**DESIGN DOCUMENT**



**Coach Application**

**Registration System**

by

Devin DuBose, Donald Frank, Preecha Anuvattanachai

System Design CS246 Students

March 20th, 2017

[**1. Intro**](#_rnjdrxambyve) **4**

[1.1 KidSports’ And Registration Process](#_ulo9hvspn4mi) 4

[1.2 Proposal](#_nhzrrzgap5ji) 4

[1.3 Scope](#_x8n9xjayrigj) 5

[1.4 Purpose](#_wqu6bmkqdelf) 5

[**2. Application Uses**](#_ny96pcivqwqw) **6**

[2.1 System Users](#_q4hatwh1qn92) 6

[2.2 Application Uses](#_ahvnt0vhskte) 6

[2.3 Requirements](#_uc9dgci0apzt) 8

[**3. Architectural Overview**](#_tzd4g4gl63dp) **9**

[3.1 Topology](#_mvmhgfaw9b1p) 9

[3.2 Assumptions](#_aqx9joke6ho4) 9

[3.2.1 Constraints](#_w9836oejpeni) 10

[3.2.2 Exceptions](#_jk3oel2q633v) 10

[**4. Coach Application System Process**](#_n4we5ayjcmfg) **11**

[4.1 Registration](#_4t8131ptenlo) 11

[4.1.1 Process](#_wevazqhaiipd) 11

[4.2 CRIS API](#_r3cym3k2hwo3) 11

[4.2.1 Documentation](#_v0hsbt94oz8q) 11

[4.2.2 Response Examples](#_21agzyecxm5c) 11

[4.2.3 CRIS API Interaction SSD](#_bdg3rjwwvvp5) 12

[**5. Design Considerations**](#_v65yihhu6743) **13**

[5.1 System Goals](#_ytqnacxzf111) 13

[5.2 Development Methodologies](#_4exz29rllxl7) 13

[5.2.1 MVC](#_t912ii65rrq8) 13

[5.2.2 Entity Framework](#_jhukdz7k6x3r) 14

[**Appendices**](#_jz2uj8o1rosn) **16**

[Appendix A](#_fyusk8o6j7hs) 16

[Glossary](#_c8bjx39cppha) 16

[Appendix B](#_h2jwcxke4jfx) 17

[Registration Use Case Diagram](#_duq99d6gmcb2) 17

[Application Use Case Diagram](#_hfsmjsr6ax76) 17

[Appendix C](#_makj9znqhqgb) 18

[Workflow Diagram](#_f4z2n9h4bawp) 18

[Workflow Diagram - Cont.](#_a67mufk9gi9u) 19

[Appendix D](#_9gdg2pjy6bec) 20

[CRIS API - resultsList Response](#_phllgz9dqif5) 20

[Appendix E](#_d433xwgxetxe) 22

[Class Diagram](#_bhnialft5k0y) 22

[Appendix F](#_13x8ukcs7w5k) 23

[Sitemap](#_lg1vzb1a28kg) 23

[Appendix G](#_iu9kutvpm66f) 24

[Data Dictionary](#_eg8f6u7r0bdh) 24

[Appendix H](#_jq8wod1uq5t7) 26

[Timeline](#_6j9guzcu46ko) 26

[Design Phase](#_bbxx55nbiq8v) 26

[Development Phase](#_xhhc8xkutf9e) 26

[Appendix I](#_oax128pxkyvm) 29

[System Sequence Diagram](#_xx79n6ollblm) 29

# 

# 

# 1. Intro

## 1.1 KidSports’ And Registration Process

Definition

KidSports is a 501(c)(3) nonprofit organization that provides eight different sports programs-- year around -- for approximately 13,000 children in the Bethel, Springfield, and 4J school districts, In 2016, KidSports had 778 head coaches and 445 assistant coaches registered through their process.

Registration Process

Currently, all forms are static on the KidSports Website. Application, certificate, photo, and ID must be delivered via email, fax or in person.

The following are steps the applicant must perform during the registration process:

* **Coaches Application/Background Check Form**. Every year, complete, sign, date, submit to Sports Manager New coaches must submit valid ID
* **KidSports Coaches Pledge:** read, sign, submit to Sports Manager (yearly)
* **NFHS Concussion Course:** “Concussion in Sports - What you need to know”. Complete the course and submit the certificate to Sports Manager (yearly)
* **Positive Coaching Alliance Course:** “Double-Goal Coach: Coach For Winning Lessons”(Head Coaches).Complete course, submit certificate to Sports Manager
* **KidSports Coach ID Badge.** Send individual photo (minimum 8 mega pixel resolution

The following are steps taken by an Administrator after receiving an application:

* Write on the top left corner the Initials of the Sports Manager
* Highlight ‘Head Coach’ or ‘Assistant Coach’
* Date Stamp the top of all pages that came with the packet
* Highlight the areas that have missing paperwork
* Run the Background check via [Criminal Information Services, Inc.](https://www.criminalinfo.com/index.php)
* Print the Summary Page to attach to Packet
* Initial and Date Stamp that the BG Approved Line
* Make a Black and White Copy for Administration
* Make a Color Copy to give to the Sports Manager
* Staple the Packet together
* Input all information into an Excel Spreadsheet:
* Enter any updates within excel and mark the changes on the Sports Manager Color Copy
* Date Stamp the ‘Application Processed’ Line once it’s a fully completed

## 1.2 Proposal

Design and develop an interactive data-driven Coach Application Registration System to expedite the current processes

## 1.3 Scope

The scope of this project is limited to developing a web application for the Coach Application Registration System defined in the requirements gathering stage. And not including any of KidSports’ other processes.

## 1.4 Purpose

The purpose of this document is to outline the key components of the technical design phase, and to explain the methodologies of the system. Included in the document are the following:

* Users
* Requirements
* Processes
* Application Architecture

# 

# 

# 

# 

# 2. Application Uses

## 2.1 System Users

Guest

A guest is an unregistered user who visits the application. This role will be in place to allow users to view applicant requirements and other details regarding the process of coaching registration.

Applicant

An applicant is a registered user within the system. When logged in, a user with this role will have access to coach application forms and other registration processes.

Sports Manager

A sports manager is a user that is an ‘employee’ of the KidSports organization. When logged in, a user with this role will have access to viewing coaching applications. This user will be able to filter through applications and approve valid information.

Administrator

An administrator is a user that is an ‘employee’ of the KidSports organization with permission to view sensitive information, for example a member of the human relations team. When logged in, a member with this role will have access to all privileges granted by the sports manager role. Administrators will also have permission to view background check results for applicants.

## 2.2 Application Uses

Some uses may be shared across multiple levels of user authorization. All user roles will have permission to access the use cases that may be operated by a role with preceding levels of authorization. The level of access granted to users from most access to least access are as follows.

(Administrator -> Sports Manager -> Applicant -> Guest)

Guest

|  |  |
| --- | --- |
| **Use Case** | **Description** |
| Register user | Using full name, email, and password |
| View application instructions | View and/or download instructions for the coaching application process |

Applicant

|  |  |
| --- | --- |
| **Use Case** | **Description** |
| Login user | Using username and password |
| Submit application | Enter new detailed information regarding both applicant’s personal background and their interests in coaching. **See appendix B.** |
| Update application | Update existing information with changes to the applicant’s background and their interests in coaching. |
| View application | Display a list the users past submitted applications |
| Update account | Update contact information such as full name, phone number, email. Users will also be able to update their account password for ease of access. |
| Upload Image | Upload images of certifications and of the applicant’s driver’s license. The user will also be able to upload a photo for their ID badge if their application is successful. |

Sports Manager

|  |  |
| --- | --- |
| **Use Case** | **Description** |
| Login user | Using username and password |
| View applicants | Using name or email |
| View applications | Display a list of applications filtered by subjects (sport, district, age group, season) |
| Validate application | Mark the successful completion of application components. |
| Register applicant (manually) | File an application into the system on behalf of a non-electronic submission by an applicant. |
| Update account | Update contact information such as full name, phone number, email. Users will also be able to update their account password for ease of access. |

Administrator

|  |  |
| --- | --- |
| **Use Case** | **Description** |
| Login user | Using username and password |
| View applicants | Using name or email |
| View applications | Display a list of applications filtered by subjects (sport, district, age group, season) |
| Validate application | Mark the successful completion of application components. |
| Register applicant (manually) | File an application into the system on behalf of a non-electronic submission by an applicant. |
| Update account | Update contact information such as full name, phone number, email. Users will also be able to update their account password for ease of access. |
| View BG checks | Display a list of background checks by applicant  Display a single background check by application |
| Validate background checks | Mark successful completion and passing status of a background check |
| Change permissions | Promote and demote a user’s authorization levels |

**For use case diagrams, see appendix B.**

## 2.3 Requirements

Functional

|  |  |  |
| --- | --- | --- |
| Functional | Functions | System (Business Rules)  Require authentication and authorization, accept user application input, validate user application input, store application data, display application data, datestamp application, retrieve application data, update application, transfer background check information to CRIS API, notify user, expire application, upload file(s), populate renewal application |

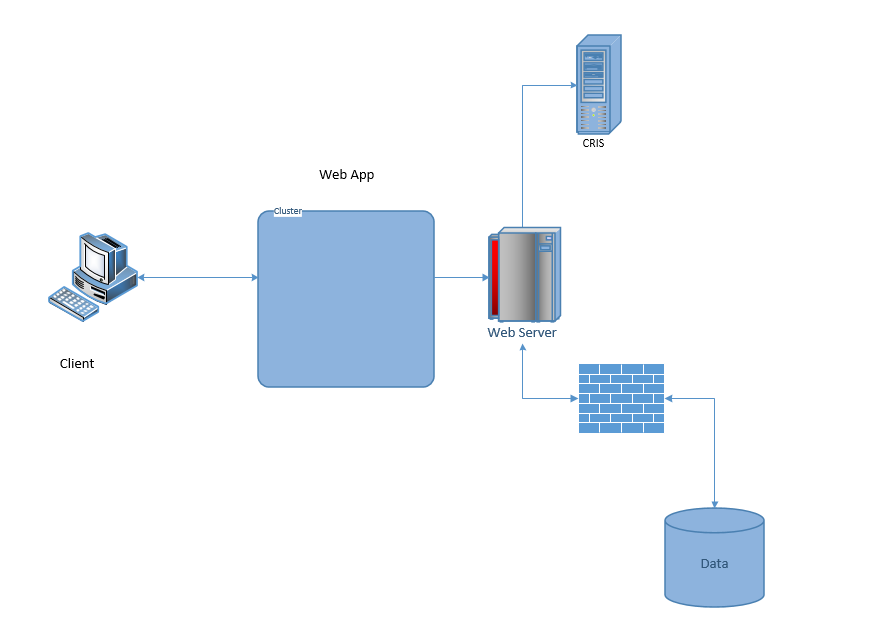
Non-Functional

|  |  |  |
| --- | --- | --- |
| Non-functional | Usability  Reliability  Performance  Security  + Design constraints  Implementation  Interface  Physical  Support | User interface, UX, Easy to operate  Low failure rate, Exception handling  Response time, Pages load quickly  Authorization, Encrypted Identification  EF may limit the control of DB design  .Net core framework may be volatile  Accessing CRIS API for bg checks  Server stability and strength  Installation and maintaining updates |

# 3. Architectural Overview

## 3.1 Topology

This diagram below provides an illustration of the System Architecture and communication scheme in the network.

Topology Diagram

**Client:** The client allows users to access the application via web browser.

**Web Application:** The web application provides the business logic. The web application will process and validate all requests made by clients and provide a response displaying relevant information.

**Web Server:** The application server will host the web application.

**Criminal Information Services (CRIS)**: CRIS provides service for the background check. Requests will be made from our web application to their API. CRIS will return a callback response to our web application when our information is processed.

**Firewall:** The firewall will identify and authorize requests made to the sql server, rejecting invalid requests. Only the web application server may access the data servers.

**SQL Server:** The data server will host the sql database.

**SQL Database:** The sql database will store information regarding the coaching applications, users, and background check results.

## 3.2 Assumptions

KidSports will pay a 3rd party provider for hosting services.

KidSports will separate the application server from the data server.

KidSports will maintain, update and backup the data regularly.

KidSports will have an active account with Criminal Information Services to use their API.

The web application will be hosted on the most recent windows OS.

A web browser will be the primary client used to access the application.

### 3.2.1 Constraints

Entity Framework code-first mapping may limit the control of DB design

.Net core framework may be volatile. Some features may not be supported.

Dependency on a 3rd party application for processing background checks. Accessing the CRIS API for background checks may have potential problems.

Depending on the increase in applicants and the current KidSports servers may not be strong enough to process user requests. This could potentially Lead to stability issues and crashes.

### 3.2.2 Exceptions

If the CRIS network is down, store the background check in a temporary state to be re-submitted later.

If the sql server is down, display a network error page. Please try again later.

# 

# 

# 4. Coach Application System Process

## 4.1 Registration

### 4.1.1 Process

* Applicant registers an account.
* Applicant logs in.
* Applicant opens new coach application.
* System populates the application with all previous existing information.
* Applicant completes the form, updating information as necessary.
* System validates the applicant’s input.
* System stores the submission, and create a new background check.
* System transcribes the background check into XML and sends request to CRIS API.
* Sports manager and Administrators may now view and validate components of the pending application at any time. **For System Sequence Diagram, see appendix I.**
* CRIS provides a response call back.
* Administrator views background check results.
* Administrator validates the background check based on pass/fail.
* Applicant uploads any missing components of the application.
* Sports manager will view the new material and validate it.
* Sports manager will approve the application.

**For a workflow diagram, See appendix C.**

## 4.2 CRIS API

### 4.2.1 Documentation

* **testConnect** – All this does is verify that you are able to connect to

the web service and that your user ID and password are valid.

* **searchName** – Submit a search request for a person’s name. A result XML

document will be returned highlighting the important data elements of .

matching records (Name, Description, Cases, Sentences etc).

* **getResults** – Submit a request for the next set or page of records, providing the

SessionID acquired from searchName.

* **getDetails** – Submit a search request for a particular record. A detail XML document will be returned which provides all data available regarding the particular case

### 4.2.2 Response Examples

**Success**

The response for a testConnect request would look like the following:

<CRISResponse>

<RESPONSETYPE>testConnect</RESPONSETYPE>

<STATUS>YES</STATUS>

</CRISResponse>

**Failure**

An example of an error response is as follows:

<CRISResponse>

<ERROR>

<CODE>100</CODE>

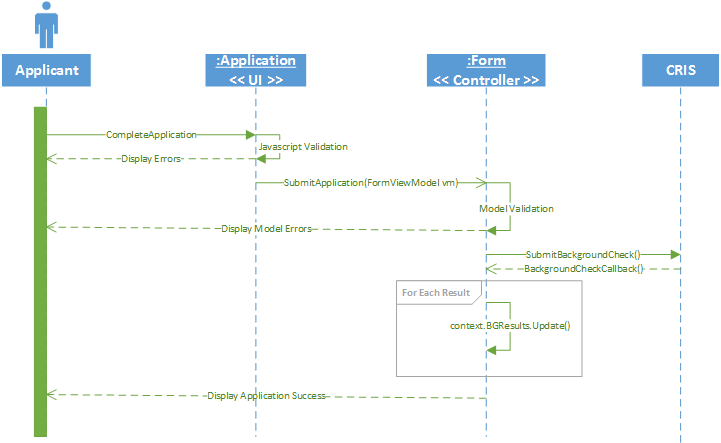
<MESSAGE>Invalid Search Criteria</MESSAGE>

</ERROR>

</CRISResponse>

**For further response examples, see appendix D.**

### 4.2.3 CRIS API Interaction SSD



# 

# 

# 5. Design Considerations

## 5.1 System Goals

Our goals with this system is to increase the number of coaches that register in a given year. We would also like to help expedite and ease the process of the application by removing some of the busy work that must be completed by the Sports Managers and Administrators. This will be accomplished by tasking the the system with validation restraints and data manipulation.

**For a timeline, see appendix H.**

## 5.2 Development Methodologies

### 5.2.1 MVC

MVC stands for Model, View, Controller. It is a design pattern that we chose due to a few

of its key strengths. It allows for separation of concerns and parallel development. This

will allow us to separate our development and assign tasks based on our specializations.

Models

|  |  |
| --- | --- |
| **Model Name** | **Fields:** |
| Application | applicant, pledge, position, sport, district, gender, season, concussion, sportsGender, pca, bgCheck, bgcResult, photoID, processedDate |
| User : IdentityUser | firstName, middleName, lastName, previousStates, address, city, state, zipcode, dob, currentEmployer, jobTitle, gender |
| District | name, schools |
| School | name |
| <<abstract>> Course | complete, valid, completeDate, validDate |
| PCA : Course | pcaFile |
| Concussion : Course | concFile |
| Pledge | signature, pledgeDate |
| BGCheck | dateSubmitted, bgCheckResult, passed |
| BGCheckResult | resultSet |

**For a class diagram, See appendix E.**

Views

|  |  |
| --- | --- |
| **View Name** | **Displays:** |
| Register | Display username/password/email input boxes. |
| Login | Display username/password input boxes. |
| Application | Display an application form. |
| Coaches Pledge | Display terms of the coaches pledge, with an I agree to terms / signature requirement. |
| Concussion Course | Display an upload / remove (if not verified) for the concussion course certification. |
| PCA Course | Display an upload / remove (if not verified) for the PCA course certification. |
| Driver’s Licence | Display an upload / remove photo option for the applicants driver’s licence |
| Badge Photo | Display an upload / remove photo option for their coach ID badge. |
| Applications | Display a list of coach applications to SM’s and Admins. |
| Applicants | Display a list of coaches to SM’s and Admins. |
| Background Checks | Display a list of unresolved background checks. |

**For a sitemap, see appendix F.**

Controllers

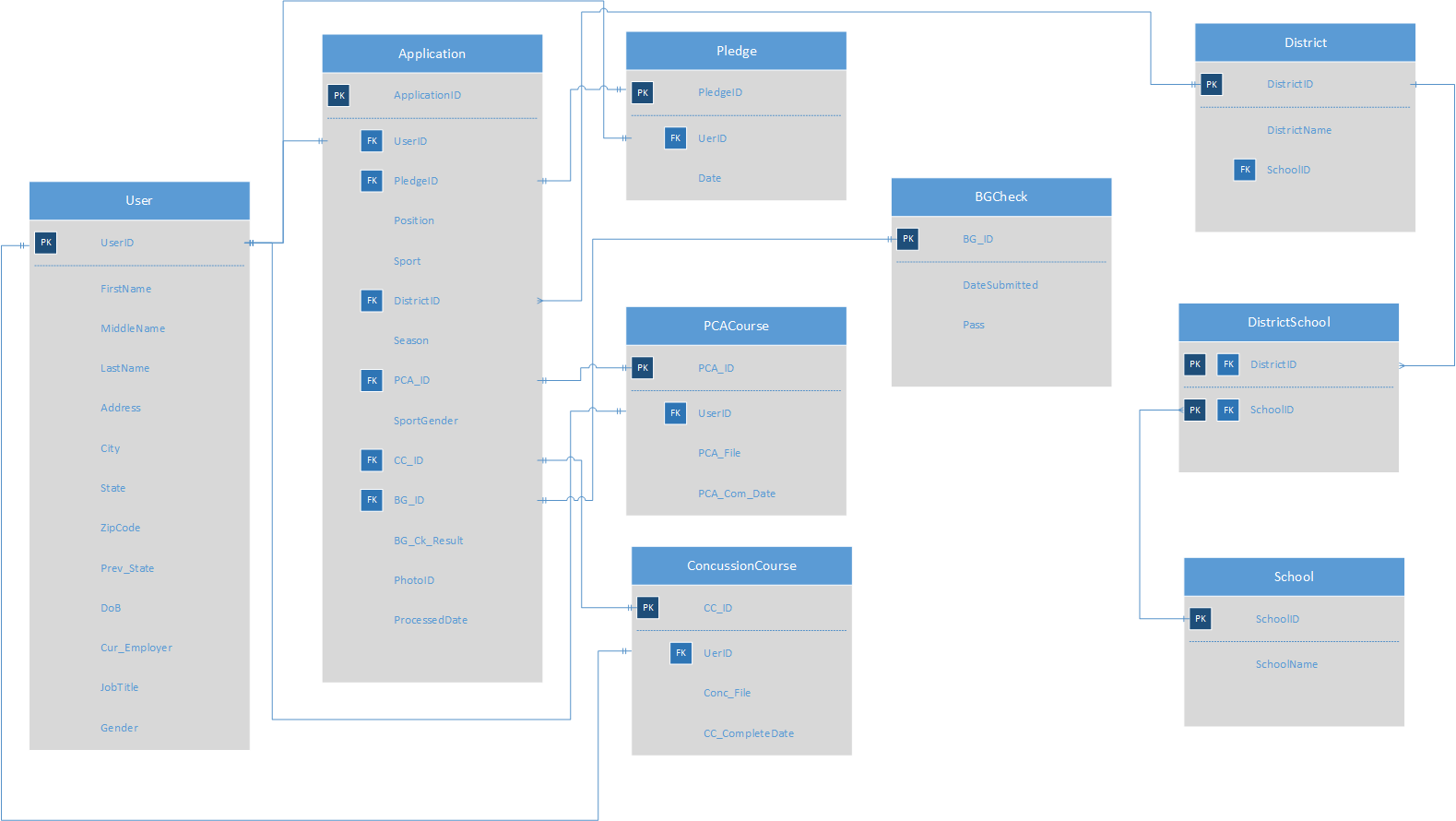
|  |  |
| --- | --- |
|  |  |
| Home | Index |
| Form | Apply, Update, View, Upload |
| Account | Register, Login, Profile |
| SMController | Applicants, Applications |
| AdminController | Applicants, Applications, BGCheckResults |

### 5.2.2 Entity Framework

Entity framework will be a crucial factor in the construction of our database. We look to

use a code-first design strategy in which we will build our system’s architecture and use

EF to map out a database structure that matches appropriately.



**For a data dictionary, see appendix G.**

# Appendices

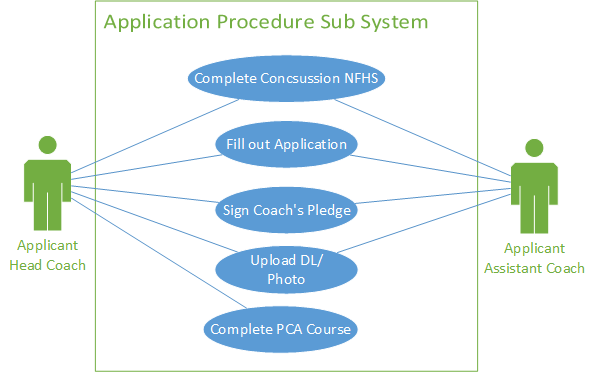
## Appendix A

### Glossary

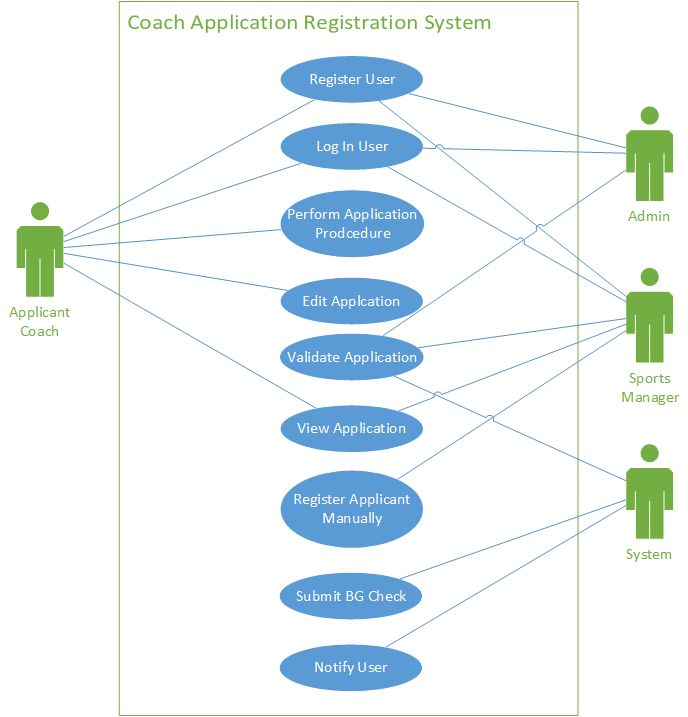
|  |  |
| --- | --- |
| **4J** | Eugene School district |
| **Admin** | Administrator |
| **API** | Application Program Interface |
| **BG** | Background |
| **CRIS** | Criminal Information Services |
| **DBMS** | Database Management System |
| **DL** | Driver’s License |
| **EF** | Entity Framework |
| **MVC** | Model View Controller |
| **NFHS** | National service and administrative organization of high school athletics and fine arts programs |
| **OS** | Operating System |
| **PCA** | Positive Coaching Alliance |
| **SSD** | System Sequence Diagram |
| **SM** | Sports Manager |
| **SQL** | Structured Query Language |
| **UI** | User Interface |
| **UX** | User Experience |
| **XML** | Extensible Markup Language |

## Appendix B

### Registration Use Case Diagram



### Application Use Case Diagram



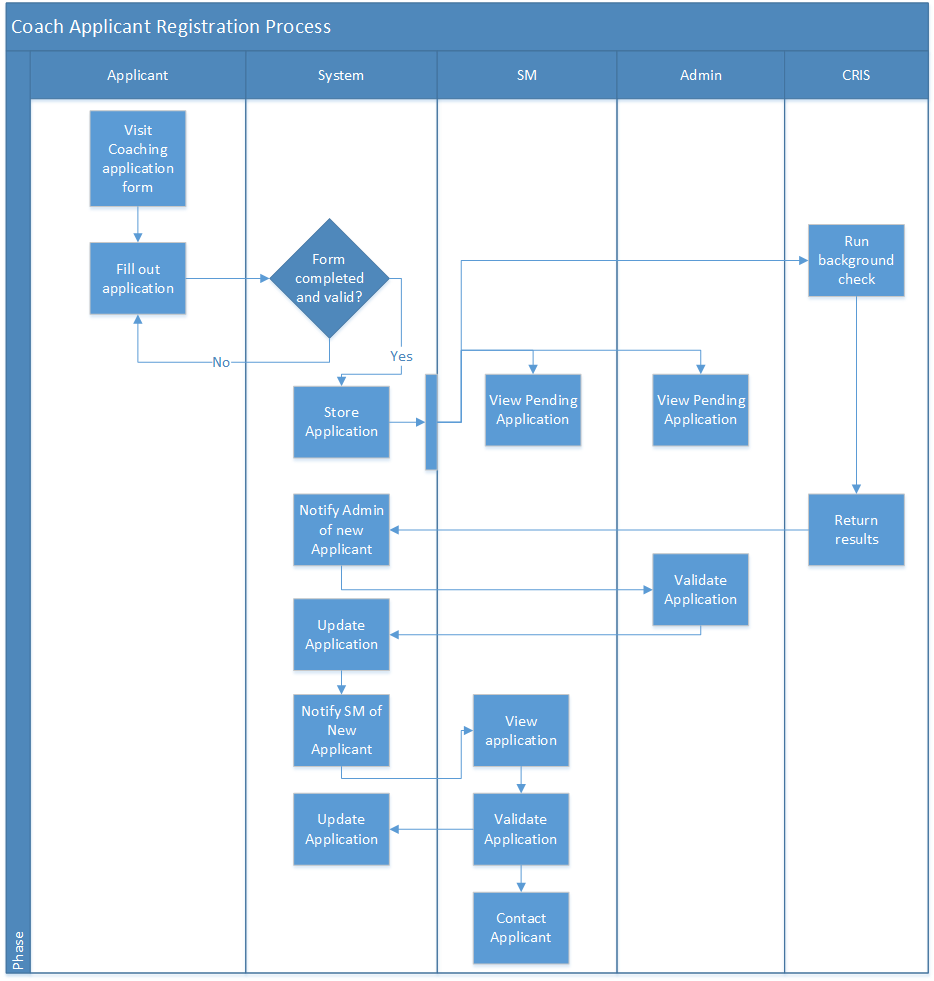
## 

## 

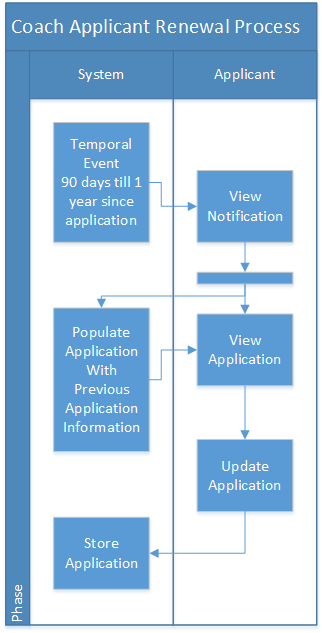
## 

## Appendix C

### Workflow Diagram



### Workflow Diagram - Cont.



## Appendix D

### CRIS API - resultsList Response

<CRISResponse>

<ResponseType>resultsList</ResponseType>

<Matches/>

<SessionID>rmT1uosIHLUy5idhclbKaA</SessionID>

<RecStart>0</RecStart>

<ClientCode/>

<Results>

<Case state='ID' type='DOC' rec\_num="1">

<Identification>

<Person>

<Name type="DEF-Record">

<LName>JONES</LName>

<FName>JAMES</FName>

<MName>ALAN</MName>

</Name>

<DOB>

<Year4>1925</Year4>

<Month>10</Month>

<Day>15</Day>

</DOB>

<Gender>MALE</Gender>

<Race>WHITE</Race>

</Person>

<Person>

<Name type="DEF-Alias">

<LName>JONES</LName>

<FName>JAMES</FName>

<MName>ALAN</MName>

</Name>

<DOB>

<Year4>1925</Year4>

<Month>10</Month>

<Day>15</Day>

</DOB>

</Person>

</Identification>

<CaseInformation>

<DataType>DOC</DataType>

<UniqueID>IDDOC\*179999</UniqueID>

<Jurisdiction>Idaho Department of Corrections</Jurisdiction>

<StatusDate>

<Year4>1982</Year4>

<Month>02</Month>

<Day>11</Day>

</StatusDate>

<StatusDescription>Probation</StatusDescription>

</CaseInformation>

<OffenseInformation>

<OffenseIncident>

<CaseNumber>C-98872148-A</CaseNumber>

<County>BANNOCK</County>

<Offense>

<Description>PETIT LARCENY</Description>

</Offense>

<OffenseType>

<Level>Misdemeanor</Level>

</OffenseType>

<FiledDate>

<Year4>1989</Year4>

<Month>01</Month>

<Day>13</Day>

</FiledDate>

<Sentence>Minimum</Sentence>

<SentenceLength type="text"> 1 yr 0 mon 0 days</SentenceLength>

</OffenseIncident>

</OffenseInformation>

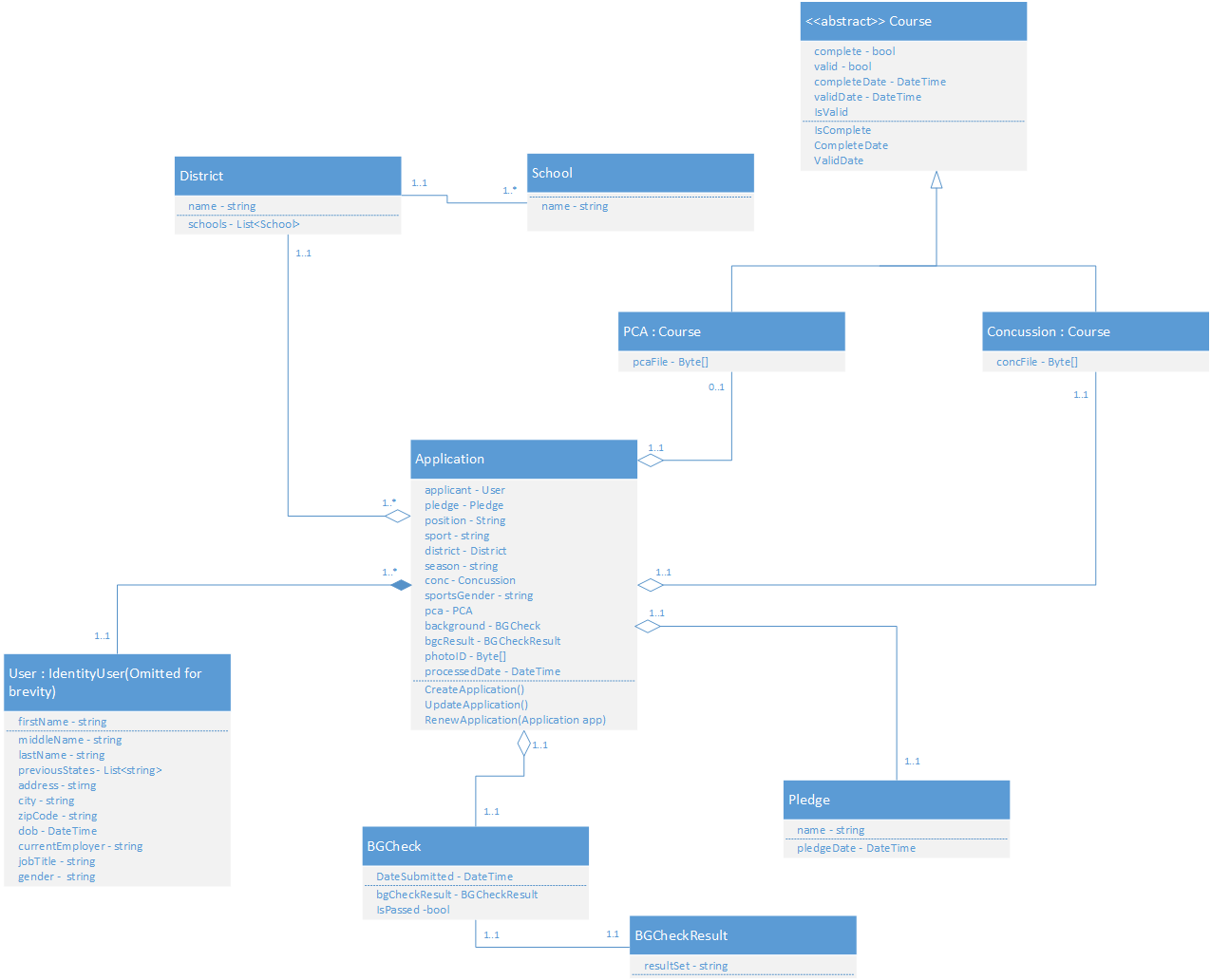
</Case>

</Results>

</CRISResponse>

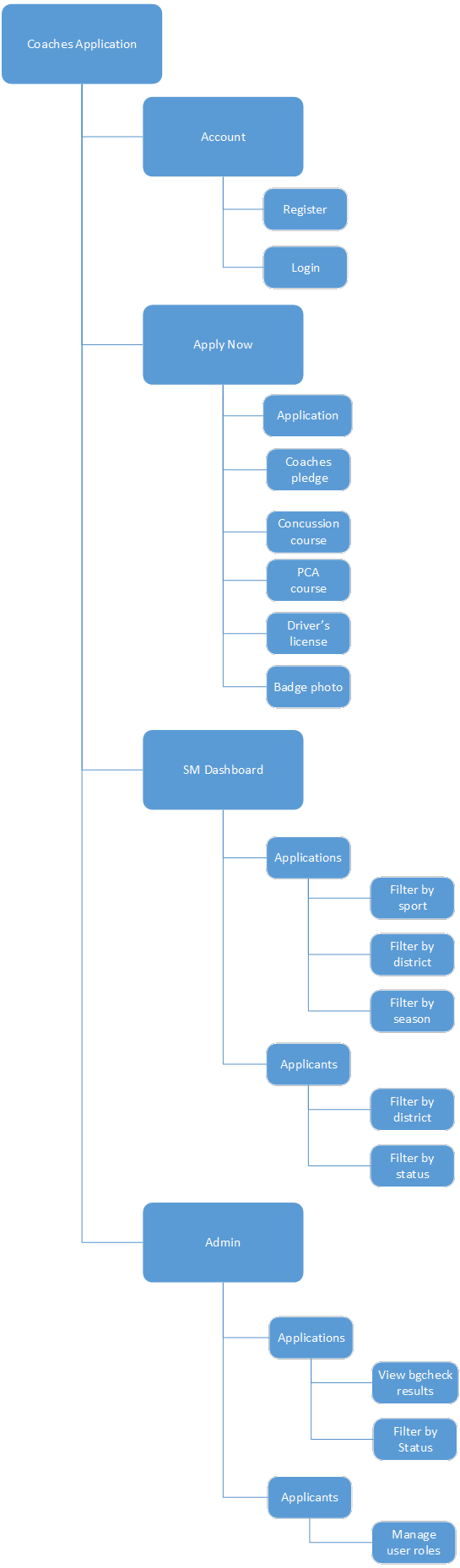
## Appendix E

### Class Diagram



## Appendix F

### Sitemap



## Appendix G

### Data Dictionary

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Table | Key | Field Name | Data Type | Field size |
| Application | PK | ApplicationID | varchar | 10 |
| Application | FK | UserID | varchar | 10 |
| Application | FK | PledgeID | varchar | 10 |
| Application |  | Position | varchar | 16 |
| Application |  | Sport | varchar | 16 |
| Application | FK | DistrictID | varchar | 10 |
| Application |  | Season | varchar | 8 |
| Application | FK | PCA\_ID | varchar | 10 |
| Application |  | SportGender | varchar | 6 |
| Application | FK | CC\_ID | varchar | 10 |
| Application | FK | BG\_ID | varchar | 10 |
| Application |  | BG\_Ck\_Result | varchar | 40 |
| Application |  | PhotoID | varchar | 10 |
| Application |  | ProcessedDate | Date | 10 |
| User | PK | UserID | varchar | 10 |
| User |  | FirstName | varchar | 10 |
| User |  | MiddleName | varchar | 10 |
| User |  | LastName | varchar | 20 |
| User |  | Address | varchar | 30 |
| User |  | City | varchar | 20 |
| User |  | State | varchar | 20 |
| User |  | ZipCode | varchar | 12 |
| User |  | Prev\_State | varchar | 20 |
| User |  | DoB | Date | 10 |
| User |  | Cur\_Employer | varchar | 30 |
| User |  | JobTilte | varchar | 20 |
| User |  | Gener | varchar | 6 |
| Pledge | PK | PledgeID | varchar | 10 |
| Pledge | Fk | UserID | varchar | 10 |
| Pledge |  | Date | varchar | 10 |
| PCACourse | PK | PCA\_ID | varchar | 10 |
| PCACourse | FK | UserID | varchar | 10 |
| PCACourse |  | PCA\_File | varchar | 50 |
| PCACourse |  | PCA\_Com\_Date | Date | 10 |
| ConcussionCourse | PK | CC\_ID | varchar | 10 |
| ConcussionCourse | FK | UserID | varchar | 10 |
| ConcussionCourse |  | Conc\_File | varchar | 50 |
| ConcussionCourse |  | CC\_Com\_Date | Date | 10 |
| BGCheck | PK | BG\_ID | varchar | 10 |
| BGCheck |  | DateSubmited | Date | 10 |
| BGCheck |  | Pass | varchar | 10 |
| District | PK | DistrictID | varchar | 10 |
| District |  | DistrictName | varchar | 20 |
| District | FK | SchoolID | varchar | 10 |
| School | PK | SchoolID | varchar | 10 |
| School |  | SchoolName | varchar | 20 |
| DistrictSchool | Composite key(PK, FK) | DistrictID | varchar | 10 |
| DistrictSchool | Composite key(PK, FK) | SchoolID | varchar | 10 |

## Appendix H

### Timeline

#### Design Phase

**WEEK 1**

Find Clients.

**WEEK 2**

Met with prospective clients

Select KidSports as our proposed client

Propose project concepts

Client / Project accepted

**WEEK 3**

Met with KidSports

Gather requirements

Discuss current processes

Refine scope of application

**WEEK 4**

Primary / Secondary Research

Course Requirements

Background check API

HR processes

SM processes

Met with kidsports

Compare our understanding of their process to their needs

Formalize requirements

**WEEK 5**

Design decisions

Modeling - Database and Class architecture

Diagramming

Definitions

**WEEK 6**

Present application system

#### Development Phase

**WEEK 7**

UI Models

Create interfaces

Create shared layouts

Create partial views

Meet with kidsports

Demonstrate application UX/UI

**WEEK 8**

Design user stories

Assign tasks for users

Template UI Designs

Create styles HTML templates for UI

Add bootstrap and custom css

**WEEK 9**

Create domain models and repositories

Setup DBcontext using EF, including Identity

Registering

Login

**WEEK 10**

Controllers

Dependency Injection

Unit Testing

**WEEK 11**

UI / View Models

Add razor within UI’s

Add view models pertaining to actions

**Meet (First Week of May)**

**WEEK 12**

Implement the CRIS API

Encode strings of data to XML

Decode XML to strings

Make calls to CRIS servers

**Meet (Second Week of May)**

**WEEK 13**

Refactoring

Complete any tasks that were not successfully completed on schedule.

Refactor to OOP best practices

**BETA RELEASE**

**Meet (Third Week of May)**

**WEEK 14**

User testing

Test all functionality.

Attempt to exploit the website.

Look for distorted UI.

**WEEK 15**

Bug Fixing

Correct any UI mistakes.

Improve UX.

Patch exploits.

**WEEK 16**

**Meet (First Week of June)**

Meet with kidsports

Run through the application

Demonstrate UI and functionality

Implement UI suggestions

**WEEK 17**

Refactoring

Fix any last minute bugs

Implement any content that was not finished on schedule

**WEEK 18**

Present production release

**Meet (Second Week of June)**

**PRODUCTION RELEASE**

## Appendix I

### System Sequence Diagram

