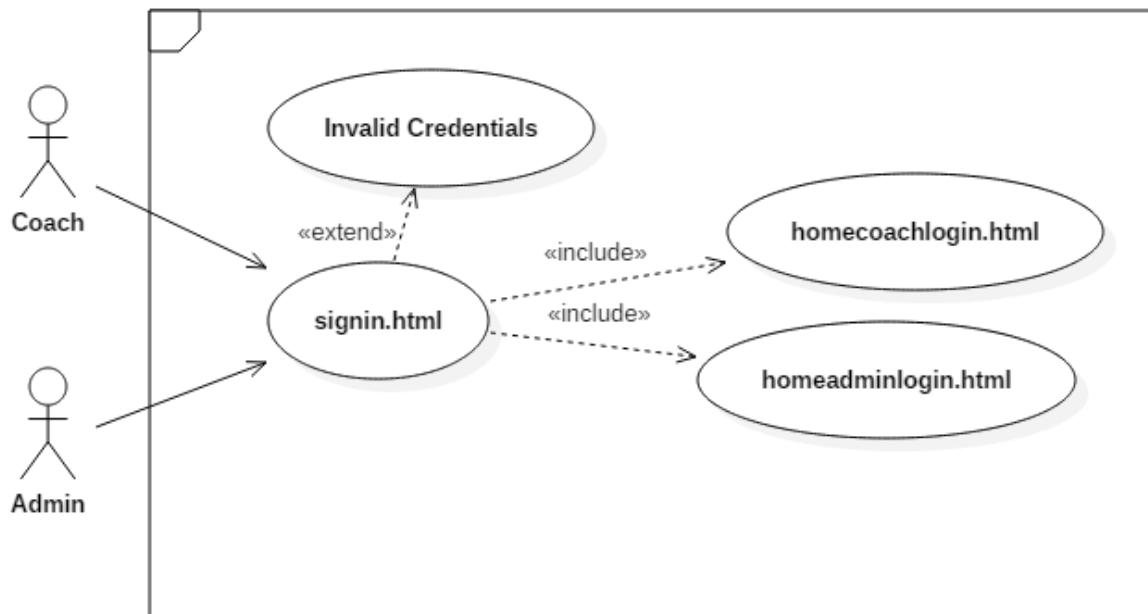


Figure #4 - User Story #688 Use Case Diagram

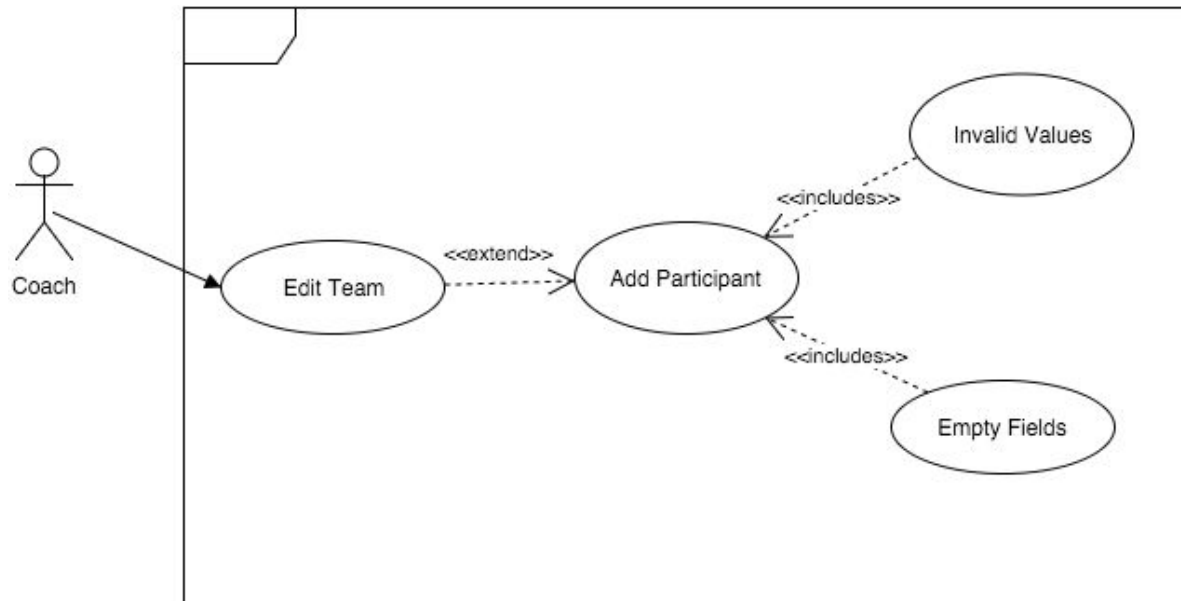


Figure #5 - User Story #679 Use Case Diagram

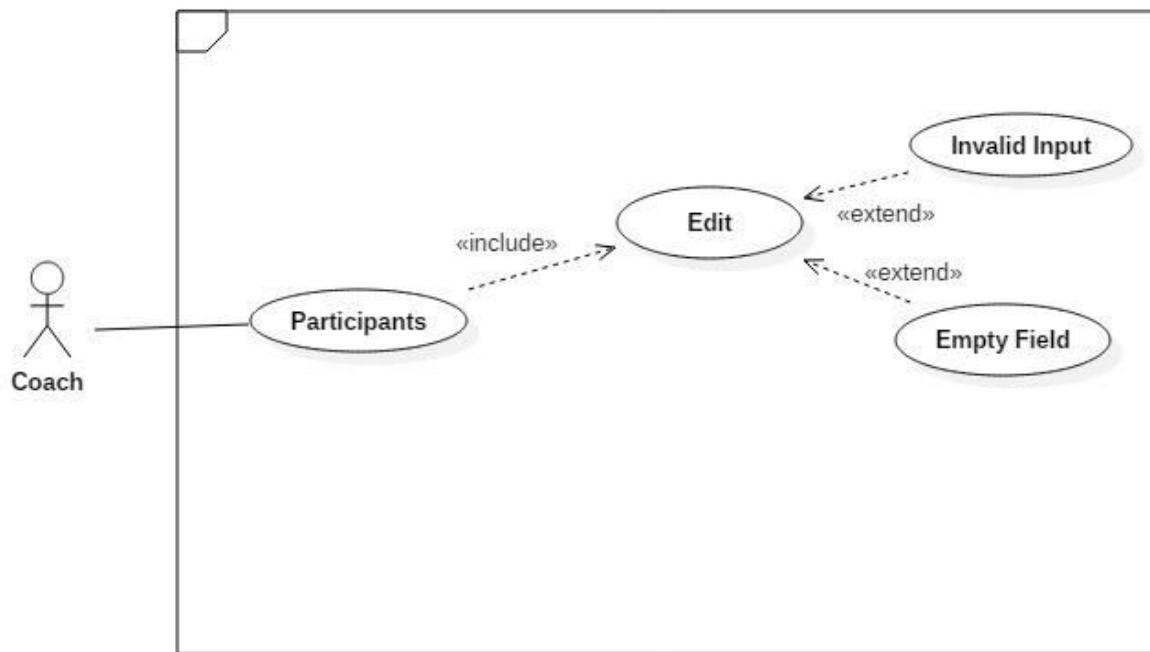


Figure #6 - User Story #677 Use Case Diagram

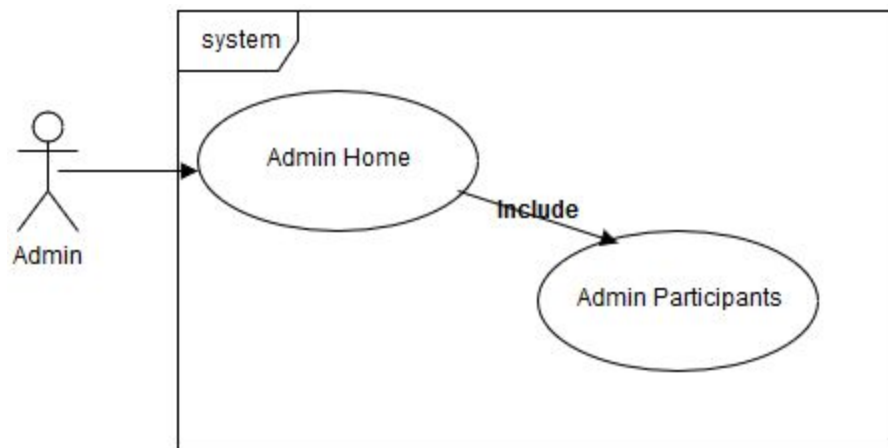


Figure #7 - User Story #681 Use Case Diagram

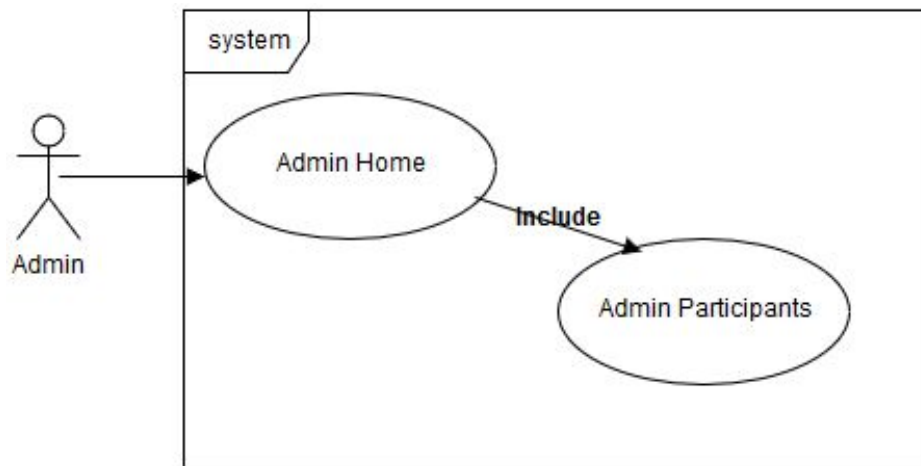


Figure #8 - User Story #685 Use Case Diagram

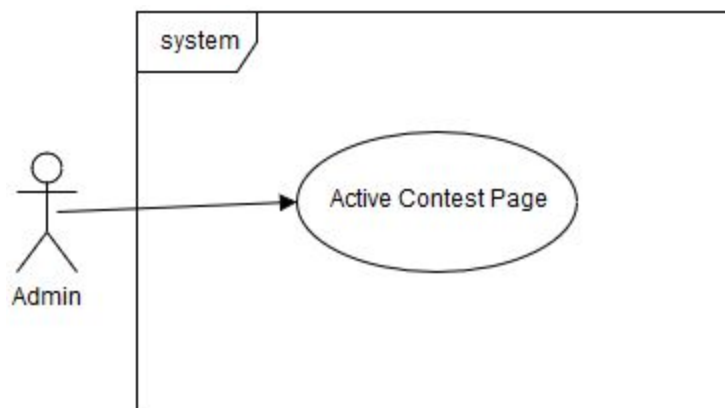


Figure #9 - User Story #694 Use Case Diagram

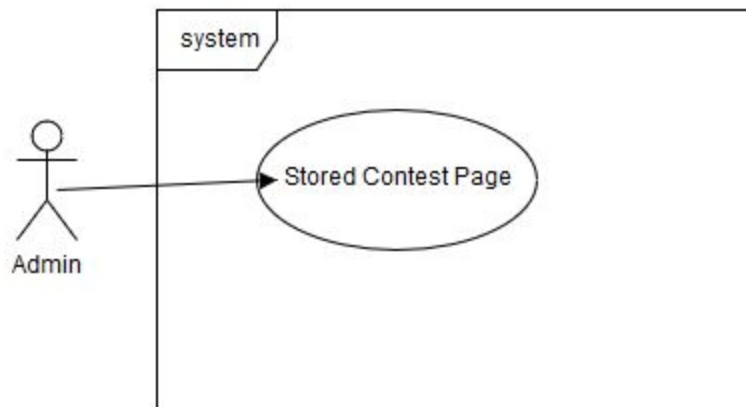


Figure #10 - User Story #695 Use Case Diagram

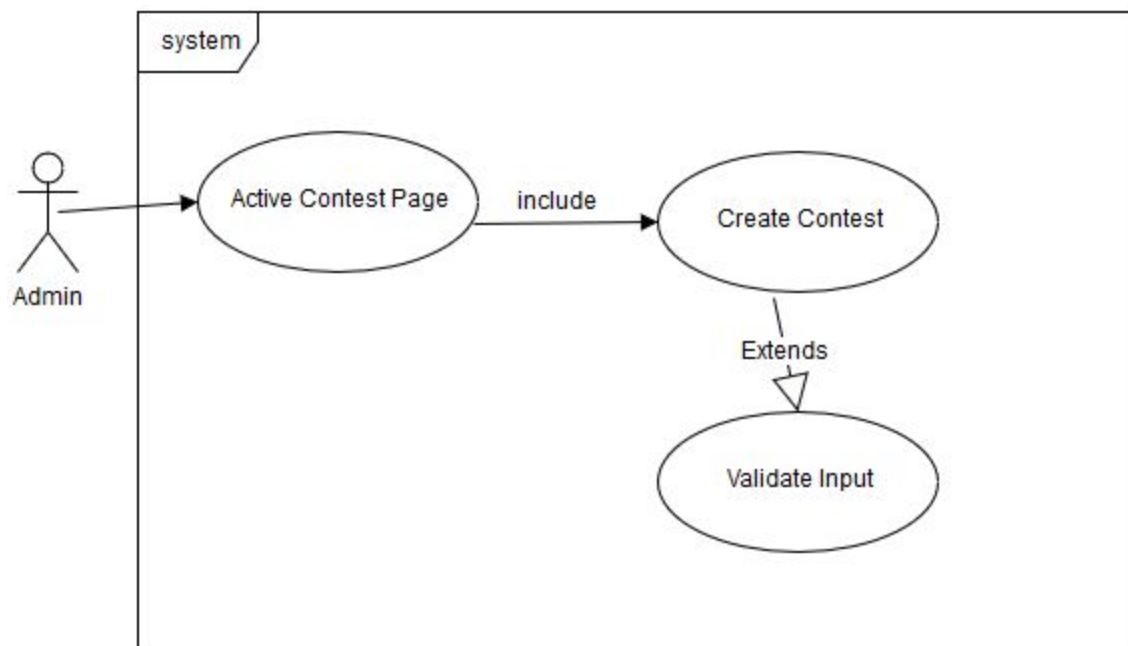


Figure #11 - User Story #696 Use Case Diagram

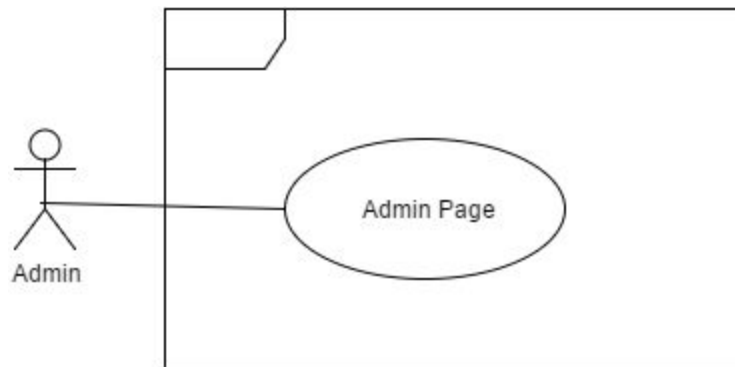


Figure #12 - User Story #683 Use Case Diagram

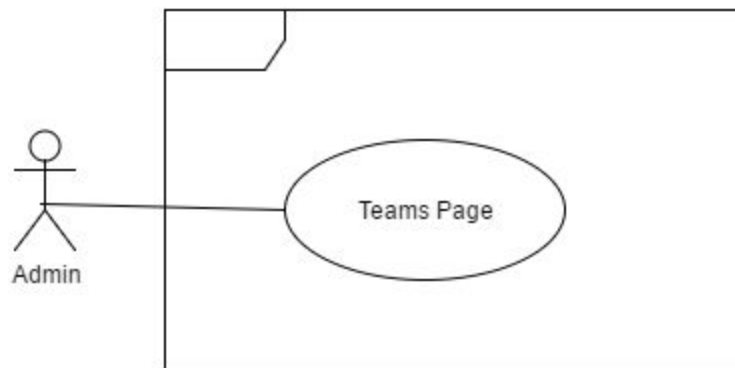


Figure #13 - User Story #684 Use Case Diagram

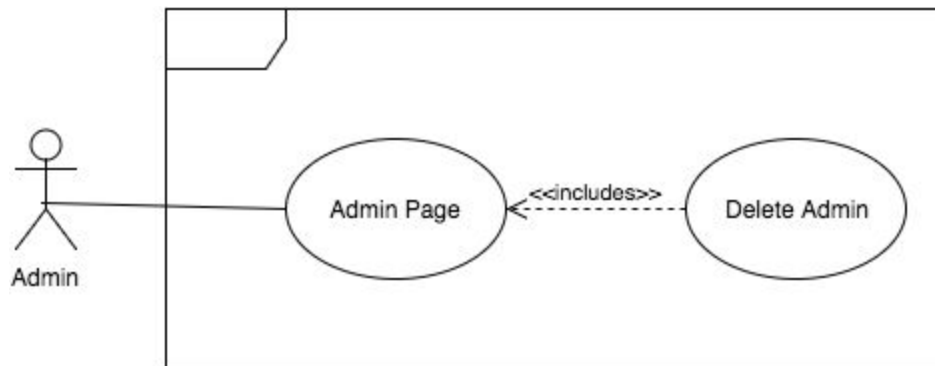


Figure #14 - User Story #718 Use Case Diagram

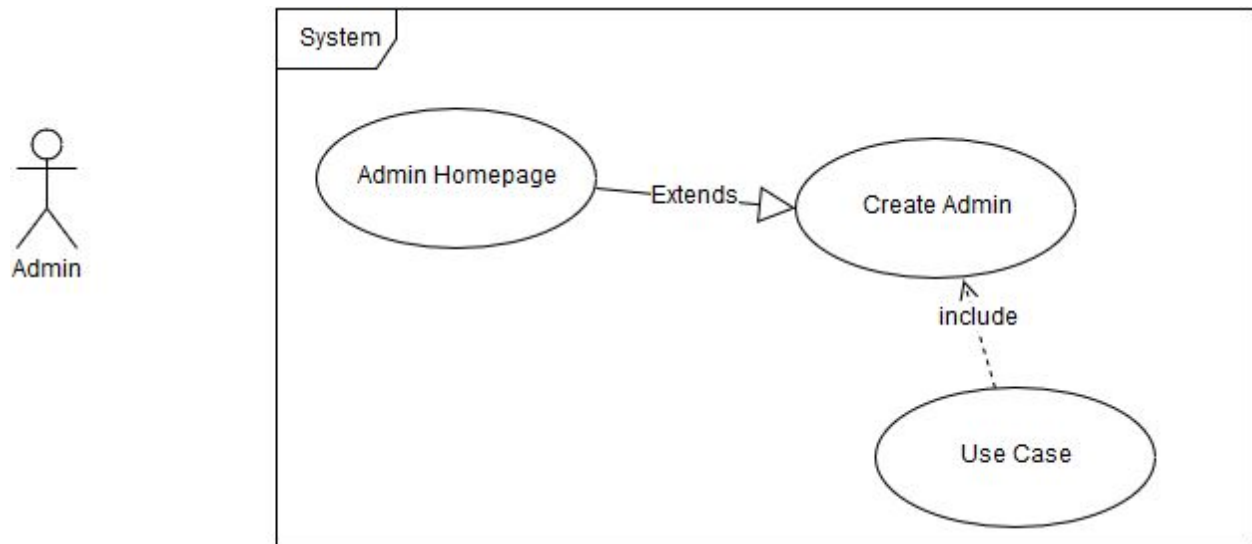


Figure #15 - User Story #680 Use Case Diagram



Figure #16 - User Story #686 Use Case Diagram

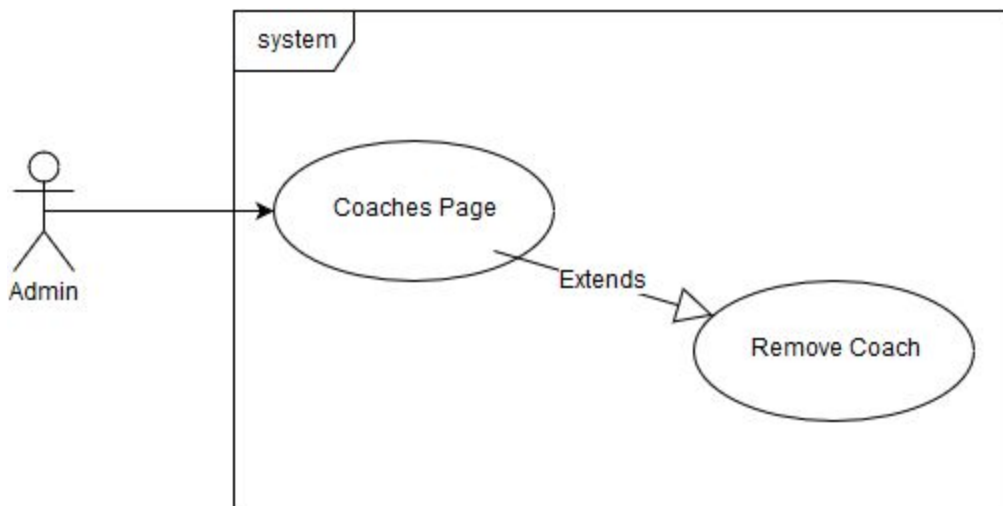


Figure #17 - User Story #709 Use Case Diagram

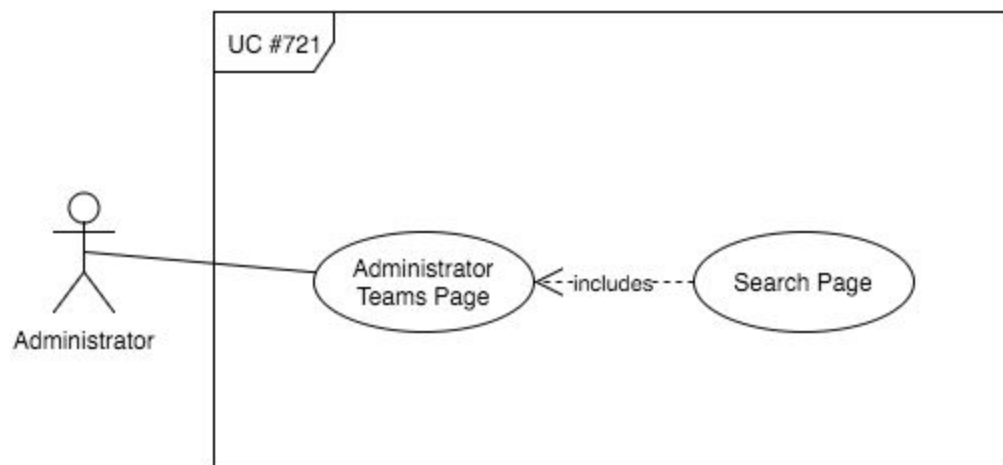


Figure #18 - User Story #721 Use Case Diagram

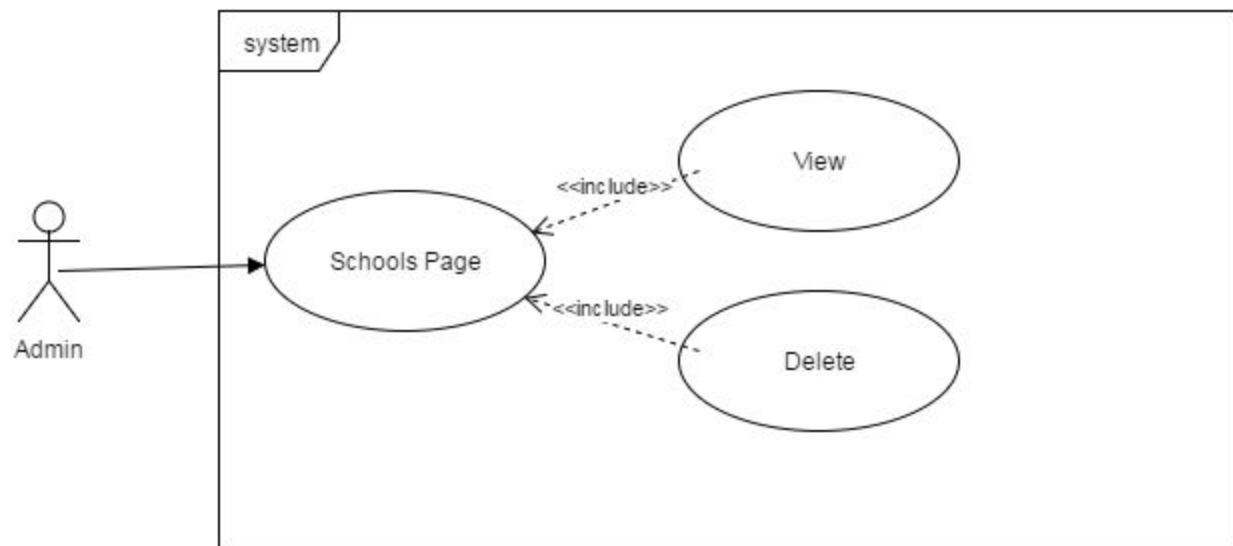


Figure #19 - User Story #727 Use Case Diagram

Appendix B - User Interface Design

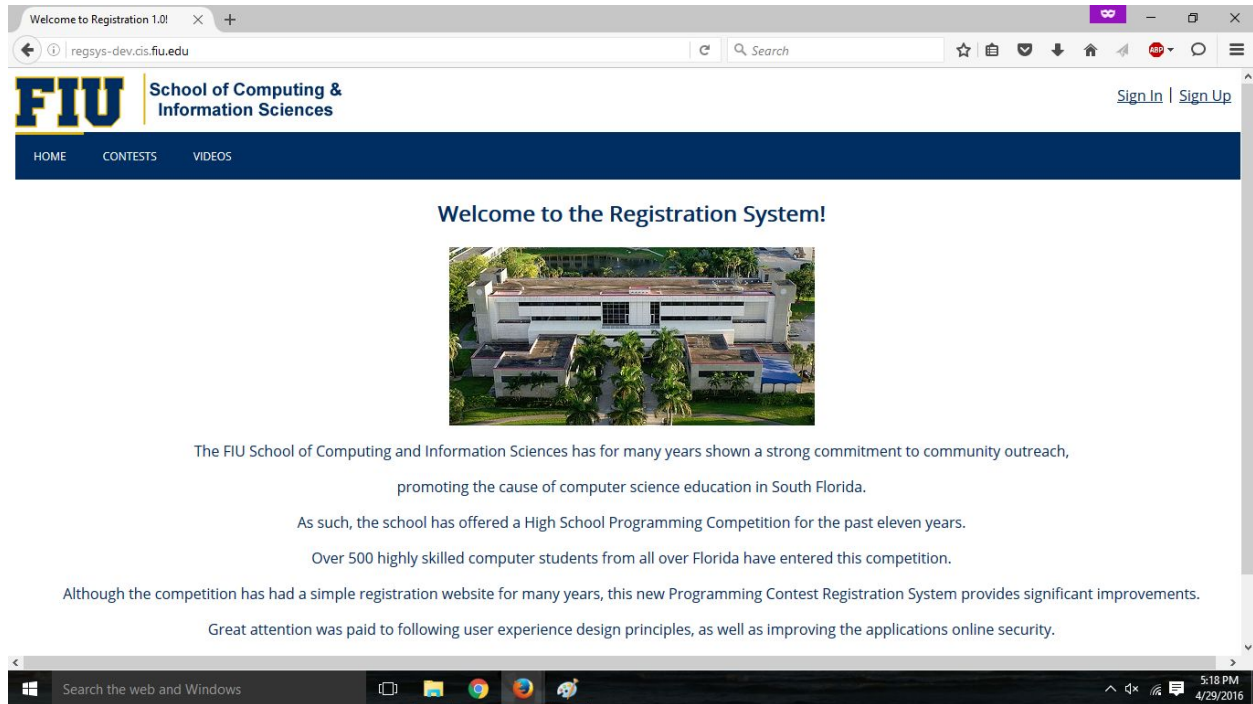


Figure #1 - Home Page

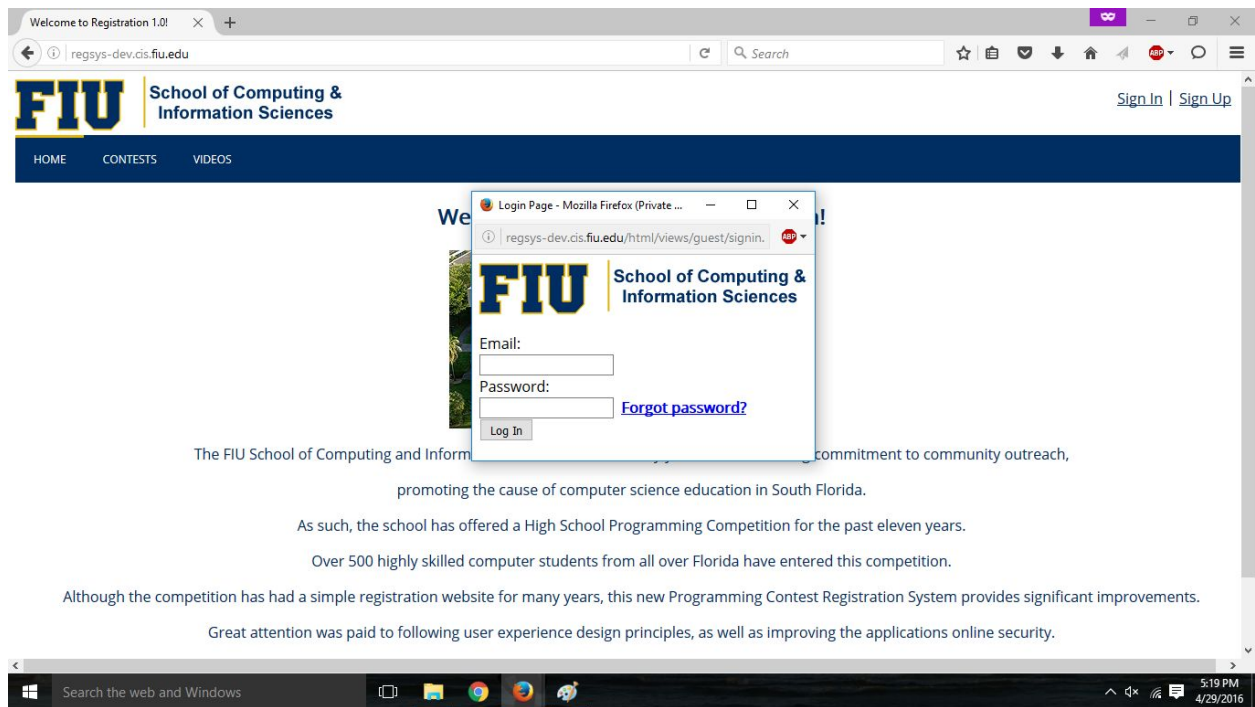


Figure #2 - Login Page

Registration Page

regsys-dev.cs.fiu.edu/html/jsp/signup.jsp

FIU School of Computing & Information Sciences

[Sign In](#) | [Sign Up](#)

HOME CONTESTS VIDEOS

Register here:

First Name:

Last Name:

Email:

School Name:

Password:

Confirm Password:

Search the web and Windows

5:19 PM 4/29/2016

Figure #3 - Registration Page

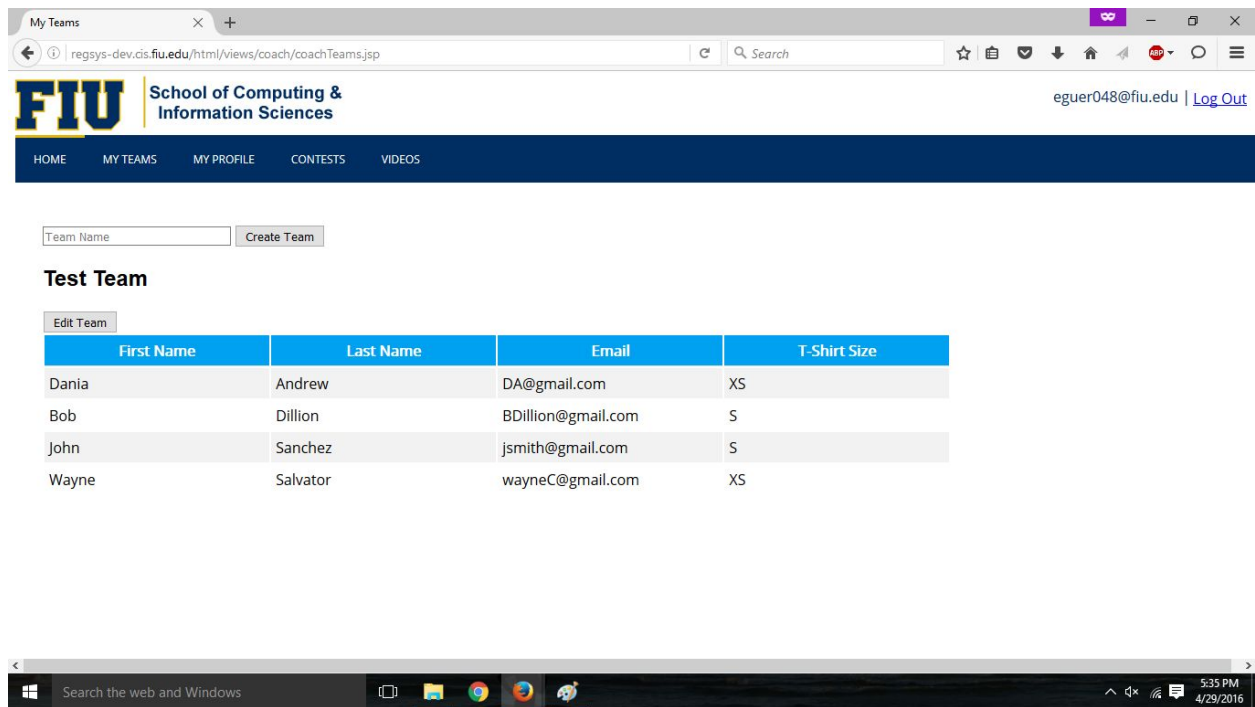


Figure #4 - Coach Views Teams

The screenshot shows a web browser window titled "Update Team" with the URL `regsys-dev.cs.fiu.edu/html/views/coach/coachEditTeam.jsp?teamId=1`. The page header includes the FIU logo and "School of Computing & Information Sciences". The user is logged in as `eguer048@fiu.edu` with a "Log Out" link. A navigation bar contains links for HOME, MY TEAMS, MY PROFILE, CONTESTS, and VIDEOS.

First Name	Last Name	Email	T-Shirt Size	
<input type="text" value="John"/>	<input type="text" value="Sanchez"/>	<input type="text" value="jsmith@gmail.com"/>	<input type="text" value="S"/>	<input type="button" value="Remove Participant 1"/>
<input type="text" value="Wayne"/>	<input type="text" value="Salvator"/>	<input type="text" value="wayneC@gmail.com"/>	<input type="text" value="XS"/>	<input type="button" value="Remove Participant 2"/>
<input type="text" value="Dania"/>	<input type="text" value="Andrew"/>	<input type="text" value="DA@gmail.com"/>	<input type="text" value="XS"/>	<input type="button" value="Remove Participant 3"/>
<input type="text" value="Bob"/>	<input type="text" value="Dillion"/>	<input type="text" value="BDillion@gmail.com"/>	<input type="text" value="S"/>	<input type="button" value="Remove Participant 4"/>

Below the table are buttons for "Save & Close" and "Cancel". At the bottom, there are input fields for "First Name", "Last Name", "Email", and a dropdown for "T-Shirt Size", followed by an "Add Participant" button.

The Windows taskbar at the bottom shows the search bar, task icons, and system clock (5:35 PM, 4/29/2016).

Figure #5 - Coach Modifies a Team

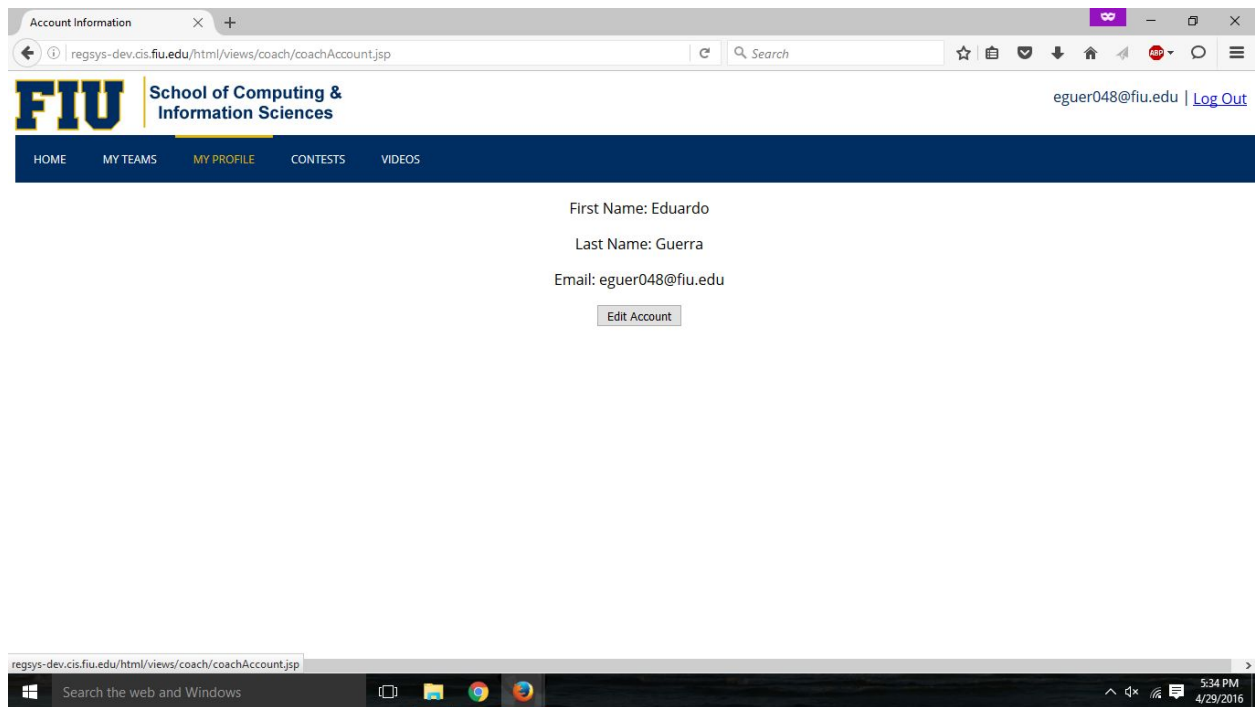


Figure #6 - Coach Profile

The screenshot shows a web application interface for the FIU School of Computing & Information Sciences. The page is titled "Participants" and displays a search result for "Coral Gables High School - Coach: John Curl". Below this, there is a table listing participants with columns for First Name, Last Name, Email, T-Shirt Size, and Team Name. Another section for "Florida International University - Coach: Eduardo Guerra" is also visible, showing a single participant.

Search for School:

Coral Gables High School - Coach: John Curl

First Name	Last Name	Email	T-Shirt Size	Team Name
Dania	Andrew	DA@gmail.com	XS	Test Team
Bob	Dillion	BDillion@gmail.com	S	Test Team
Bob	Ross	bross@gmail.com	XS	Panthers
Chris	Test	cTest@gmail.com	XL	Panthers
Hector	Jones	test@gmail.com	L	Senior Project
Bob	Dillion	bDill@gmail.com	XS	Team K

Florida International University - Coach: Eduardo Guerra

First Name	Last Name	Email	T-Shirt Size	Team Name
John	Sanchez	jsmith@gmail.com	S	Test Team

Figure #7 - Administrator Views Teams

The screenshot shows a web browser window with the title "Administrator Coach List". The address bar displays the URL "regsys-dev.cis.fiu.edu/html/views/admin/admincoachlist.jsp". The page header includes the FIU logo and the text "School of Computing & Information Sciences". A navigation bar contains links for "HOME", "CONTESTS", "PARTICIPANTS", and "CREATE ADMIN". The main content area features a table with the following data:

School Name	First Name	Last Name	Email	
Florida International University	Eduardo	Guerra	eguer048@fiu.edu	Remove School 1
Coral Gables High School	John	Curl	jcurl@gmail.com	Remove School 2
Coral Park High School	Wayne	Curling	wcurl001@fiu.edu	Remove School 3
Test School 123	FirstName	LastName	test123@gmail.com	Remove School 4

The browser's taskbar at the bottom shows the Windows logo, a search bar, and several application icons. The system clock indicates the time is 5:22 PM on 4/29/2016.

Figure #8 - Administrator Views Schools

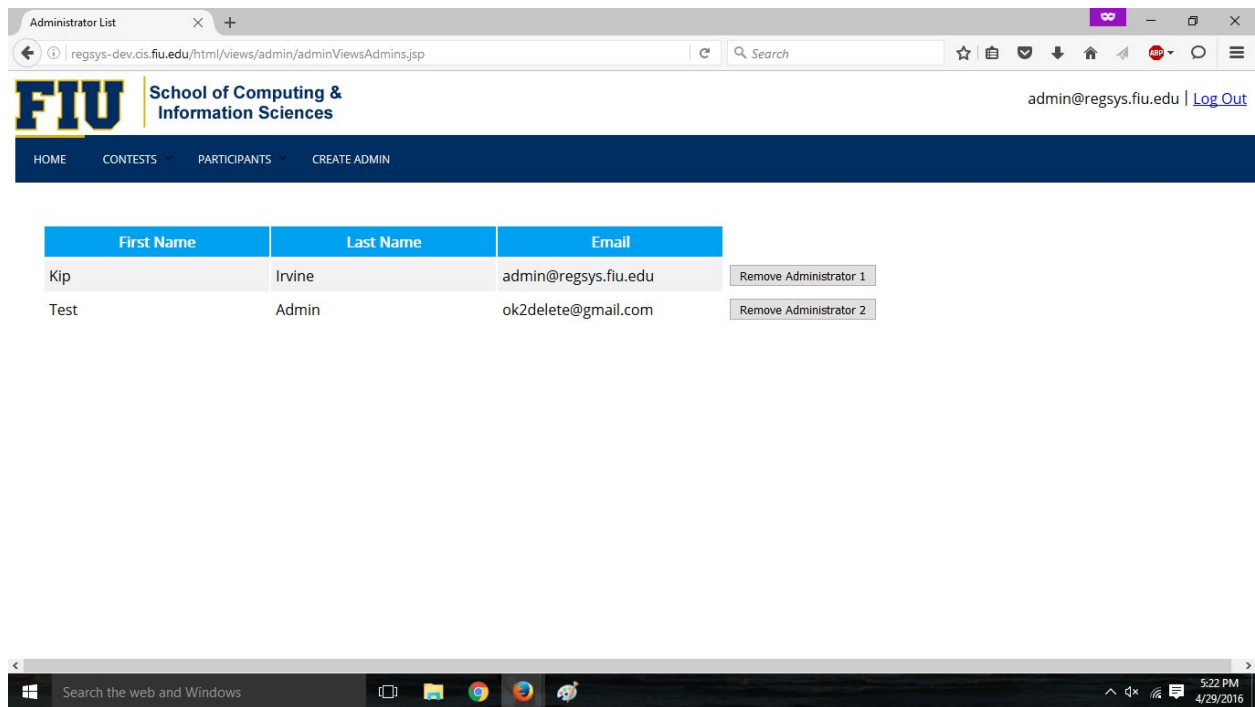


Figure #9 - Administrator Views Administrators

The screenshot shows a web browser window with the URL `regsys-dev.cis.fiu.edu/html/views/admin/adminSearchTeam.jsp?schoolName=Coral+Gables&submit=Search`. The page header includes the FIU logo and the text "School of Computing & Information Sciences". A navigation bar contains links for HOME, CONTESTS, PARTICIPANTS, and CREATE ADMIN. The user is logged in as `admin@regsys.fiu.edu` with a "Log Out" link.

The main content area displays the title "Coral Gables High School - Coach: John Curl" above a table of participants. The table has five columns: First Name, Last Name, Email, T-Shirt Size, and Team Name. Below the table is a "Go Back" button.

First Name	Last Name	Email	T-Shirt Size	Team Name
Dania	Andrew	DA@gmail.com	XS	Test Team
Bob	Dillion	BDillion@gmail.com	S	Test Team
Bob	Ross	brross@gmail.com	XS	Panthers
Chris	Test	cTest@gmail.com	XL	Panthers
Hector	Jones	test@gmail.com	L	Senior Project
Bob	Dillion	bDill@gmail.com	XS	Team K

Go Back

Figure #10 - Administrator Searches for Team

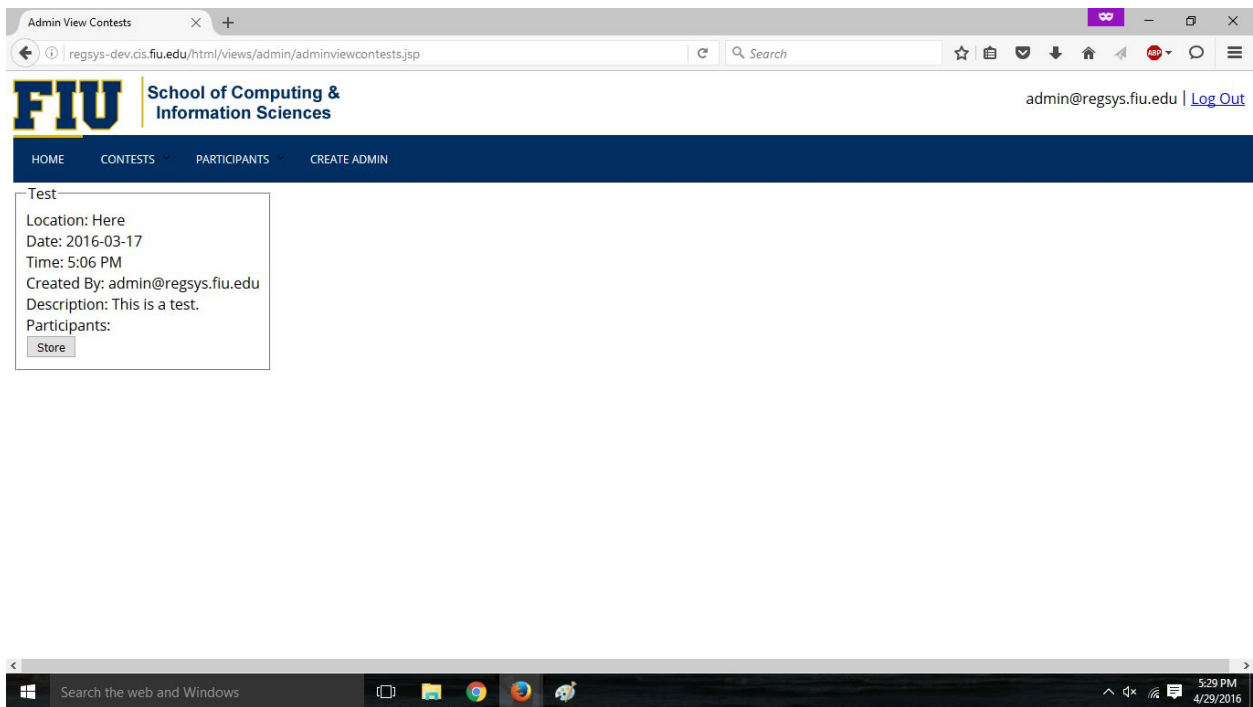


Figure #11 - Administrator View Active Contest

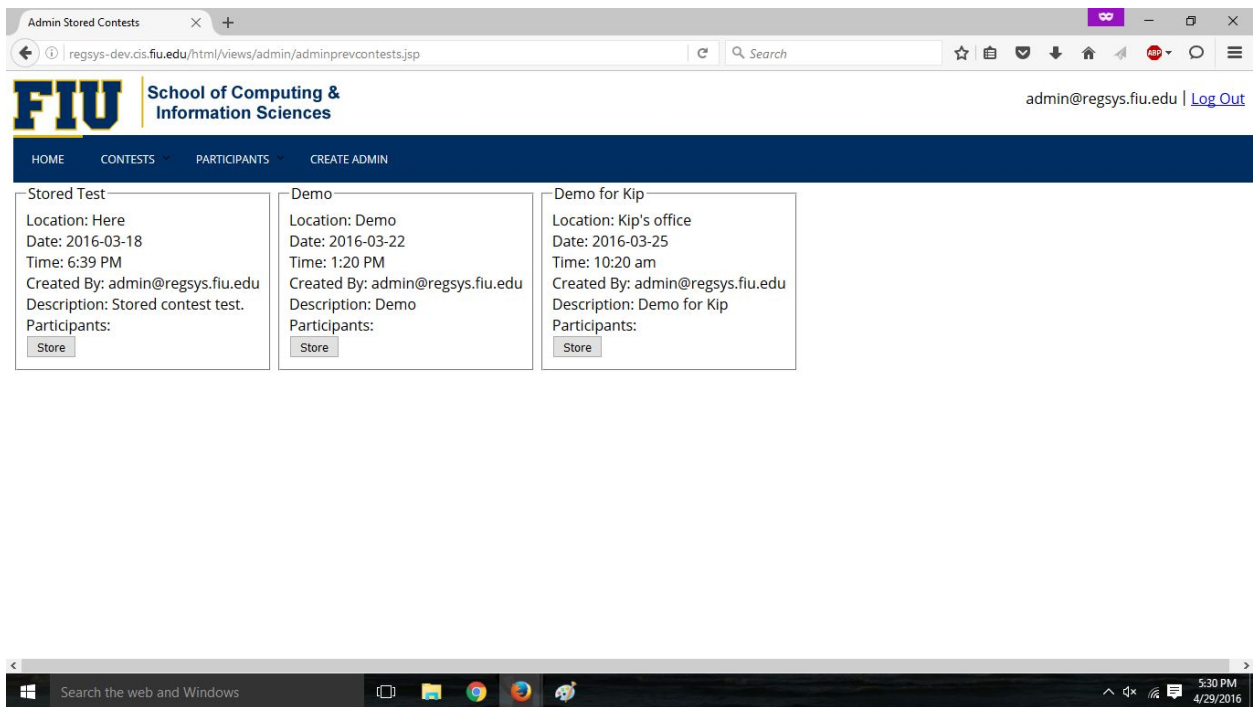
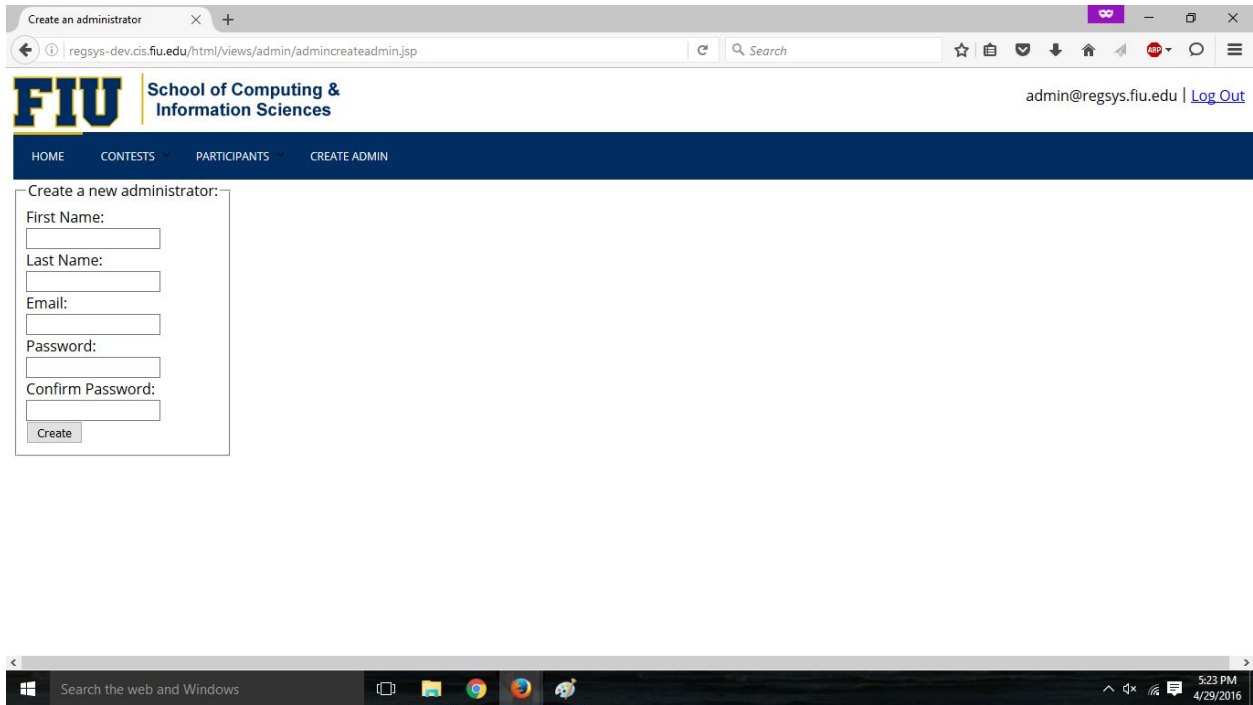


Figure #12 - Admin View Stored Contest



The screenshot shows a web browser window with the title 'Create an administrator'. The address bar shows the URL 'regsys-dev.ds.fiu.edu/html/views/admin/admincreateadmin.jsp'. The page header includes the FIU logo and the text 'School of Computing & Information Sciences'. The user is logged in as 'admin@regsys.fiu.edu' with a 'Log Out' link. The main navigation bar contains links for 'HOME', 'CONTESTS', 'PARTICIPANTS', and 'CREATE ADMIN'. The main content area is titled 'Create a new administrator:' and contains a form with the following fields: 'First Name:', 'Last Name:', 'Email:', 'Password:', and 'Confirm Password:'. Each field has a corresponding text input box. A 'Create' button is located at the bottom of the form. The browser's taskbar at the bottom shows the Windows logo, a search bar, and several open applications. The system clock in the bottom right corner indicates the time is 5:23 PM on 4/29/2016.

Figure #13 - Create Administrator

Administrator Create Contest

regsys-dev.cis.fiu.edu/html/views/admin/admincreatecontest.jsp

FIU School of Computing & Information Sciences

admin@regsys.fiu.edu | [Log Out](#)

HOME CONTESTS PARTICIPANTS CREATE ADMIN

Create a contest.

Contest Name:

Location:

Date:

Time:

Description:

Create

Search the web and Windows

5:21 PM 4/29/2016

Figure #14 - Create Contest

Appendix C - Sprint Review Reports

Sprint 1 Report

Date: 1/29/2016

Attendees: Eduardo Guerra, Wayne Curling, Kip Irvine

Start time: 10:30

End time: 10:45

After a show and tell presentation, the implementation of the following user stories were accepted by the product owners:

- User Story Documentation
- Coach Storyboards
- System Prototype

The following ones were rejected and moved back to the product backlog to be assigned to a future sprint at a future Spring Planning meeting.

- N/A
 - Reason for rejection:
 - N/A
 - How this should be reflected on the user story definition in Mingle:
 - N/A

Sprint 2 Report

Date: 2/12/16

Attendees: Eduardo Guerra, Wayne Curling, Kip Irvine

Start time: 11:00 AM

End time: 11:30 AM

After a show and tell presentation, the implementation of the following user stories were accepted by the product owners:

- None

The following ones were rejected and moved back to the product backlog to be assigned to a future sprint at a future Spring Planning meeting.

- User Story #678
 - Reason for rejection:
 - Incomplete
 - Dependency on User Story #687
 - How this should be reflected on the user story definition in Mingle:
 - User Story moved to Sprint 3
- User Story #687
 - Reason for rejection:
 - Incomplete
 - How this should be reflected on the user story definition in Mingle:
 - User Story moved to Sprint 3
- User Story #676
 - Reason for rejection:
 - Incomplete
 - How this should be reflected on the user story definition in Mingle:
 - User Story moved to Sprint 3

Sprint 3 Report

Date: 2/26/16

Attendees: Eduardo Guerra, Wayne Curling, Kip Irvine

Start time: 10:00 AM

End time: 10:30 AM

After a show and tell presentation, the implementation of the following user stories were accepted by the product owners:

- User Story #676
- User Story #678
- User Story #687
- User Story #688

The following ones were rejected and moved back to the product backlog to be assigned to a future sprint at a future Spring Planning meeting.

- N/A

Sprint 4 Report

Date: 3/11/16

Attendees: Eduardo Guerra, Wayne Curling, Kip Irvine

Start time: 11:00 AM

End time: 11:20 AM

After a show and tell presentation, the implementation of the following user stories were accepted by the product owners:

- User Story #679
- User Story #677
- User Story #681
- User Story #685
- User Story #694
- User Story #695
- User Story #696

The following ones were rejected and moved back to the product backlog to be assigned to a future sprint at a future Spring Planning meeting.

- N/A

Sprint 5 Report

Date: 4/1/16

Attendees: Eduardo Guerra, Wayne Curling, Kip Irvine

Start time: 11:00 AM

End time: 11:20 AM

After a show and tell presentation, the implementation of the following user stories were accepted by the product owners:

- User Story #684
- User Story #683
- User Story #718
- User Story #680
- User Story #686
- User Story #709

The following ones were rejected and moved back to the product backlog to be assigned to a future sprint at a future Spring Planning meeting.

- N/A

Sprint 6 Report

Date: 4/15/16

Attendees: Eduardo Guerra, Wayne Curling, Kip Irvine

Start time: 11:00 AM

End time: 11:20 AM

After a show and tell presentation, the implementation of the following user stories were accepted by the product owners:

- User Story #721

- User Story #727

The following ones were rejected and moved back to the product backlog to be assigned to a future sprint at a future Spring Planning meeting.

- User Story #682. Reason for rejection: Unable to get email system to work.

Sprint 6 Report

Date: 4/29/16

Attendees: Eduardo Guerra, Wayne Curling, Kip Irvine

Start time: 11:00 AM

End time: 11:20 AM

After a show and tell presentation, the implementation of the following user stories were accepted by the product owners:

- None

The following ones were rejected and moved back to the product backlog to be assigned to a future sprint at a future Spring Planning meeting.

- User Story #682. Reason for rejection: Unable to get email system to work.

Appendix D - Sprint Retrospective Reports

Sprint 1 Retrospective

Date: 1/29/16

Attendees: Eduardo Guerra, Wayne Curling

Start time: 11:00 AM

End time: 11:30 AM

What went wrong?

- We could have estimated velocity better because we had to research how to build the project
- Too much time spent on prototyping and not enough on documentation and development

What went right?

- We finished user stories and they were approved, we each picked one user story to work on
- Created storyboards for the website design
- Created documentation for user stories

How to address the issues in the next sprint?

- How to improve the process?
 - Spend more time on product development
- How to improve the product?
 - Stick to approved storyboards for documentation and development

Sprint 2 Retrospective

Date: 2/12/16

Attendees: Eduardo Guerra, Wayne Curling

Start time: 11:00 AM

End time: 11:30 AM

What went wrong?

- We could have estimated velocity better because we had didn't factor in possible server problems we might run into
- spent too much time on setting up the dev environment and git
- spent too much time on server issues and not on documentation and development

What went right?

- We were able to solve server issues
- Created documentation for user stories that were assigned
- began development on user stories

How to address the issues in the next sprint?

- How to improve the process?
 - Spend more time on product development
 - Refine documentation to factor in new server setup
- How to improve the product?
 - N/A

Sprint 3 Retrospective

Date: 2/29/16

Attendees: Eduardo Guerra, Wayne Curling

Start time: 11:00 AM

End time: 11:30 AM

What went wrong?

- N/A

What went right?

- We were able to complete 4 user stories this sprint
- We were able to complete testing on completed user stories
- We are happy with the progress this sprint

How to address the issues in the next sprint?

- How to improve the process?
 - Continue to maintain development pace
- How to improve the product?
 - N/A

Sprint 4 Retrospective

Date: 3/11/16

Attendees: Eduardo Guerra, Wayne Curling

Start time: 11:20 AM

End time: 11:30 AM

What went wrong?

- N/A

What went right?

- We were able to complete 7 user stories this sprint
- We were able to complete testing on completed user stories
- We are happy with the progress this sprint

How to address the issues in the next sprint?

- How to improve the process?
 - Continue to maintain development pace
- How to improve the product?
 - N/A

Sprint 5 Retrospective

Date: 4/1/16

Attendees: Eduardo Guerra, Wayne Curling

Start time: 11:15 AM

End time: 11:30 AM

What went wrong?

- N/A

What went right?

- We were able to complete 6 user stories this sprint
- We were able to complete testing on completed user stories
- We are happy with the progress this sprint

How to address the issues in the next sprint?

- How to improve the process?
 - Continue to maintain development pace
- How to improve the product?
 - N/A

Sprint 6 Retrospective

Date: 4/15/16

Attendees: Eduardo Guerra, Wayne Curling

Start time: 11:15 AM

End time: 11:30 AM

What went wrong?

- Could not get email system to work.

What went right?

- We were able to complete 2 user stories this sprint and a good amount of debugging.
- We were able to complete testing on completed user stories
- We are happy with the progress this sprint

How to address the issues in the next sprint?

- How to improve the process?
 - Continue to maintain development pace
- How to improve the product?
 - N/A

Sprint 7 Retrospective

Date: 4/29/16

Attendees: Eduardo Guerra, Wayne Curling

Start time: 11:15 AM

End time: 11:30 AM

What went wrong?

- Still could not get email system to work.

What went right?

- We were able to do a lot of UI changes and bug fixing.
- We are happy with the progress this sprint

How to address the issues in the next sprint?

- How to improve the process?
 - Continue to maintain development pace
- How to improve the product?
 - N/A

REFERENCES

You must reference any work that is not your own.