

فصل یک

مفاهیم اولیه

## اینا رو همین طوری رها نکن. فرض کن اصلا در مورد هیچ کدوم اطلاع نداری.

## الف. تعاریف

1. تعاریف اولیه
   1. مزرعه: یک واحد دامداری
   2. گله: یک دسته دام شیری در یک مزرعه. یک مزرعه می تواند یک یا بیشتر گله داشته باشد
   3. گروه دام: هر گروه دام متشکل از دام هایی با وضعیت "تولید شیر" یکسان و مشخص هستند (پیش فرض طبق تعریف جدول بخش ه)
      1. گوساله Calf –
      2. تلیسه – Heifer
      3. تازه زا Fresh
      4. سوپر (very high producing)
      5. پر تولید (high producing)
      6. متوسط تولید (mid producing)
      7. کم تولید (low producing)
      8. خشک (dry-off)
      9. حذفی (discharged)
   4. رویداد: رویداد هایی که برای یک دام روی میدهد ( مثال : لقاح ، زایمان ، درمان ، بیماری خاص و...)
   5. هشدار یا آلارم: وضعیت غیرعادی در دام یا بهاربند که سیستم آنرا بطور خودکار شناسایی می کند. هشدارها در یک لیست ارایه می شود.
   6. اعلان یا نوتیفیکیشن: اطلاع رسانی آلارمها که کاربران با توجه به نقششان در سیستم دریافت میکنند. این اعلانها می تواند بصورتهای متنوعی مثل نوتیفیکیشن موبایل، ایمیل و اس ام اس باشد.
   7. زون: یک زون ناحیه مورد توجه برای موقعیت دام در مزرعه است. بطور پیش فرض، زونها شامل بهاربندها و شیردوشها می شود.
   8. اطلاعات شناسنامه ای دام: مجموعه اطلاعات دام که با تولد و یا خرید آن و اضافه شدن به مزرعه در سیستم درج می شود.
2. کاربران فارم (نقشهای معمول در فارم)
   1. مدیر سطح یک (مدیرعامل یا جانشین): حداکثر دسترسی کاربران فارم
   2. مدیر سطح دو (مدیر مزرعه): برخی دسترسی های سطح بالای مدیر سطح یک را ندارد (عمدتا در زمینه ویرایش و حذف)
   3. دامپزشک: درج اطلاعات محدود به موارد پزشکی است.
   4. پرسنل سطح یک: امکان درج و ویرایش بخش عمده ای از اطلاعات را دارد
   5. پرسنل سطح دو: امکان مشاهده اطلاعات را دارد
3. کاربران دفتر اسمارت کتل (نقشهای معمول در دفتر)
   1. مدیر: دسترسی کامل به اطلاعات کلیه مزارع (درج، ویرایش، حذف، مشاهده)
   2. پرسنل: دسترسی کامل به اطلاعات کلیه مزارع (فارم ها) (مشاهده)
4. اطلاعات شناسنامه ای دام
   1. شماره شناسنامه Cattle ID
   2. نام Name
   3. جنسیت Sex
   4. شماره شناسنامه مادر (اگر عضو این مزرعه است) Mother ID
   5. تاریخ تولد DoB
   6. ژنتیک: نوع و شماره اسپرم Genetics: Sperm Type and Number
   7. تعداد زایمان (به روز می شود) Lactation Number
   8. آخرین تاریخ زایمان (به روز می شود) Last Calving Date
5. اطلاعات پایه دام (آخرین تاریخ به روزرسانی نیز درج شود)
   1. شناسه حسگر Sensor ID
   2. گله Herd
   3. گروه Group
   4. بهاربند Barn
   5. آخرین "وضعیت دام"
   6. امتیازات دام Cattle Score
6. وضعیت دام: Cattle Status
   1. وضعیت سلامتی: Health status

* سالم Healthy
* مشکوک به بیماری suspicious
* بیمار sick
  1. وضعیت باروری:status Fertility
* تلقیح شده inseminated
* آبستن pregnant
* باز (پس از زایش) open
  1. وضعیت فحلی: Heat status
* عادی normal
* مشکوک suspicious
* فحل In-heat
  1. روزهای باز Days Open
  2. روزهای شیردهی Days in Milk

1. امتیازات دام

* شرایط بدنی Body conditioning score
* تمیزی Cleanliness
* هوک Hock
* حرکتی Mobility
* مدفوع Manure
* شکمبه Rumen
* سرپستان Teat
* تولید شیر Milk production (این در واقع یک امتیاز نیست، بلکه مکانیزمی برای ورود موردی شیر تولیدی برای دامداری هایی است که میلکمتر یکپارچه با اسمارت کتل ندارند)

1. رویدادها
   1. وضعیت فحلی - مشکوک به فحلی
   2. وضعیت فحلی – فحلی
   3. وضعیت فحلی – عادی
   4. وضعیت سلامتی - مشکوک به بیماری
   5. وضعیت سلامتی – سالم
   6. وضعیت سلامتی – بیمار
   7. وضعیت باروری – باز Fertility – Open
   8. وضعیت باروری – تلقیح شده Fertility – Inseminated
   9. وضعیت باروری – آبستن Fertility - Pregnant
   10. دامپزشکی – معاینه Vet – Examination
   11. دامپزشکی - اقدام درمانی Vet – Treatment
   12. دامپزشکی – زایش Vet – Calving
   13. انتساب به گله (از لیست گله ها) Assign herd
   14. انتساب به گروه (از لیست گروهها) Assign group
   15. انتقال به بهاربند (از لیست بهاربندها) Move to barn
2. اطلاعات شناسنامه ای ادوات هوشمند مزرعه
   1. شماره سریال
   2. آدرس مک
   3. نوع دستگاه:

* تگ هوشمند
* سنسور محیطی
* ادوات ارزیابی مشخصه های دام
* ادوات شبکه
  1. مدل دستگاه:
* تگ: گردنبند یا گوشواره، باتری یا سولار
* سنسور محیطی: رطوبت، دما، دما و رطوبت، ...
* ادوات ارزیابی مشخصه های دام: میلکمتر، دوربین BCS، فورس پلیت، ...
* ادوات شبکه: دی جی ان رزبری زیرو، دی جی ان رزبری سه، روتر مدل یک، ...

## ب. ورودی ها

### ورودی از سنسورها-آی پارس

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| شماره | عنوان | شرح | نرخ داده | محدوده و فرمت |
| 1 | ادوات هوشمند مزرعه | وضعیت عملکرد یا باتری  تاریخ فعال شدن  آخرین نسخه نرم افزار بارگذاری شده |  | تگها، سنسورهای محیطی، ادوات ارزیابی، ادوات شبکه |
| 2 | تگ-دمای دام | دمای اندازه گیری شده از گوش دام | 15 تا 60 دقیقه یکبار | اعشاری – بین 36 تا 40 |
| 3 | تگ-بودجه زمانی فعالیت | درصد زمانی که دام صرف (1) نشستن، (2) راه رفتن، (3) نوشخار کردن، (4) خوردن، (5) ایستادن (6) نوشیدن می کند | 15 تا 60 دقیقه یکبار | درصد – صفر تا صد  گرد شده به یکان |
| 4 | تگ-مکان دام | موقعیت دام در بهاربند | هر دقیقه | طول و عرض جغرافیایی |
| 5 | سنسور محیطی-دمای محیط | دمای اندازه گیری شده در سایه | 30 تا 60 دقیقه یکبار | اعشاری بین -50 تا +50 |
| 6 | سنسور محیطی-رطوبت محیط | رطوبت اندازه گیری شده در ارتفاع 2 متر بالاتر از دام | 30 تا 60 دقیقه یکبار | درصد – اعشاری بین 0 تا 200 |
| 7 | ادوات هوشمند-ادوات ارزیابی | مشخصه مورد ارزیابی دام مربوطه |  | مثل میلکمتر، دوربین بی سی اس |

## ورودی سیستمی

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| شماره | عنوان 1 | عنوان 2 | شرح | توضیحات دیگر |
| 1 | دام |  |  |  |
| 1-1 |  | شناسنامه دام | -بارگذاری فایل شناسنامه دام براساس فرمت معین برای یک گله  -ممکن است اطلاعات وضعیت دام نیز در این فرمت گنجانده شود | -درج  -بر اساس خروجی دریافتی از نرم افزار مدیران یا بنی اسدی یا وست فالیا |
|  |  | حجم شیر تولیدی | -بصورت یک فایل از سیستم وستفالیا |  |
| 2 | ادوات هوشمند |  |  |  |
| 2-1 |  | تگهای هوشمند | بارگذاری فایل شناسنامه تگ براساس فرمت معین برای یک مزرعه |  |
| 8-2 |  | سنسورهای محیطی | بارگذاری فایل اطلاعات سنسور محیطی براساس فرمت معین برای یک مزرعه | اطلاعات:  اطلاعات شناسنامه ای  انتساب سنسور به مزرعه-زون (بهاربند یا شیردوشی)  موقعیت سنسور |
| 8-3 |  | ادوات ارزیابی مشخصه های دام | بارگذاری فایل اطلاعات ادوات ارزیابی براساس فرمت معین برای یک مزرعه | مثل دوربین بی سی اس، میلکمترها، ...  اطلاعات:  اطلاعات شناسنامه ای  در مورد میلکمتر موقعیت آن در شیردوشی مهم است. |
| 8-4 |  | ادوات شبکه | بارگذاری فایل اطلاعات ادوات شبکه براساس فرمت معین برای یک مزرعه | اطلاعات:  اطلاعات شناسنامه ای  انتساب ادوات به مزرعه-زون (بهاربند یا شیردوشی)  موقعیت ادوات |

### ورودی دستی

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| شماره | عنوان 1 | عنوان 2 | شرح | توضیحات دیگر |
| 1 | تنظیمات |  |  |  |
|  |  | مزرعه | هر مزرعه در ابتدا با یک اکانت اصلی ایجاد می شود.  یک اکانت اصلی ممکن است به بیش از یک مزرعه مرتبط باشد. بطور مثال دامپروری صفاری دو مزرعه دارد. |  |
| 1-1 |  | گله | یک گله متناظر به یک مزرعه است. | -درج، ویرایش، حذف یک گله  -حذف تنها زمانی مقدور است که هیچ دامی در این گله نباشد |
| 1-2 |  | گروه | گروه  یک گله دامی به گروههای معینی تقسیم می شود. این گروه ها بطور اختصاصی برای هر مزرعه تعریف می شود. | -درج، ویرایش، حذف یک گروه  -حذف تنها زمانی مقدور است که هیچ دامی در این گله-گروه نباشد |
| 1-3 |  | بهاربند | مزرعه-بهاربند  هر مزرعه متشکل از تعدادی بهاربند است. هر بهاربند (بخشی از) یک گروه دامی را در خود دارد. | -درج، ویرایش، حذف یک بهاربند  -حذف تنها زمانی مقدور است که هیچ دامی در این مزرعه-بهاربند نباشد |
| 1-4 |  | نقش | مزرعه-نقش  نقش های مختلف و دسترسی ایشان | -درج، ویرایش، حذف یک نقش  نقشهای یک مزرعه ممکن است با مزرعه دیگر متفاوت باشد  نقشهایی برای بک آفیس هم تعریف می شود |
| 1-5 |  | اکانت | مزرعه-نقش-اکانت  اکانت ها در نقش های معین برای هر مزرعه تعریف می شود | -درج، ویرایش، حذف یک اکانت |
| 1-6 |  | سیستم |  |  |
|  |  |  | زبان پیش فرض | فارسی یا انگلیسی |
|  |  |  | سیستم اندازه گیری  متریک یا استاندارد | دمای سلسیوس یا فارنهایت  وزن به کیلوگرم یا پوند  ... |
|  |  |  | ناحیه زمانی | تایم زون |
| 2 | هشدار |  |  |  |
| 2-1 |  | تنظیم هشدار  مزرعه-گروه-هشدار | هشدارها به صورت شرایطی بر مقادیر متغیرهای سلامت دام باملاحظه سابقه خود ایشان یا سابقه دامهای گروه یا بهاربند تنظیم می شود. همچنین نقشهایی که یک هشدار را دریافت می کنند نیز در اینجا تنظیم می شود. | -درج، ویرایش، حذف |
| 2-2 |  | مدیریت هشدار  هشدار-نقش | هشدارها به افراد ذیربط ارسال می شود و ایشان اقداماتی را انجام می دهند و در سیستم درج می کنند. شیوه مدیریت هشدارها در طرح مربوطه آمده است. |  |
| 3 | دام |  |  |  |
| 3-1 |  | شناسنامه دام | اطلاعات شناسنامه ای دام  -با تولد یک گوساله  -با خرید یک گاو جدید  -ورودی دستی برای مزرعه ای که نرم افزار مشمول بند 1-7 ندارد | -درج، ویرایش، حذف |
| 3-2 |  | رویداد | زایمان، درمان، لقاح،... | -درج، ویرایش، حذف  تقویم و ساعت لازم است |
| 3-3 |  | امتیازات دام | -امتیاز بدنی، تمیزی و...  -شیر تولیدی | -درج، حذف  تقویم و ساعت لازم است |
| 3-4 |  | گله، گروه و بهاربند | دراپ دان منو | درج می شود و جایگزین قبلی می شود  مثل رویدادها به همراه زمان انجام آن در لیست رویدادها تحت عنوان رویداد تغییر گله، تغییر گروه و تغییر بهاربند درج می شود. |
| 4 | ادوات هوشمند |  |  |  |
| 4-1 |  | تگهای هوشمند | انتساب تگ به گله-دام و تاریخ آن | درج، ویرایش، حذف انتساب |
| 4-2 |  | سنسورهای محیطی | انتساب سنسور به مزرعه-زون (بهاربند یا شیردوشی) و تاریخ آن  موقعیت سنسور | درج، ویرایش، حذف: انتساب و موقعیت |
| 4-3 |  | ادوات ارزیابی مشخصه های دام | انتساب ادوات به مزرعه و تاریخ آن | مثل دوربین بی سی اس، میلکمترها، ...  درج، ویرایش، حذف: انتساب و موقعیت (راهرو و ردیف در مورد میلکمتر) |
| 4-4 |  | ادوات شبکه | انتساب ادوات به مزرعه-زون (بهاربند یا شیردوشی) و تاریخ آن  موقعیت ادوات | درج، ویرایش، حذف انتساب و موقعیت |

## ج. انواع خروجی

### گزارش دام

|  |  |  |  |
| --- | --- | --- | --- |
| شماره اجزا | عنوان | شرح | توضیحات دیگر |
| 1 | خلاصه اطلاعات پرونده دام | -اطلاعات شناسنامه ای دام  - اطلاعات پایه دام |  |
| 2 | دمای دام | گزارش دمای دام در بازه زمانی مشخص |  |
| 3 | فعالیت دام | گزارش فعالیت دام در بازه زمانی مشخص |  |
| 4 | موقعیت دام | موقعیت فعلی دام |  |
| 5 | رویداد ها | رویداد هایی که برای دام رخ داده | تلقیح ، زایش ، درمان و.... |
| 6 | امتیازات | امتیازات درج شده برای دام |  |
| 7 | شیر تولیدی | گزارش شیر تولیدی دام در بازه زمانی مشخص | منحنی مثل دمای دام |

### گزارش جمعی

|  |  |  |  |
| --- | --- | --- | --- |
| شماره | عنوان | شرح | توضیحات دیگر |
| 1 | گزارش خروجی | -انتخاب روی مزرعه، گله، گروه بهاربند  -انتخاب روی فیلدهای مربوط به یک دام  -تولید یک گزارش خروجی به فرمت CSV |  |

### گزارش وضعیت بهاربند

|  |  |  |  |
| --- | --- | --- | --- |
| شماره | عنوان | شرح | توضیحات دیگر |
| 1 | گزارش THI | گزارش تغییرات THI برای یک بهاربند در بازه زمانی مشخص |  |
| 2 | گزارش توزیع مکانی | هیت مپ مکان دامها در بازه زمانی مشخص برای یک بهاربند |  |
| 3 | گزارش زمان شیردوشی | متوسط زمان انتظار و انجام شیردوشی با تحلیل داده های مکانی برای گاوهای یک بهاربند | منحنی شبیه منحنی دمای دام  محور افقی روز است |

### گزارش ادوات هوشمند

|  |  |  |  |
| --- | --- | --- | --- |
| شماره اجزا | عنوان | شرح | توضیحات دیگر |
| 1 | ادوات هوشمند مزرعه | به شرح بخش ز |  |

## د. آلارم ها

### جدول آلارم ها

* هر ردیف در جدول آلارم ها نشان دهنده یک آلارم است.
* برای هر آلارم، اقدامات زیر ممکن است صورت پذیرد:
  + رسیدگی، غیرفعال کردن، فعال کردن مجدد.
  + Acknowledgement, Activate, De-activate
* ستونهای جدول شامل موارد زیر است:

1. شناسه آلارم (یک عدد یکتا که خود سیستم ایجاد می کند) Alarm ID
2. شناسه سوژه (شماره دام یا بهاربند) Subject ID
3. نوع آلارم (دامی یا بهاربند) Alarm Type (Cattle, Barn)
4. عنوان آلارم Subject
5. پیام Description
6. وضعیت آلارم (برقرار یا برطرف شده) Alarm State (Active, Inactive)
7. زمان رخداد Time Issued
8. زمان رسیدگی (در صورت تهی بودن یعنی رسیدگی نشده) Ack Time
9. زمان غیرفعال شدن (در صورت تهی بودن یعنی فعال است) Deactivation time
10. شرح اقدام (متنی که فرد اقدام کننده در آخرین اقدام وارد کرده است) Note
11. نام اقدام کننده (نام آخرین کاربری که روی یک آلارم اقدام کرده: رسیدگی یا غیرفعال کردن/فعال کردن مجدد) User

* با انتخاب هر ردیف جدول آلارم ها، تاریخچه اقدامات انجام شده برای یک آلارم ارایه می شود. این تاریخچه در قالب یک جدول مثل جدول بالا عرضه می گردد با این تفاوت که هر ردیف آن مربوط به یک اقدام بر آلارم انتخاب شده است.
* ایجاد آلارم ها براساس شرایطی که در تعریف آنها در نظر گرفته شده است، صورت می پذیرد.

**گردش کاری آلارم ها**

* یک - آلارم ایجاد شد و رسیدگی شد، پرونده بسته می شود.
* دو -آلارم ایجاد شد و رسیدگی نشد، باز می ماند.
* سه -پس از رسیدگی به یک آلارم، به اندازه پنجره Resolution Margin این نوع آلارم برای این گاو یا این بهاربند تولید نمی شود. بعد از گذشت این پنجره، و پس از محاسبه مجدد شرایط آلارم در موعد مقرر، در صورت برقراری شرایط، یک آلارم جدید تولید شود.
* چهار -در صورتی که یک آلارم غیرفعال شود، این نوع آلارم برای این گاو یا بهاربند تا زمانی که مجدد فعال شود، ایجاد نمی شود.
* در اینجا دو مورد هست که در وضعیت آلارم و نکات بالا اثری نداشته است:
  + یک - اینکه پس از صدور آلارم و در بازه RM، شرایط آلارم هنوز برقرار است یا خیر. اگرچه در حالت دوم، این موضوع قابل مشاهده هست.
  + دو - اندازه بازه اک : Acknowledgement margin
  + ولیکن از این دو مورد می توان برای نحوه صدور اعلانها (نوتیفیکیشن ها) بهره برد

## ه- رویدادها و ارتباط بین ورودی ها و خروجی ها

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| ردیف | عنوان متغیر | نوع متغیر | نوع داده | شرح |
| 1 | وضعیت فحلی - مشکوک  Heat status– suspicious | مستقل | ورودی - رویداد – درج دستی و احتمالا پس از دریافت آلارم مربوطه | درج این رویداد، وضعیت فحلی را به مشکوک تبدیل می کند.  \*بعد از "دوره بررسی فحلی" که مقدار پیش فرض آن 48 ساعت است وضعیت فحلی به عادی بر می گردد. البته در این بین اگر رویداد فحلی درج شود طبیعتا به حالت فحلی می رود. |
| 2 | وضعیت فحلی - فحل  Heat status – In-heat | مستقل | ورودی - رویداد – درج دستی و احتمالا پس از دریافت آلارم مربوطه | درج این رویداد، وضعیت فحلی را به فحل تبدیل می کند.  \*بعد از "دوره بررسی فحلی" که مقدار پیش فرض آن 48 ساعت است وضعیت فحلی به عادی بر می گردد. |
| 3 | وضعیت فحلی - عادی  Heat status – Normal | مستقل | ورودی - رویداد – درج دستی | درج این رویداد، وضعیت فحلی را به عادی تبدیل می کند. |
| 4 | وضعیت سلامتی -سالم  Health Status – Healthy | مستقل | ورودی - رویداد – درج دستی | درج این رویداد، وضعیت سلامت را به سالم تبدیل می کند |
| 5 | وضعیت سلامتی -مشکوک  Health Status – suspicious | مستقل | ورودی - رویداد – درج دستی | درج این رویداد، وضعیت سلامت را به مشکوک تبدیل می کند |
| 6 | وضعیت سلامتی - بیمار  Health Status – Sick | مستقل | ورودی - رویداد – درج دستی | درج این رویداد، وضعیت سلامت را به بیمار تبدیل می کند |
| 7 | وضعیت باروری – تلقیح شده  Fertility – Inseminated | مستقل | ورودی - رویداد – درج دستی | درج این رویداد، وضعیت باروری را به تلقیح شده تبدیل می کند |
| 8 | وضعیت باروری - آبستن  Fertility – Pregnant | مستقل | ورودی - رویداد – درج دستی | درج این رویداد، وضعیت باروری را به آبستن تبدیل می کند |
| 9 | وضعیت باروری - باز  Fertility – Open |  | ورودی - رویداد – درج دستی | درج این رویداد، وضعیت باروری را به باز تبدیل می کند. |
| 10 | دامپزشکی – معاینه  Vet - Examination | مستقل | ورودی - رویداد – درج دستی |  |
| 11 | دامپزشکی - اقدام درمانی  Vet – Treatment | مستقل | ورودی - رویداد – درج دستی | تزریق، جراحی، ... |
| 12 | دامپزشکی - زایش  Vet – Calving | مستقل | ورودی - رویداد – درج دستی | \*درج این رویداد، وضعیت باروری را به باز تبدیل می کند.  \*درج این رویداد، وضعیت گروه را به تازه زا تبدیل می کند. |
| 13 | انتساب به گله (از لیست گله ها) Assign herd | مستقل | ورودی - رویداد – درج دستی | درج این رویداد، شماره گله را برای دام به روز می کند |
| 14 | انتساب به گروه (از لیست گروهها) Assign group | مستقل | ورودی - رویداد – درج دستی | درج این رویداد، شماره گروه را برای دام به روز می کند |
| 15 | انتقال به بهاربند (از لیست بهاربندها) Move to barn | مستقل | ورودی - رویداد – درج دستی | درج این رویداد، شماره بهاربند را برای دام به روز می کند |
|  |  |  |  |  |
| 16 | گروه (وضعیت تولید شیر)  Group | دو مورد اول وابسته  بقیه موارد در صورت نبود میلکمتر بطور مستقل و دستی درج می شود | ورودی – خروجی | # یک - گوساله Calf – با تولد در این گروه است  # دو - تلیسه Heifer  \*گوساله با گذشت زمان بطور خودکار به این گروه منتقل می شود.  # سه - تازه زا Fresh  چهار - سوپر (very high producing)  پنج - پر تولید (high producing)  شش - متوسط تولید (mid producing)  هفت - کم تولید (low producing)  هشت - خشک (dry-off)  # نه - حذفی (Discharged) |
|  |  |  |  |  |
| 17 | وضعیت باروری  Fertility Status | وابسته | خروجی | تلقیح شده inseminated  آبستن pregnant  باز (پس از زایش) open |
| 18 | وضعیت فحلی  Heat Status | وابسته | خروجی | عادی normal  مشکوک suspicious  فحل In-heat |
| 19 | وضعیت سلامت دام | وابسته | خروجی | سالم healthy  مشکوک suspicious  بیمار sick |
| 20 | روزهای باز | وابسته – عدد طبیعی | خروجی | -اگر آخرین وضعیت باروری باز یا تلقیح شده است:   * اگر تابحال زایمان داشته: از زمان آخرین زایمان تا امروز محاسبه می شود. * اگر تابحال زایمان نداشته: از تاریخ تولد تا امروز.     -اگر آخرین وضعیت باروری آبستن است:  خط تیره نمایش داده می شود. |
| 21 | روزهای شیردهی | وابسته – عدد طبیعی | خروجی | -اگر تابحال زایمان نداشته: خط تیره نمایش داده می شود.  -اگر تابحال زایمان داشته:   * اگر در هر گروهی غیر از خشک یا حذفی است: از زمان آخرین زایمان تا امروز محاسبه می شود. * اگر در گروه خشک یا حذفی است: خط تیره نمایش داده می شود. |
| 22 | تعداد شکم زایش | وابسته – عدد طبیعی | خروجی | مقدار اولیه در شناسنامه درج می شود  با درج هر رویداد زایمان یکی اضافه می شود  با حذف هر رویداد زایمان یکی کم می شود |
| 23 | تاریخ آخرین زایمان | وابسته | خروجی | مطابق تاریخ آخرین زایمان درج شده  اگر موردی نیست خط تیره نمایش داده می شود |

## و- مدیریت حساب کاربری

1. **اطلاعات حساب کاربری:**

* نام کاربری
* گذر واژه
* نام و نام خانوادگی
* جنسیت
* زمینه تخصصی
* آدرس و اطلاعات تماس
* عکس پروفایل

1. **موارد کاربرد- مدیریت حساب کاربران:**

* ایجاد حساب کاربری
* مشاهده و ویرایش
* انتساب نقش
* کنترل دسترسی (فعال-غیرفعال)

1. **مثالها:**

* کاربر دفتر اسمارت کتل برای یک مزرعه جدید حساب کاربری اصلی ایجاد میکند.
* مدیر مزرعه برای کارکنان مزرعه حساب کاربری ایجاد می کند.
* مدیر مزرعه برای یک دامپزشک که قبلا در اسمارت کتل حساب کاربری دارد، نقش جدید در مزرعه خودش تعریف می کند و به او دسترسی می دهد.

1. **ملاحظات:**

* برای نقش های مختلف می توان امکان ایجاد و ویرایش و نقش و....(موارد کاربرد) را در نظر گرفت یا نگرفت.
* نوعاً ویرایش پروفایل (اطلاعات حساب) می بایست توسط فرد صاحب حساب یا کارمند دفتر اسمارت کتل مقدور باشد.
* گذر واژه اولیه می تواند یک فیلد اختیاری باشد یا حذف شود. ایجاد حساب در یک سامانه یا انتساب نقش می بایست به نوعی تأیید کاربر مربوطه را داشته باشد. بنابراین در حالت کلی مدل ارسال دعوت توسط ایمیل و تایید آن مدل بهتری است.
* تغییر پسورد فقط از طریق ایمیل مقدور است.
* اضافه کردن یک کاربر به یک مزرعه (مثلا توسط مدیر مزرعه) می تواند با جستجوی نام کاربری یا ایمیل آغاز شود که مشخص شود قبلا حساب کاربری در اسمارت کتل دارد یا نه (مثل ترلو).
* بر روی طرح ایجاد رول سیستمی کارمند و رول سیستم فارم و رول فارم کار شد. به همین منظور در بخش تعریف Role یک گزینه اضافه شد که تعیین می کند Role که کاربر در حال تعریف آن هست از کدام نوع هست. اگر کاربر SuperAdmin باشد، می تواند سه نوع رل سیستمی کارمند، سیستمی فارم و فارم تعریف کند. البته این کاربر امکان انتخاب فارم همزمان با تعریف رل را دارد. اگر کاربر اسمارت کتل باشد بسته به این که اجازه تعریف رل را دارد یا نه، می تواند تنها یک نوع رل از نوع فارم و برای همان مزرعه ای که در آن قرار دارد تعریف کند. این به این معنی است که دراپ دان مربوط به انتخاب مزرعه و انتخاب نوع رل را که کارمند بک آفیس می بینه، نمی بیند مگر این که دو تا مزرعه داشته باشد. آن وقت تنها همین دوتا مزرعه رو در دراپ دان می بینه و می تونه بدون سوییچ برای هر کدوم که خواست رل تعریف کند.
* در ادامه کار برای هر سه نوع رل، ویژگی هایی تعریف شد که قیود ذیل را برآورده می کند.
* رل های سیستمی فارم در بک آفیس اسمارت کتل قابل انتساب به مزارع هستند ولی صاحب مزرعه نمی تواند از بین رل های سیستمی موردی رو برای خود یا اکانت دیگری انتخاب و انتساب نماید. مثل FarmManager\_AssetType\_1
* رل های سیستمی کارمند، از سوی مدیر مزرعه قابل انتساب به کاربری نیستند مثل Support\_Tracking
* رل های فارم فقط در داخل همان فارمی که تعریف شده قابل استفاده هست و در بیرون آن نه قابل انتساب به مزرعه دیگری هستند و نه قابل انتساب به کارمند بک آفیس هستند. مثل Vet

## ز- مدیریت ادوات (هوشمند) مزرعه

در حال حاضر ادوات هوشمند مزرعه عبارتند از:

* ادوات سنسوری (SD: Sensor Devices)
  + تگهای هوشمند Smart Tags
  + سنسورهای محیطی Environmental Sensors
  + میلکمتر Milkmeter
  + فورسپلیت Forceplate
  + دوربین اسکور بدنی Camera BCS
  + RFID Reader
* ادوات شبکه (ND: Network Devices)
  + Access Point
  + DGN: Data Gathering Node

مستقل از اینکه سمت اسمارت کتل چه اطلاعاتی از هر کدام از این ادوات نمایش داده شود، امکان تعریف و تخصیص این ادوات برای هر فارم باید با استفاده از لود شدن یک فایل Excel سمت آیپارس فراهم شود و اطلاعات مورد نظر نیز باید به طریقی مناسب در اختیار اسمارت کتتل نیز قرار داده شود و این دو قسمت با هم سینک باشند. فیلدهای این فایل به صورت زیر خواهد بود:

|  |  |  |  |
| --- | --- | --- | --- |
| Example1 | Example1 | Fields |  |
| SD | NE | Device category | 1 |
| Tag | DGN | Device type | 2 |
| Neck | Raspberry pi 0 | Device subtype | 3 |
| Raw |  | Packet type | 4 |
| W1 | W1 | Version | 5 |
| 123 | 123 | Equipment ID | 6 |
| A1:00:00:00:00:02 | A1:00:00:00:00:01 | MAC | 7 |
|  |  | Project ID | 8 |
|  |  | Sub-project ID | 9 |
|  |  | Zone ID | 10 |
|  |  | Location, X | 11 |
|  |  | Location, Y | 12 |
|  |  | Location, Z | 13 |
|  |  | Date 1: Manufacture | 14 |
|  |  | Date 2: Install | 15 |
|  |  | Reserved1: Cattle ID | 16 |
|  |  | Reserved 2 | 17 |

گزارش ادوات هوشمند Smart Devices

شامل ستونهای زیر است و در فرمتی شبیه لیست دام نمایش داده می شود

|  |  |  |
| --- | --- | --- |
| ملاحظات | Fields |  |
| امکان جستجو | Device category | 1 |
| امکان جستجو | Device type | 2 |
| امکان جستجو | Device subtype | 3 |
| امکان جستجو | Packet type | 4 |
| امکان جستجو | Version | 5 |
| امکان جستجو | Equipment ID | 6 |
| امکان جستجو | MAC | 7 |
| متناظر به شماره بهاربند  امکان فیلتر | Zone ID | 8 |
| آخرین تاریخ دریافت داده از این دستگاه در اسمارت کتل | Last updated | 9 |
| وضعیت باتری یا عملکرد (اگر از آی پارس برای این دستگاه دریافت می شود) | Status | 10 |
| امکان جستجو | Cattle ID | 11 |
|  | Installation Date | 12 |

ک - نمودار بودجه زمانی (Time Budget)

این نمودار از نظر محتوی داده، مبتنی بر داده "فعالیت دام" می باشد. ویژگیهای این نمودار به شرح زیر است:

این نمودار بصورت یک تب جدید به اطلاعات دام اضافه می شود. بنابراین دیگر نمودارهای دام به همین شکل فعلی باقی می ماند.

نمودار از نظر فرمت شبیه نمودار دما است

بازه زمانی نمایش، طبق تقویم انتخاب می شود

بازه متوسط گیری داده فعالیت برای نمایش عینا مثل قبل استفاده می شود

محور افقی نمودار زمان را نشان می دهد

محور عمودی بر مبنای صفر تا صد می باشد

سری زمانی مربوط به هرفعالیت نسبت به حداکثر مقدار خود در بازه منتخب نرمالیزه می شود (یعنی داده بر حداکثر تقسیم و در صد ضرب می شود)

دکمه هایی برای گزینش فعالیتهایی که می خواهیم آنرا نمایش دهیم درنظر گرفته می شود. بنابراین از مجموع شش فعالیت فقط موارد منتخب نمایش داده می شود. از رنگ هر یک از دکمه ها در نمایش منحنی متناظر استفاده می کنیم.

فصل دو

راهنمایی بر اسمارت کتل

مقدمه

اسمارت کتل یک نرم افزار مبتنی بر IoT می باشد که از دو بخش نرم افزار و سخت افزار تشکیل شده است. در کتابچه راهنمای اسمارت کتل تنها به توضیح بخش نرم افزار پرداخته شده است. در این بخش قرار است در مورد نحوه کار با نرم افزار اسمارت کتل توضیحاتی به همراه تصاویری از نرم افزار ارائه شود.

تصاویر استفاده شده از در این فصل، مطابق با Commit#108 گیت هاب به آدرس ذیل می باشد.

<https://github.com/DVD2/SmartCattlePA>

ساختار این فصل به این صورت هست که نرم افزار از دو دید مختلف بررسی می شود.

1. کارمند بک آفیس
2. کارمند عادی

جدای از این که هر یک از کاربران چه سطحی دسترسی به نرم افزار خواهند داشت، به دو دسته کلی تقسیم می شوند. دسته نخست مربوط به کاربرانی هستند که به آن ها "بک آفیس" گفته می شود. می توان موارد ذیل را به عنوان شرح وظایف کارمند بک آفیس بیان نمود.

1. ایجاد مزرعه متناسب با داده های دریافتی از کارشناسان اسمارت کتل
2. وارد کردن دام ها به مزرعه ایجاد شده
3. رفع مشکلات پیش آمده برای کاربران عادی نرم افزار اسمارت کتل
4. و غیره...

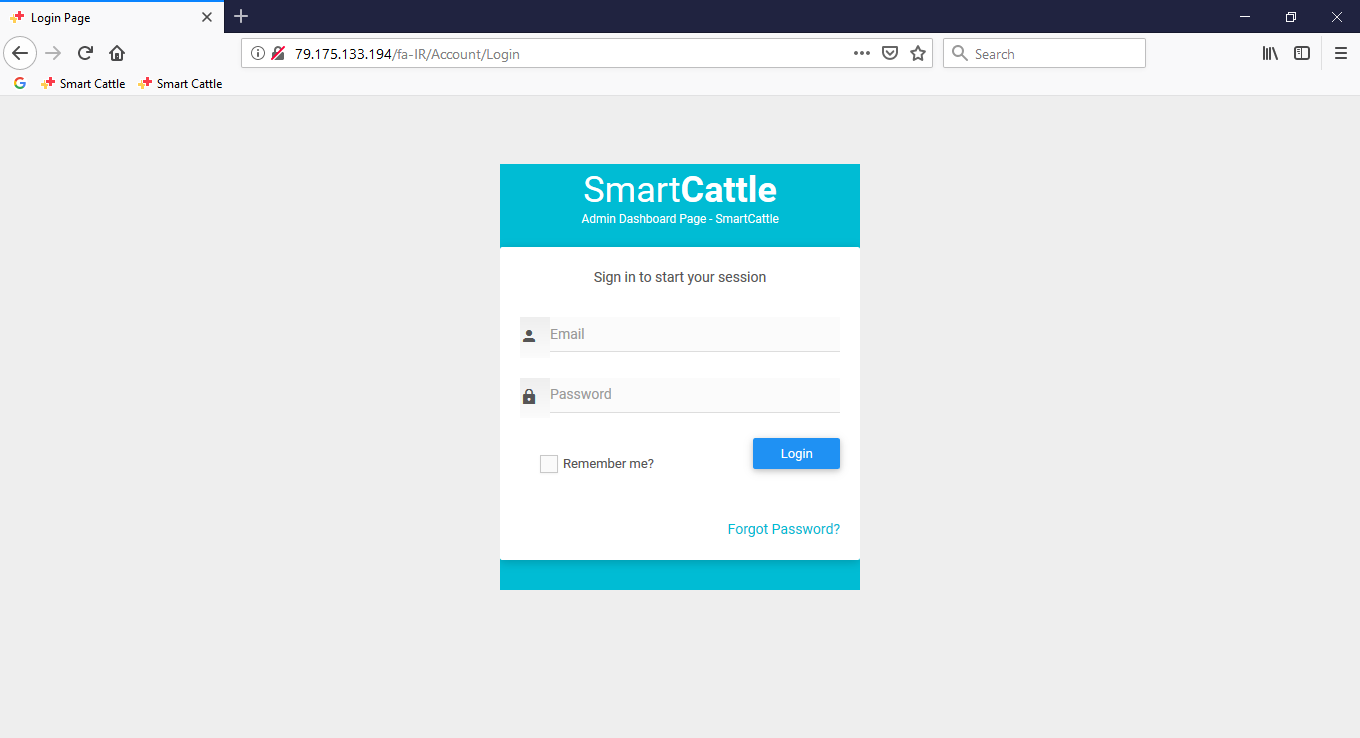
در اصل کارمند بک آفیس را می توان یک کارمند پشتیبان در نظر گرفت که دسترسی بالاتری نسبت به کاربر عادی نرم افزار دارد تا بتواند مشکلاتی که کاربر عادی حین کار با نرم افزار با آن رو به رو می شود را حل نماید.

به این ترتیب کاربر عادی را می توان صاحبان مزارع یا کارمندان آن ها در نظر گرفت که همگی در یک مزرعه مشغول به فعالیت می باشند و از نرم افزار اسمارت کتل برای مدیریت دام ها استفاده می نمایند.

در این فصل ابتدا از دید کارمند بک آفیس نرم افزار را مرور خواهیم کرد و این گونه قرار می گذاریم که تمام بخش های نرم افزار برای کاربران عادی و بک آفیس یکسان هست مگر آن که به طور مستقیم در خلال متن ها به تفاوت های بین آن ها اشاره شود. با این ذهنیت به مطالعه این فصل بپردازید که کارمند بک آفیس در حکم Parent و سایر کابران نظیر مدیران فارم، اپراتورها، کارمندان مزرعه و غیره Child محسوب می شوند.

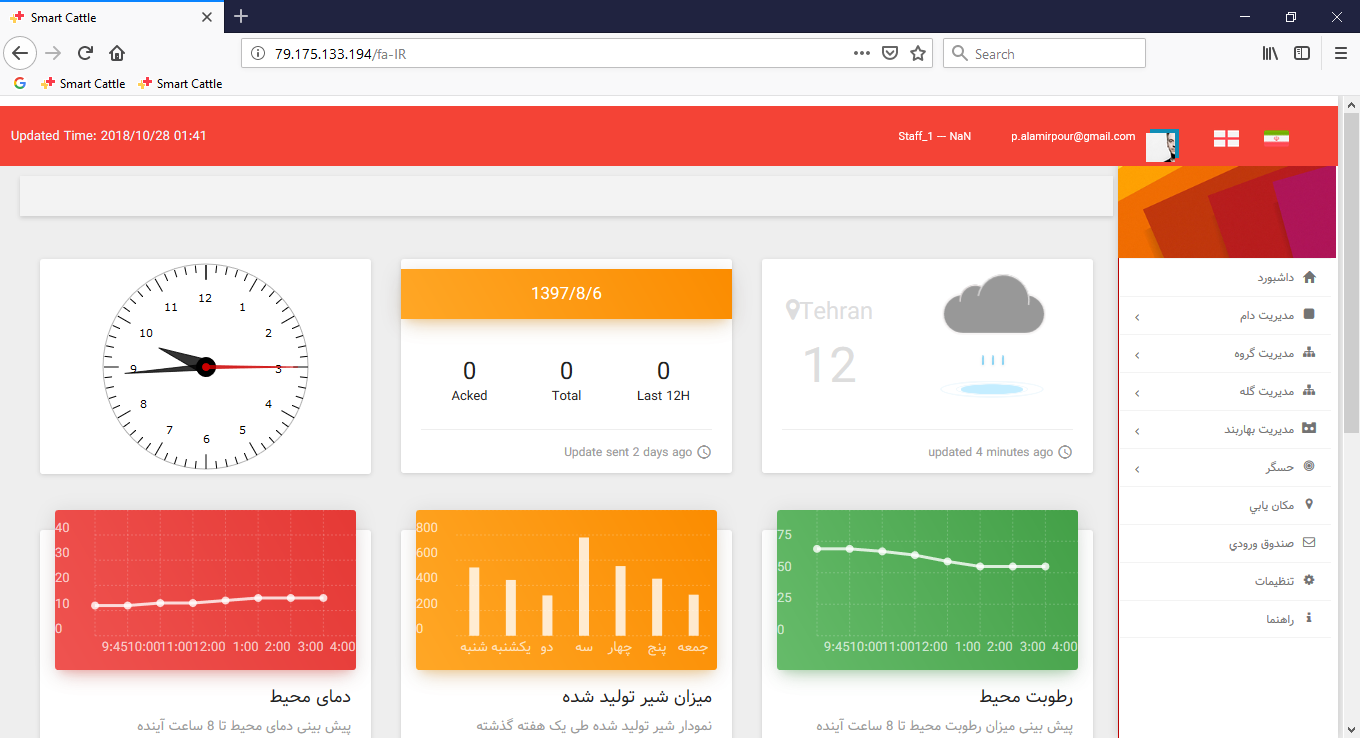
صفحه لاگین

در شکل 1- 1، صفحه لاگین نرم افزار اسمارت کتل را مشاهده می کنید. فرقی نمی کند که کاربری که می خواهد از نرم افزار استفاده کند یک کاربر عادی باشد یا کارمند بک آفیس. هر دو از طریق همین صفحه می توانند به نرم افزار لاگین شوند و از آن استفاده نمایند.



شکل 1- 1، صفحه لاگین نرم افزار اسمارت کتل

بعد از لاگین شدن کاربر، نوع کاربر توسط سیستم به صورت خودکار مشخص و سپس وی به داشبورد اسمارت کتل هدایت می شود. برای شروع به عنوان یک کاربر بک آفیس وارد نرم افزار می شویم. در شکل 1- 2، صفحه داشبورد را ملاحظه می کنید. این صفحه از بخش های مختلفی تشکیل شده است که هر یک از آن ها در کادری مجزا از یکدیگر تفکیک شده اند. برخی از این کادرها برای کاربران عادی ممکن است غیر فعال باشد. البته لازم به ذکر هست که در نرم افزار اسمارت کتل می توان ماژول های مختلف را از دسترس برخی نقش/اکانت ها خارج کرد. به عنوان مثال فرض کنید یک کاربر عادی از نوع دامپزشک وارد نرم افزار می شود به نحوی که وی زیرمجموعه کاربرعادی دیگری از نوع مدیر مزرعه هست. مدیر مزرعه نمی خواهد که یکی از زیر مجموعه های وی یعنی دامپزشک، دسترسی به نمودار میزان شیر تولید شده داشته باشد. وی می تواند به سادگی دسترسی نمودار مذکور را برای دامپزشک محدود نماید. در مورد کنترل دسترسی ها در آینده بیشتر توضیح داده خواهد شد.



شکل 1- 2، صفحه داشبورد

ابتدا نگاهی به نوار ابزار بالای نرم افزار بیاندازیم. در سمت راست دو آیکون پرچم مشاهده می شود. برای سوییچ بین دو زبان فارسی و انگلیسی از این آیکون ها استفاده می شود. در ادامه، آدرس ایمیل کاربری که به نرم افزار لاگین کرده مشاهده می شود. این آدرس در بین تمامی کاربران یکتا است. ممکن است مدیر یک مزرعه بیش از یک مزرعه داشته باشد و بخواهد تمامی آن ها را در نرم افزار مدیریت نماید. آدرس ایمیل مورد نظر برای ورود به تمامی مزرعه ها یکسان می باشد. این موضوع بدین معنی است که کاربر نیازی ندارد برای ورود به مزرعه های مختلف آدرس ایمیل های مختلفی را وارد نماید. با یک آدرس ایمیل می تواند به تمامی مزارع تحت کنترل خود دسترسی داشته باشد. این موضوع برای کاربری از نوع بک آفیس هم صدق می کند. هر تعداد مزرعه تحت پشتیبانی یک کاربر بک آفیس، تنها با وارد کردن یک و فقط یک آدرس ایمیل قابل استفاده برای لاگین، قابل دسترس هست.

در سمت چپ آدرس ایمیل درج شده در نوار بالای اسمارت کتل، دو عبارت با علامت – از هم مجزا شده اند. همانطور که در شکل 1- 2 پیداست، عبارت Staff\_1 به همراه عبارت NaN ذکر شده است. مفهوم این عبارت آن است که کاربری از نوع بک آفیس به نرم افزار لاگین شده است و مزرعه ای که هم اکنون نرم افزار اسمارت کتل در حال نمایش آن هست، NaN یا تهی هست. برای آن که این مفهوم را بهتر درک کنیم، بیایید نگاهی به مجموعه کارهایی که یک کاربر از نوع بک آفیس می تواند انجام دهد بیاندازیم.

می توان وظایف یک کاربر از نوع بک آفیس را به دو بخش کلی تقسیم کرد.

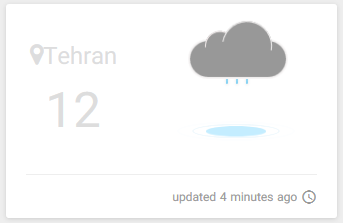
1. مجموعه کارهایی که مستقل از مزرعه قابل انجام هستند. مثل ایجاد مزرعه.
2. مجموعه کارهایی که وابسته به مزرعه هستند. مثل کارهایی که برای پشتیبانی مزرعه برعهده وی هست.

وقتی کارمند بک آفیس وارد مزرعه می شود، هنوز مشخص نیست که وی می خواهد وظایف نوع یک را انجام دهد یا نوع دو. پس به صورت پیش فرض وظایف از نوع یک برای وی فراخوانی می شود. برای آن که کاربر بک آفیس بتواند به وظایف از نوع دوم سوییچ کند باید از بخش تنظیمات و بخش انتخاب مزرعه، مزرعه مورد نظر خود که قصد پشتیبانی آن را دارد انتخاب نماید. در این رابطه در آینده بیشتر توضیح داده خواهد شد.

در این بخش از نرم افزار بین کاربر عادی و کاربر بک آفیس تفاوت وجود دارد. کاربر عادی وقتی وارد نرم افزار می شود، نیازی ندارد بین تسک نوع یک یا دو سوییچ کند. به همین منظور، نرم افزار در حالتی برای وی نمایش داده می شود که تمام داده های مورد نیاز برای مشاهده مزرعه وی، در آن لود شده است. پس وقتی کاربر عادی وارد نرم افزار می شود، عبارت نوار بالا، شامل نقش کاربر به همراه نام مزرعه خواهد بود. مثلا کاربر از نوع مدیر مزرعه – مزرعه کرج(نام مزرعه).

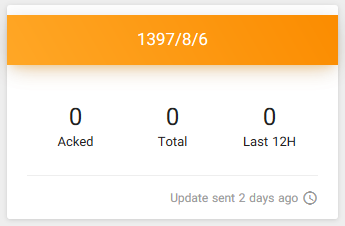
عبارت بالا سمت چپ نوار ابزار، اشاره به آخرین تاریخ آپدیت نرم افزار دارد. این تاریخ به صورت خودکار و با توجه به تاریخ ویرایش فایل های موجود در سرور فراخوانی می شود.

صفحه داشبورد شامل کارت های مختلف هست که هر یک اطلاعات مفیدی را به کاربر نمایش می دهند. هر یک از کارت ها بررسی خواهند شد.



شکل 1- 3، وضعیت آب و هوا

در شکل 1- 3، وضعیت آب و هوا جاری به همراه دمای هوا به کاربر نمایش داده می شود. اگر کاربر بک آفیس وارد نرم افزار شود، در حالت نمایش فارسی، وضعیت آب و هوای شهر تهران و در حالت نمایش انگلیسی شهر لس آنجلس نمایش داده می شود. ولی اگر کاربر بک آفیس یکی از مزرعه ها رو برای انجام کارهای پشتیبانی انتخاب کند، فقط وضعیت آب و هوای شهری که آن مزرعه در آن قرار دارد نمایش داده می شود. برای کاربر عادی، نمایش وضعیت آب و هوا بر اساس موقعیت و شهری خواهد بود که مزرعه در آن قرار دارد.



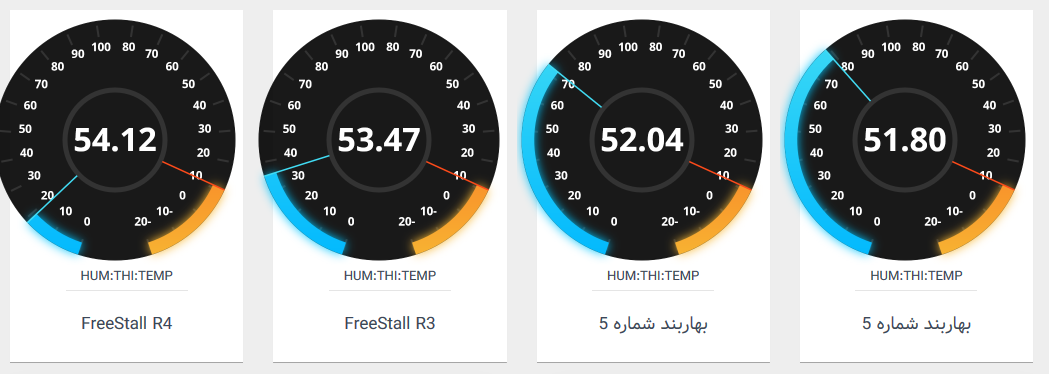
شکل 1- 4، نوتیفیکیشن های تولید شده

در شکل 1- 4، نوتیفیکیشن های تولید شده برای یک مزرعه قابل مشاهده هست. اگر کاربر به عنوان بک آفیس وارد نرم افزار شود، با توجه به این که در مزرعه خاصی هنوز لاگین نشده است، مقدار صفر را برای تمامی پارامتر های شکل 1- 4 مشاهده می کند. ولی اگر در مزرعه ای لاگین کند یا این که کاربر عادی وارد نرم افزار شود می تواند تعداد نوتیفیکیشن هایی که تا دوازده ساعت گذشته تولید شده اند، تعداد کل نوتیفیکیشن ها و تعداد نوتیفیکیشن های بررسی شده را مشاهده نماید. نوتیفیکیشن بخشی از هوش مصنوعی اسمارت کتل محسوب می شود که دام ها را بر اساس پارامترهای تعریف شده زیر نظر دارد. اگر معیارهای سلامتی و حیاتی دام از حد مشخص شده ای تجاوز کند، نرم افزار آن را شناسایی و نوتیفیکیشن مربوط به آن را برای اطلاع رسانی ایجاد می کند. همانطور که بعدا توضیح داده خواهد شد، نوتیفیکیشن ایجاد شده به بخش مربوطه جهت بررسی ارسال می شود.



شکل 1- 5، نمودار میزان رطوبت، میزان شیر تولیدی مزرعه در طول هفته و نمودار دمای محیط

در شکل 1- 5، نمودار میزان رطوبت، میزان شیر تولیدی مزرعه در طول هفته و نمودار دمای محیط را مشاهده می کنید. همان سناریویی که برای شکل 1- 3 بیان شد برای نمودارهای فوق الذکر هم صدق می کند.



شکل 1- 6، وضعیت دما، THI و میزان رطوبت بهاربندها

اگر کاربر عادی وارد نرم افزار شود، در صفحه داشبورد گجت های نمایش داده شده در شکل 1- 6 را مشاهده می کند. شکل 1- 6، وضعیت دما، THI و میزان رطوبت بهاربندها را مشخص می نماید. این ویجت ها برای کاربر بک آفیس تا مادامی که وارد مزرعه ای نشده باشد غیر فعال است. اگر مزرعه فرضا 20 بهاربند داشته باشد، در صفحه اصلی نرم افزار اسمارت کتل تنها، 4 ویجت که بیشترین میزان THI را دارد نمایش داده می شود. دلیل این امر، حساس و حیاتی بودن میزان THI است. پارامتر THI به صورت زیر محاسبه می شود:

|  |
| --- |
| THI = temperature - (0.55 - (0.55 \* humidity / 100)) \* (temperature - 58) |

آخرین پارامتری که در صفحه داشبورد می توان آن را ملاحظه کرد دو جدولی است که در انتهای صفحه قرار دارد. در این دو جدول قرار است اخبار صنفی و میزان قیمت اقلام مورد نیاز دامداری ها نمایش داده شود.

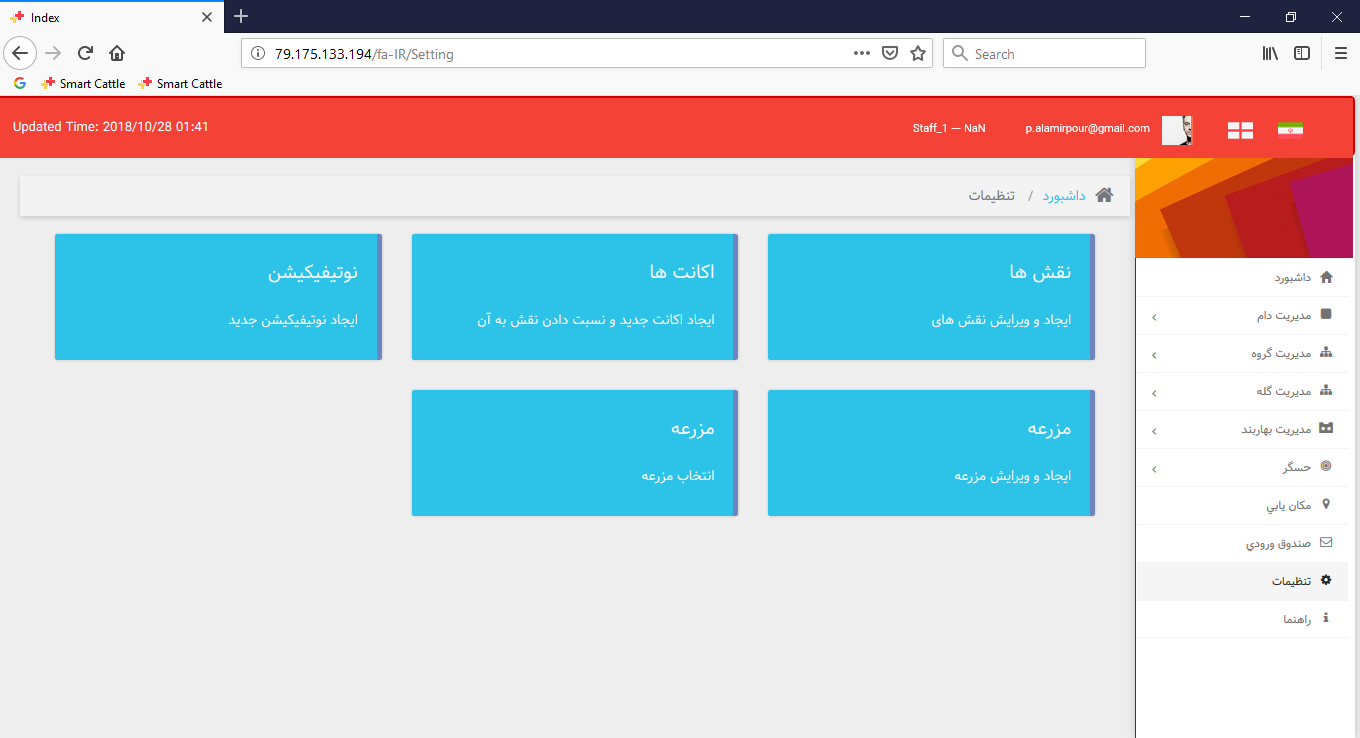


شکل 1- 7، اخبار صنف و قیمت اقلام ضروری دامداری ها

برای رفتن به بخش های مختلف نرم افزار از منوی سمت راست استفاده می شود. یکی از گزینه های مهم در این منو، بخش تنظیمات است که در ادامه در مورد آن مفصل صحبت خواهد شد.

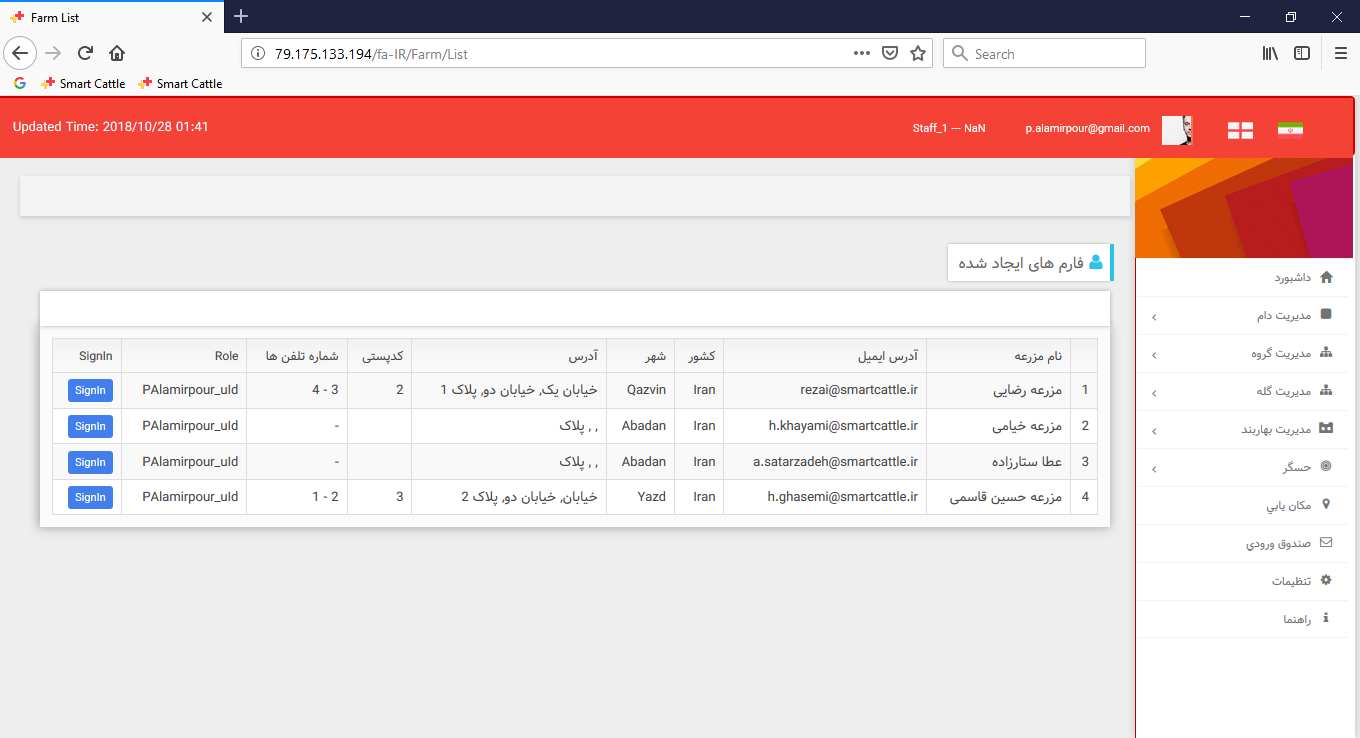
منوی تنظیمات

منوی نرم افزار اسمارت کتل از بخش های مختلفی تشکیل شده ولی به دلیل ذیل، ابتدا بخش تنظیمات برای توضیح انتخاب شده است. در توضیح بخش گذشته، بارها در مورد انتخاب مزرعه توسط کاربر بک آفیس یا ایجاد مزرعه صحبت شد. در حال حاضر و در ورژن فعلی نرم افزار، به نظر می رسد نخستین کاری که کاربر بک آفیس پس از ورود به نرم افزار انجام می دهد، رفتن به بخش تنظیمات و انتخاب یک مزرعه جهت انجام فعالیت های پشتیبانی است. از این رو به بخش تنظیمات می رویم. در شکل 1- 8، بخش تنظیمات را مشاهده می کنید.



شکل 1- 8، بخش تنظیمات

در شکل 1- 9، بخش انتخاب مزرعه را مشاهده می کنید. این بخش یکی از قسمت های پرکاربرد تنظیمات اسمارت کتل محسوب می شود.



شکل 1- 9، بخش انتخاب مزرعه

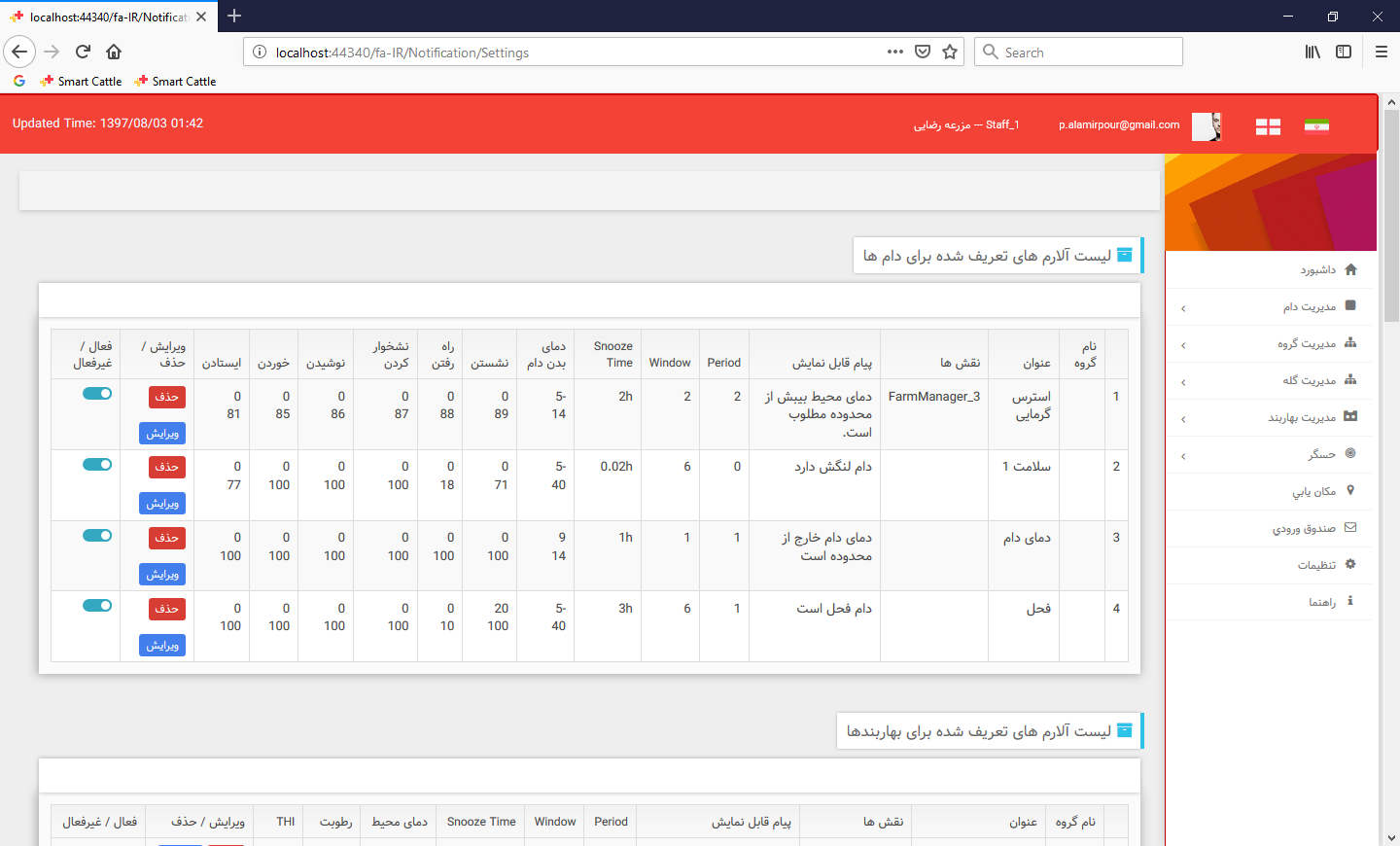
جدای از این که کاربر از نوع عادی یا بک آفیس وارد بخش انتخاب مزرعه شده اند، به هر حال لیست مزرعه ها قابل مشاهده هست. ولی این لیست برای کاربر بک آفیس شامل مزارعی می شود که دسترسی به آن به وی داده شده است. ولی برای کاربر عادی یه مقدار بحث فرق می کند. اگر کاربر عادی از نوع مدیر مزرعه باشد، در لیست مذکور، لیست مزرعه هایی را می بیند که صاحب آن محسوب می شود. اگر دامپزشک باشد، لیست مزرعه هایی را می بیند که در آن ها مشغول به فعالیت هست. حالت های خاص و مختلفی می توان مثال زد. به عنوان مثال، یک کاربر می تواند صاحب مزرعه باشد، همچنین دامپزشک هم باشد. آنگاه هر دو مزرعه را می تواند در این لیست مشاهده کند. با این تفاوت که اگر در مزرعه نخست لاگین کند، با نقش مدیر مزرعه و اگر در مزرعه دوم لاگین کند، با نقش دامپزشک که دسترسی های کمتری نسبت به مدیر مزرعه دارد وارد می شود.

کاربر بک آفیس بعد از کلیک بر روی دکمه SignIn وارد مزرعه مورد نظر می شود. همچنین وی می تواند مجدد با ورود به همین صفحه به نقش کاربر بک آفیس برگردد و از تمامی مزرعه ها خارج شود. ولی کاربر عادی فقط می تواند بین مزارعی که دسترسی به آن دارد سوییچ کند و نمی تواند از نقش خود خارج شود. برای بررسی بیشتر وارد مزرعه رضایی می شویم. این مزرعه قبلا توسط یک کاربر بک آفیس و در بخش تنظیمات ایجاد شده است. در آینده در مورد آن بیشتر صحبت خواهد شد. بعد از سوییچ به مزرعه، نرم افزار به صورت خودکار وارد داشبورد می شود و اطلاعات مربوط به آن مزرعه از دیتابیس فراخوانی می گردد. در ادامه به بررسی سایر گزینه های مربوط به صفحه تنظیمات پرداخته می شود.

1. ایجاد و ویرایش نقش ها
2. ایجاد اکانت جدید و نسبت دادن نقش به آن
3. ایجاد آلارم

در شکل 1- 10 صفحه ایجاد و ویرایش آلارم ها را مشاهده می کنید. کاربران مزارع در این صفحه می توانند از هر وضعیت بحرانی که ممکن است برای دام پیش بیایید مطلع شوند. این صفحه از چند بخش کلی تشکیل شده است. هر یک از این بخش ها جزییاتی دارند که به بررسی آن ها می پردازیم.

نخستین بخشی که در این صفحه می توان آن را مشاهده کرد، بخش لیست آلارم های تعریف شده برای دام ها است. در این بخش، هر تعداد آلارمی که کاربر عادی برای مزرعه تعریف کرده باشد قابل مشاهده، حذف/ویرایش و یا غیرفعال سازی است. همان طور که در شکل 1- 10 مشاهده می شود، لیست مذکور از تعدادی ستون تشکیل شده است.



شکل 1- 10، ایجاد و ویرایش آلارم

ستون نخست نام گروهی هست که آلارم در آن تعریف می شود. به عنوان مثال گروه سلامت، گروه فحلی و غیره. ستون بعدی عنوان آلارم هست. این عنوان در زمان ایجاد نوتیفیکیشن به کاربر نمایش داده می شود. ستون بعدی پیام قابل نمایش است. این پیام برای بیان جزییات بیشتر در رابطه با نوتیفیکیشن ایجاد شده است.

سه پارامتر بعدی شامل Period, Window و Snooze Time هست. این سه پارامتر در ایجاد نوتیفیکیشن بسیار موثر می باشد.

Window

به داده های مرتبط به دام نمی توان به صورت لحظه ای نگاه کرد. زیرا که تغییرات لحظه ای آن ها بسیار زیاد هستند. برای آن که بتوان تحلیل مناسبی از داده ها داشت لازم هست که آن ها را در بازه زمانی مشخصی بررسی کرد. به عنوان مثال می توان شش ساعت از داده ها رو جدا و از آن ها میانگین گرفت. پارامتر Window بدین منظور تعریف و استفاده می شود.

Period

از روی نام این پارامتر می توان کارکرد آن را حدس زد. به عنوان مثال اگر مقدار این پارامتر یک ساعت باشد، هر یک ساعت یک بار، سیستم به اندازه Window دقیقه/ساعت قبل را برای پیدا کردن مورد بحرانی بررسی می کند.

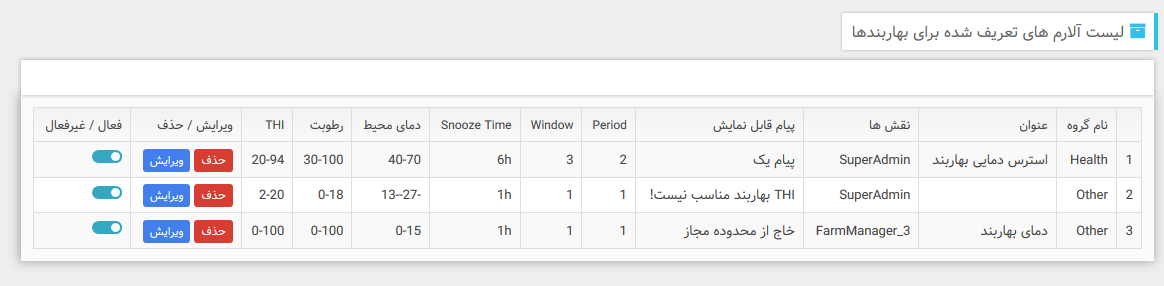
Snooze Time

از این پارامتر برای تعیین فاصله بین فعال و غیرفعال شدن نونیفیکیشن ها استفاده می شود. فرض کنیم یک نوتیفیکیشن ایجاد شده است و کاربری قصد غیر فعال سازی آن را دارد. این نوتیفیکیشن تنها می تواند برای Snooze Time دقیقه/ساعت غیر فعال باقی بماند و بعد از مدت مشخص اگر شرایط وجود آلارم صدق می کرد، نوتیفیکیشن به صورت خودکار مجدد فعال می شود. در بخش پیام های قابل نمایش، به صورت خودکار عبارتی مبنی بر این که نوتیفیکیشن مذکور به صورت خودکار توسط سیستم فعال شده است نمایش داده می شود. یکی دیگر از کاربردهای این پارامتر این است فرض کنید مشکل در دامی مشاهده شده است که برای ده ساعت این مشکل برقرار است. در این صورت هر بار که سیستم پارامترهای مربوط به مشکل را بررسی می کند، می تواند یک نوتیفیکیشن برای آن تولید کند. معنی آن این است که ممکن است هر ثانیه، هزاران نوتیفیکیشن به مدت ده ساعت تولید شود که عملا هم آزار دهنده هست و هم غیر مفید. به همین منظور از این پارامتر برای ایجاد وقفه بین بررسی آلارم مربوطه استفاده می شود.

ستون های بعدی لیست آلارم های تعریف شده برای دام ها، شامل دمای بدن دام، نشستن، راه رفتن، نشخوار کردن، نوشیدن، خوردن و ایستادن است. در این ستون ها محدوده مجاز و امنی که پارامترهای دام در آن می توانند قرار بگیرند تعریف شده است. اگر مقادیر دام از این محدوده تجاوز کنند، ممکن است منجر به تولید نوتیفیکیشن شود.

ستون بعدی مربوط به حذف و یا ویرایش تنظیمات مربوط به آلارم هاست. اگر آلارمی توسط کاربر حذف شود، دیگر سیستم نوتیفیکیشنی تولید نمی کند. ولی همچنان نوتیفیکیشن های از قبل ایجاد شده توسط آلارم مذکور قابل مشاهده خواهند بود. ولی اگر یک کاربر بخواهد به جای پاک کردن یک نوتیفیکیشن، موقتا آن را غیر فعال کند، می تواند بر روی دکمه اهرمی فعال/غیرفعال کلیک نماید. به این ترتیب تا فعال سازی مجدد آلارم توسط کاربر، سیستم نوتیفیکیشنی بر اساس این آلارم تولید نخواهد کرد.

جدول بعدی را آلارم های تعریف شده برای بهاربند تشکیل می دهد. جدول مذکور را در شکل 1- 11 می توانید مشاهده کنید.

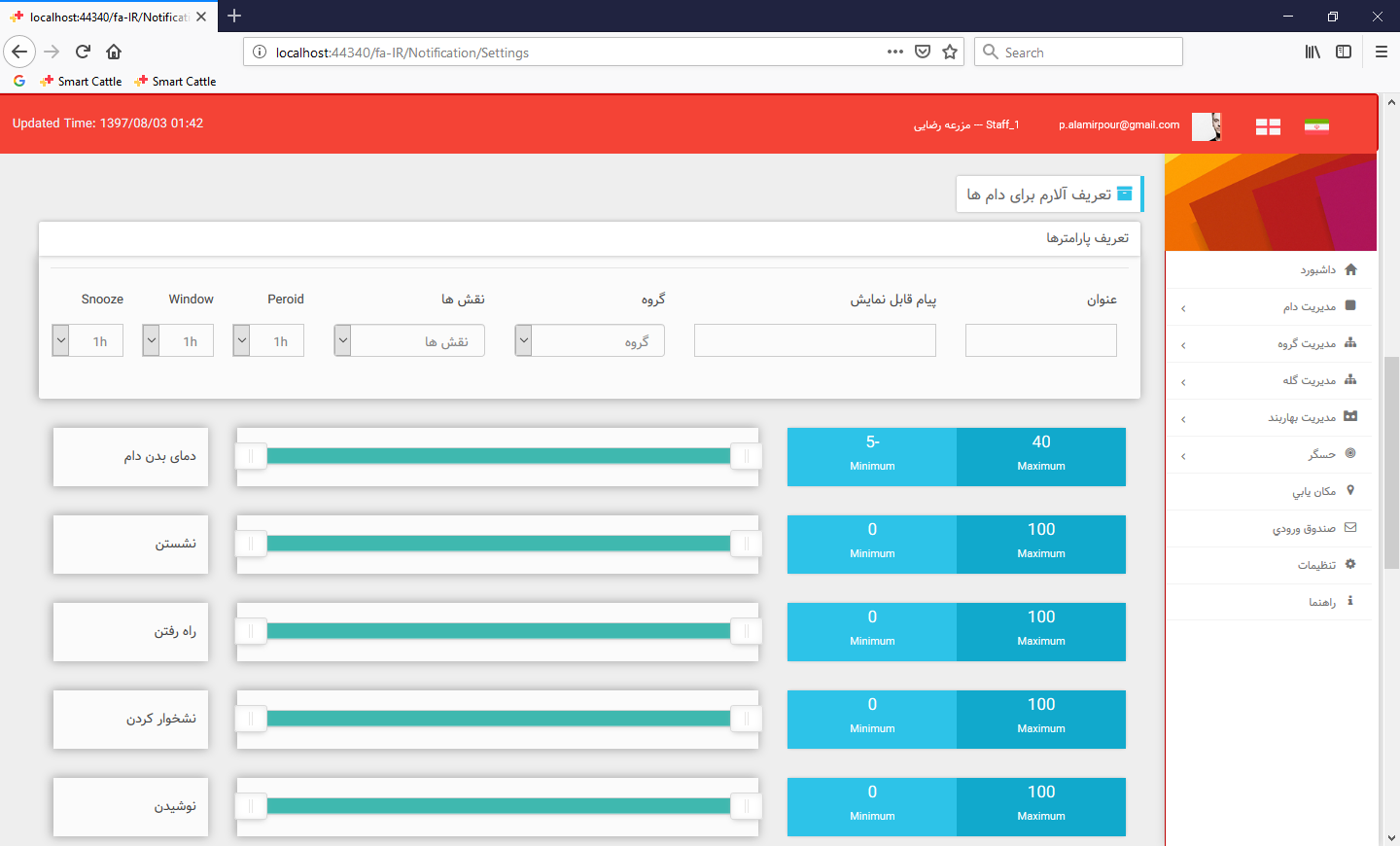


شکل 1- 11، لیست آلارم های تعریف شده برای بهاربندها

تمام مواردی که برای لیست آلارم های ایجاد شده برای دام ها بررسی کردیم برای جدول شکل 1- 11 هم صدق می کند به غیر از پارامترهای، دمای محیط، رطوبت و THI. این سه پارامتر شرایط محیطی بهاربند ها را توصیف می کنند. این پارامتر ها توسط سنسورهایی که در بهاربندها نصب شده اند دریافت می شوند و توسط تعریف آلارم قابل ردیابی هستند. بدین ترتیب کاربران مزارع می توانند در صورت عوض شدن شرایط محیطی بهاربندها، از آن مطلع شوند.

در شکل 1- 12، بخش تعریف آلارم برای دام ها مشاهده می شود. تقریبا تمام بخش های آن در قسمت های قبل توضیح داده شد. به غیر از سه نکته که لازم هست بیان شود. نکته نخست مربوط به بخش انتخاب نقش است. این بخش مشخص می کند که نوتیفیکیشن هایی که به واسطه این آلارم ایجاد می شوند برای کدام کاربر ارسال شوند. البته در نسخه های بعدی قرار است که این اطلاع رسانی به صورت سلسله مراتبی باشد.

نکته بعدی در مورد تعیین رنج فعال سازی آلارم هست. این رنج توسط دستگیره های سفید رنگ تعیین می شود. بخش سبز رنگ نمایش دهنده محدوده مجازی هست که پارامتر ها اگر در خارج از آن قرار داشته باشند منجر به تولید آلارم خواهد شد. ذکر این نکته ضروری است که تمام پارامتر ها با هم AND می شوند.



شکل 1- 12، تعریف آلارم برای دام ها

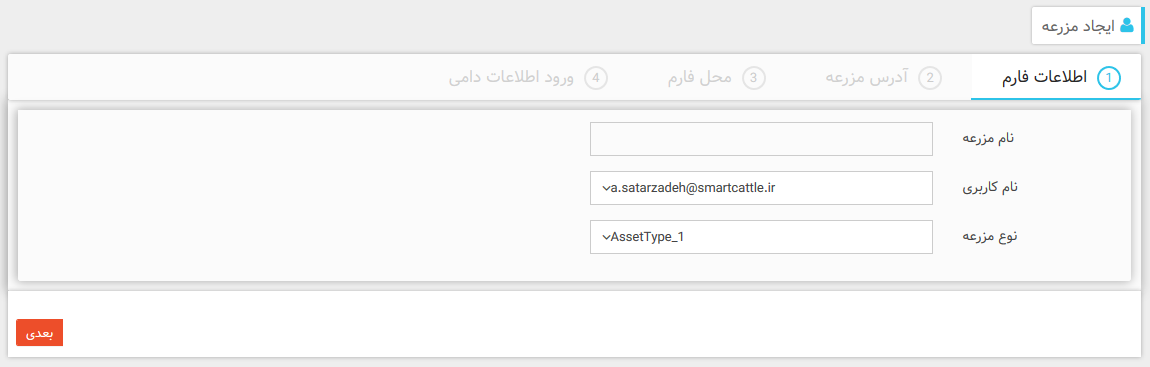
نکته آخر این است که اگر بخواهیم یک یا چند پارامتر در روند ایجاد نوتیفیکیشن بررسی نشوند، کافی است تا بر روی مقدار پیش فرض باقی بمانند. به این صورت به صورت خودکار از روند بررسی حذف خواهند شد.

1. ایجاد و ویرایش مزرعه

یکی از وظایف کاربر بک آفیس ایجاد و ویرایش مزرعه است. این تنها وظیفه ای هست که هیچ کاربر دیگه ای به آن دسترسی ندارد. پس اگر در این بخش اشاره به کاربر شد، منظور کاربر بک آفیسی است که اجازه ایجاد و ویرایش مزرعه را دارد.

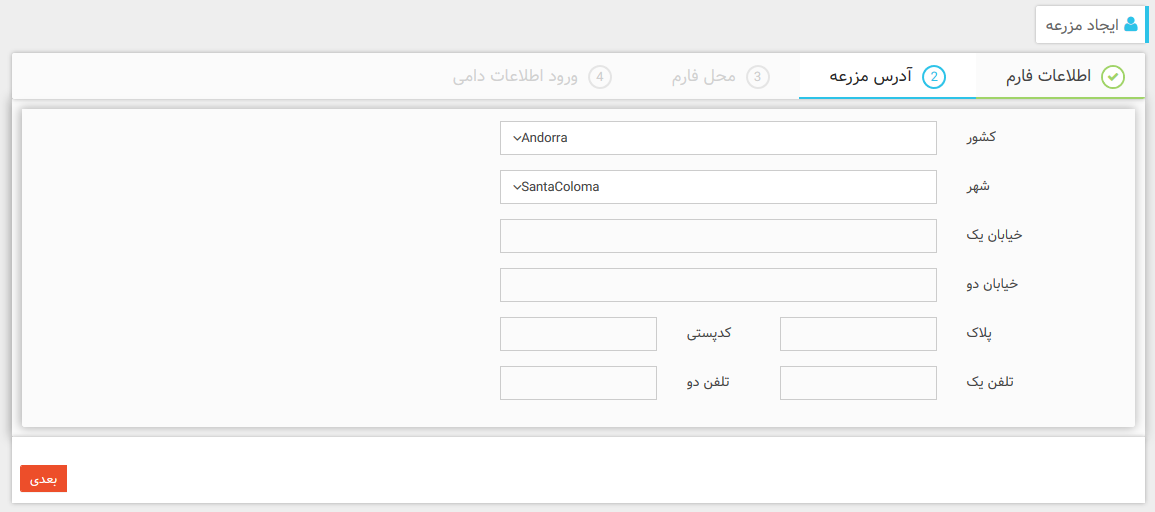
در شکل 1- 13، بخش ایجاد مزرعه مشاهده می شود. در اصل این بخش یک ویزارد چند مرحله است که در هر بخش از کاربر اطلاعاتی را برای ثبت مزرعه دریافت می کند. در اولین قدم کاربر می بایست نامی برای مزرعه انتخاب نماید. این نام می نواند غیر یکتا باشد. ولی باید به این نکته توجه کرد که بعدا اگر قرار باشد بین انبوهی از مزارع دنبال مزرعه مورد نظر بگردیم، نام مزرعه می تواند کمک بزرگی به ما کند.

هر مزرعه نیاز به یک نام کاربری دارد. اگر به یک نام کاربری همزمان دو مزرعه نسبت داده شود، کاربر مذکور می تواند به هر دو مزرعه با یک ایمیل و یک پسورد دسترسی داشته باشد.



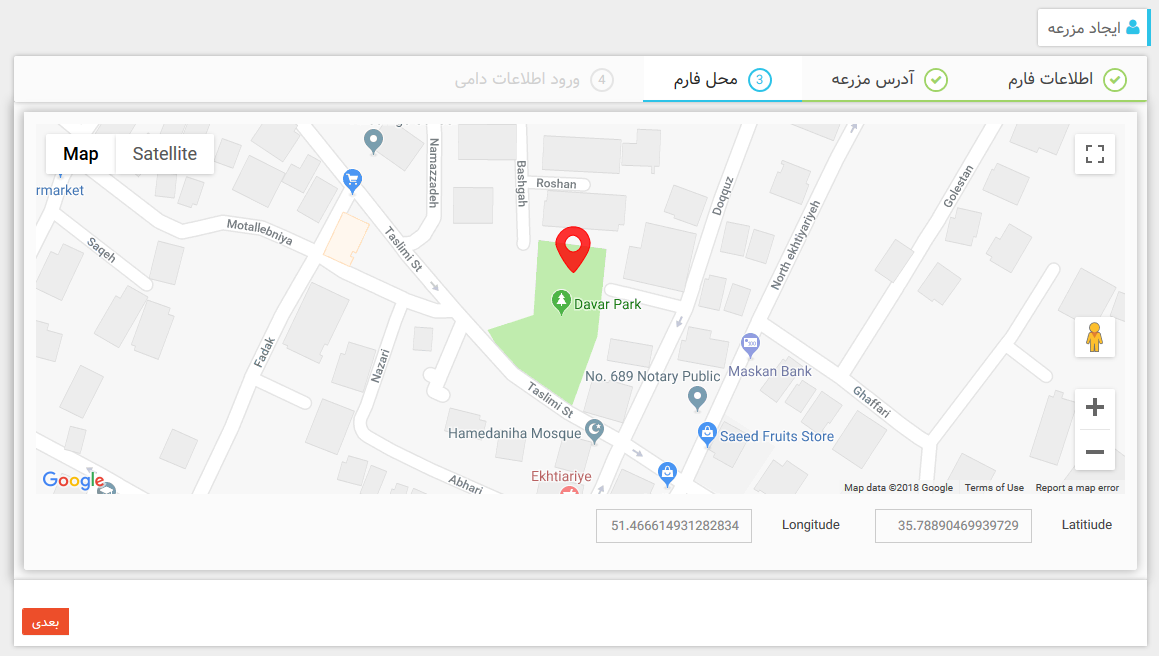
شکل 1- 13، بخش ایجاد مزرعه

در ادامه باید نوع مزرعه را مشخص کرد. منظور از نوع مزرعه، تجهیزاتی هست که یک مزرعه می تواند و یا مزرعه دیگر نمی تواند داشته باشد. در قدم بعدی می بایست اطلاعات مربوط به موقعیت مکانی مزرعه وارد شود. بدین منظور بر روی دکمه بعدی کلیک می کنیم.



شکل 1- 14، مرحله وارد کردن مشخصات مکانی مزرعه

در شکل 1- 14 فیلدهایی متناسب با اطلاعات مکانی مزرعه مشاهده می شود. این اطلاعات شامل کشور و شهر انتخابی برای مزرعه، آدرس و شماره تلفن و کدپستی است. توجه داشته باشید که انتخاب کشور و شهر، در نمایش وضعیت آب و هوایی در صفحه داشبورد تاثیر می گذارد.

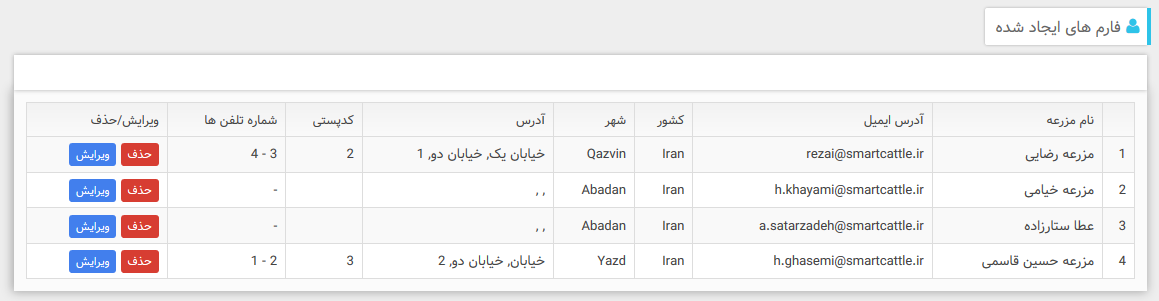


شکل 1- 15، انتخاب محل مزرعه بر روی نقشه

همانطور که در شکل 1- 15 مشاهده می شود، یک نقشه برای انتخاب محل فارم وجود دارد. لوکیشن انتخاب شده توسط کاربر در دیتابیس در کنار سایر داده های مربوط به مزرعه ذخیره می شود. ولی از آن در هیچ یک از بخش های اسمارت کتل استفاده نمی شود. البته این موقعیت در بخش ویرایش اطلاعات مزرعه قابل ویرایش است. احتمالا در آینده کاربردی برای آن در نظر گرفته شود.

در قدم بعدی و آخر، می توان لیست دام های متعلق به این مزرعه را در داخل نرم افزار بارگذاری کرد. همچنین می توان این مرحله را بدون وارد کردن دام ها رها کرد و بعدها از طریق منوی ثبت دام اقدام به این کار کرد. ولی به هر جهت اگر قصد وارد کردن لیست دام ها به مزرعه را دارید، کافی است تا فایل اکسل با فرمت مشخص را از طریق نرم افزار اسمارت کتل روی سرور آپلود کنید تا به صورت خودکار در لیست دام ها قرار بگیرد.

در زیر بخش ایجاد مزرعه، بخش دیگری قرار دارد که تصویر آن در شکل 1- 16 آمده است. ستون های جدول را خلاصه ای از اطلات مربوط به مزرعه تشکیل می دهد. با کلیک بر روی دکمه حذف می توان یک مزرعه را حذف کرد. همچنین با کلیک بر روی دکمه ویرایش می توان اطلاعات مربوط به مزرعه دلخواه را ویرایش نمود. لازم به ذکر است که کاربر بک آفیس تنها می تواند مزارعی را مشاهده نماید که یا دسترسی آن به وی اعتا شده یا این که خود آن مزرعه را ایجاد کرده باشد.

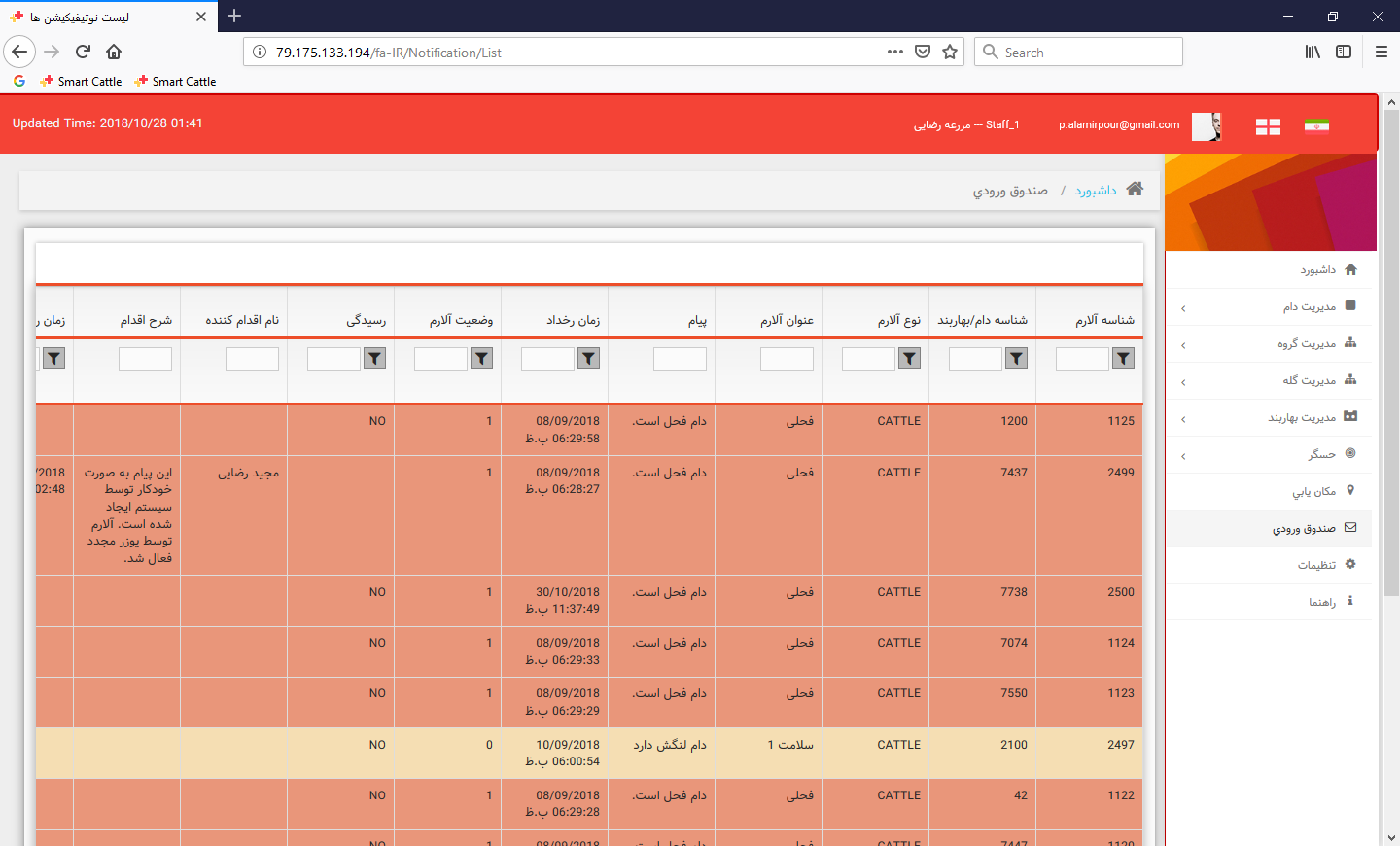


شکل 1- 16، مزرعه های ایجاد شده

منوی اصلی برنامه اسمارت کتل شامل موارد زیر است:

1. داشبورد
2. مدیریت دام
3. مدیریت گروه
4. مدیریت گله
5. مدیریت بهاربند
6. حسگر
7. مکان یابی
8. صندوق ورودی

همانطور که در بخش تنظیمات – ایجاد و ویرایش نوتیفیکیشن ها شرح داده شد، می توان نوتیفیکیشن های مختلف ایجاد شده در نرم افزار اسمارت کتل را در صندوق ورودی مشاهده کرد. در شکل 1- 13، صندوق ورودی را مشاهده می کنید. طبق قراری از این پس تنظیماتی که برای ایجاد نوتیفیکیشن انجام می شود را آلارم و اعلان های ایجاد شده به واسطه تنظیمات را نوتیفیکیشن می نامیم.



شکل 1- 17، صندوق ورودی

در این جدول سطرهایی که با رنگ قرمز مشخص شده اند، نوتیفیکیشن های جدیدی هستند که توسط نرم افزار تشخیص داده و ایجاد شده اند. در این جدول امکان جستجو و فیلتر گذاری وجود دارد.

ستون نخست، شناسه نوتیفیکیشن هست. یک عدد یکتا که به هر نوتیفیکیشن ایجاد شده نسبت داده می شود. تنها کاربرد این ستون، امکان ارجاع دادن کاربران به رخدادی هست که توسط نوتیفیکیشن مشخص شده است. ستون بعدی شناسه دام/بهاربند نام دارد. اگر نوتیفیکیشن ایجاد شده برای دام باشد، شماره ای که در زیر این ستون درج می شود، مربوط به شماره بدنه دام است. اگر نوتیفیکیشن ایجاد شده برای بهاربند باشد، مقادیر این ستون، شماره زون بهاربندی است که نوتیفیکیشن در آن رخداده است. در ستون بعدی نوع آلارم مشخص می شود. همان طور که در بخش تنظیمات – ایجاد و ویرایش آلارم بیان شد، دو نوع آلارم دام و بهاربند وجود دارد. عبارت CATTLE در زیر این ستون نشان دهنده آن است که نوتفیکیشن ایجاد شده از نوع دام است.

ستون بعدی مربوط به عنوان نوتیفیکیشن و پیام است. در هنگام ایجاد آلارم جدید توسط کاربر، وی می بایست برای بهتر مشخص شدن دلیل بروز نوتیفیکیشن، عنوان و پیامی را به دلخواه درج نماید. چنین اطلاعات درج شده ای هنگام بروز نوتیفیکیشن به کاربر نمایش داده می شود.

ستون بعدی مربوط به زمان رخداد نوتیفیکیشن هست. ستونی با نام وضعیت نوتیفیکیشن، وجود دارد که فعال یا غیر فعال بودن یک نوتیفیکیشن را نمایش می دهد. سطرهایی که به رنگ زرد کمررنگ هستند، نوتیفیکیشن های غیر فعال هستند. ستون بعدی رسیدگی نام دارد. اگر مقدار درج شده در این ستون NO باشد، معنی آن این است که آلارم ایجاد شده است ولی کسی به آن توجه نکرده است. همچنین اگر مقدار آن YES باشد، یعنی کاربری اقدامی برای برطرف کردن مشکل گزارش شده توسط آلارم انجام داده است و سپس گزارشی مبنی بر ACK شدن آلارم ثبت کرده است. در رابطه با انجام دادن اقدام های مختلف برای نوتفیکیشن ها به زودی توضیحاتی را مشاهده خواهید کرد. در ستون بعدی نام اقدام کننده جهت حل مشکل به صورت خودکار درج می شود. در حقیقت نام اقدام کننده، نام همان کاربری است که هم اکنون در نرم افزار لاگین کرده است. اگر وی توضیحاتی برای اقدام خود درج کرده باشد، در زیر ستون شرح اقدام می توان آن را مطالعه کرد. اگر این اقدام مربوط به غیرفعال/فعال کردن یک نوتفیکیشن باشد، پیامی به صورت خودکار توسط سیستم تولید و ذخیره می شود. همچنین زمان رسیدگی در ستون بعدی قابل پیگیری می باشد.

ستون آخر مربوط به زمان غیر فعال شدن آلارم است. اگر یک آلارم توسط کاربری غیرفعال شود، تاریخ آن در زیر این ستون درج می شود.

ممکن است برای یک دام یا بهاربند مشخص، بیش از یک نوتیفیکیشن مرتبط با آلارم تولید شده باشد. تمام آن ها در یک دسته بندی مشخصی قابل مشاهده هستند. برای دستیابی به آن، کافی است تا بر روی یک نوتیفیکیشن کلیک شود تا پنجره کوچکی باز گردد. این پنجره از تعدادی دکمه تشکیل شده است که گاهی کاربری دکمه ها با هم متفاوت خواهند شد. تمام دکمه هایی که ممکن است در این پنجره با آن رو به رو شوید به شرح ذیل است:

1. تاریخچه

کاربر با کلیک بر روی این دکمه می تواند تمامی نوتیفیکیشن های تولید شده برای یک دام مشخص و به ازای یک آلارم مشخص در یک جدول مجزا مشاهده نماید.

1. رسیدگی

اگر کاربر اقدام مشخصی برای برطرف شدن آلارم مشخصی انجام داده باشد، می تواند با کلیک بر رو این دکمه، در پنجره جدید آن اقدام را درج نماید. در کتابچه اسمارت کتل، هرجا اشاره به عبارت Ack کردن نوتیفیکیشن شد، منظور همین اقدامی هست که شرح داده شد. با Ack کردن یک نوتیفیکیشن در حقیقت پرونده آن بسته می شود. در این صورت بلافاصله بعد از آن، اگر همچنان سیستم گزارشی مبنی بر حیات مشکل دریافت کند، به مدت Snooze صبر می کند و نوتیفیکیشنی تولید نمی کند. ولی اگر بعد از گذشت مدت مشخص، نوتیفیکیشن جدیدی تولید شود، این موضوع در سطر جدیدی به عنوان موضوع جدید در نظر گرفته می شود ولو این که به واسطه به وجود آمدن یک آلارم و برای یک دام/بهاربند خاص باشد. نوتیفیکیشن هایی که با موفقیت رسیدگی شده اند به رنگ سبز نمایش داده خواهند شد.

1. غیرفعال

با کلیک بر روی این دکمه، ایجاد نوتیفیکیشن برای موضوع و دام/بهاربند خاصی موقتا غیرفعال می شود. مدت زمانی که ایجاد نوتیفیکیشن برای موضوع خاصی می تواند غیر فعال باشد، توسط پارامتری به اسم Snooz مشخص می شود. این پارامتر در زمان ایجاد آلارم مقدار دهی می شود. ذکر این نکته بدیهی است که اگر ایجاد نوتیفیکیشن برای یک موضوع خاص و دام/بهاربند خاص غیر فعال شود، برای همان موضوع و سایر دام/بهاربندها همچنان فعال است و در صورت لزوم ممکن است منجر به تولید نوتیفیکیشن هم شود.

1. فعال سازی مجدد

اگر ایجاد نوتیفیکیشن برای موضوع خاصی توسط کاربر غیر فعال شده باشد، دکمه فعال سازی مجدد به جای دکمه غیرفعال سازی ظاهر می شود.

1. بستن

با کلیک بر روی این دکمه تنها پنجره جاری بسته می شود و تاثیری بر عملکرد آلارم ها ندارد.

در انتها به عنوان جمع بندی می توان به این مساله اشاره کرد که، ممکن است برای یک دام چند نوتیفیکیشن مختلف به ازای آلارم های مختلف ایجاد شود. تمام این آلارم ها در سطور مختلف در شکل 1- 10 از هم تفکیک شده اند. ولی اگر نوتیفیکیشن های ایجاد شده برای یک دام به واسطه یک آلارم باشد، تمامی آن ها را در تاریخچه نوتیفیکیشن می توان مشاهده کرد و تنها آخرین وضعیت، جهت اطلاع، یک سطر از جدول شکل 1- 10 را به خود اختصاص خواهد داد.

فصل سه

معماری اسمارت کتل

مقدمه

در این فصل در رابطه با معماری بخش های مختلف اسمارت کتل صحبت خواهد شد. نرم افزار اسمارت کتل برای تغذیه داده های خود، نیاز دارد تا به سرور اسمارت کتل وصل شود. شیوه برقراری ارتباط با سرور اسمارت کتل در فصول بعدی شرح داده خواهد شد. داده ها پس از دریافت از سرور اسمارت کتل به کاربر نمایش داده می شوند. شکل زیر در مورد معماری دریافت داده اسمارت کتل توضیحاتی را بیان می کند.

Farm sensor Data

SmartCattle Listener

Sql server

IPARS

MongoDB

RabbitMQ

SmartCattle Software

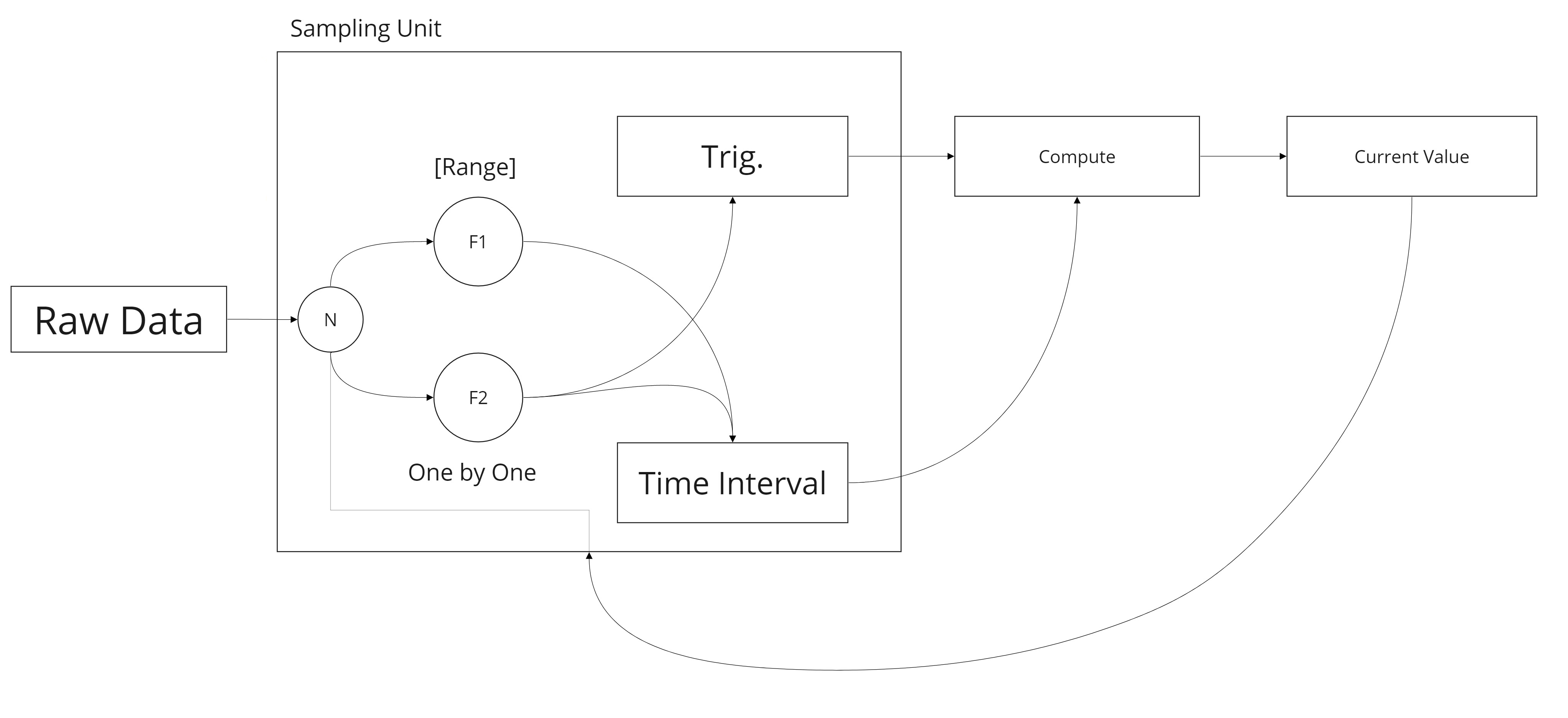
SmartCattle Services

همانطور که در تصویر بالا مشخص شده است، داده ها در سرور آی پارس در یک صفی قرار می گیرند. برای هر فارم یا به عبارتی برای هر مشتری مستقل، صف جدایی تعریف می شود و داده های مربوط به سنسورهای محیطی، دمای بدن دام و فعالیت های دام در داخل آن ریخته می شود. Listener اسمارت کتل به محض آن که داده ای در صف قرار گیرد متوجه آن می شود و اطلاعات را خوانده و در دیتابیس ذخیره می کند. در ادامه هر زمان که کاربر وارد نرم افزار اسمارت کتل شد، داده های مورد نیازش را با درخواست شناسه ها و پارامتر های مشخصی دریافت می کند. در مورد پارامترهای مورد نیاز برای فراخوانی داده ها از فصل مربوط به توضیح اکشن های اسمارت کتل به صورت مفصل توضیح داده خواهد شد. همچنین در فصل مربوط به توضیح API ها، پارامترهای مورد نیاز برای فراخوانی سرویس های اسمارت کتل به صورت مفصل بررسی خواهند شد.

----Break----

معماری نونیفیکیشن

در شکل زیر ساختار کلی نوتیفیکیشن نمایش داده شده است. در این ساختار داده ها در ابتدا وارد واحد نمونه برداری می شود. در این واحد دو فیلتر f1 و f2 وجود دارد که یکی روی یک بازه و دومی روی یک فیلد می باشد.



دو حالت برای انجام محاسبات روی داده های ورودی وجود دارد. در حالت اول Trig، زمانی رخ می دهد که یک تغییر در یک فیلد مشخص از داده اتفاق بیفتند. این تغییر می تواند یکی از دو عمل Update, Insert باشد. به عنوان مثال وقتی داده جدیدی برای دمای بدن دام ثبت می شود یک تریگ رخ می دهد و داده مربوطه به واحد محاسبات ارسال می شود.

در حالت دوم، داده های ورودی در یک Time Interval از پیش تعیین شده ای جمع آوری می شوند و به واحد محاسبات ارسال می شوند.

با توجه به این که واحد محاسبات به عنوان یک ماژول مستقل طراحی و پیاده سازی شده، می توان انواع محاسبات را برای بررسی رفتارهای مختلف سیستم در آن تعریف کرد. به عنوان مثال می توان هر ده دقیقه یک بار داده های دمای بدن دام یک بهاربند را جمع کرد و از آن ها میانگین گرفت. سپس نتیجه در CurrentValue با یک Dynamic Label ذخیره می شود. یعنی اگر از همین نوع دیتا قبلا در جدول ذخیره شده باشد، مقدار جدیدی Insert نمی شه و مقدار قبلی Update می شه.

بر روی جدول CurrentValue یک تریگر تعریف شده که هر بار مقدار جدیدی در آن قرار می گیرد یا سطری آپدیت می شود، آن تریگر فعال می شود و مقادیر درج شده را با تنظیماتی که در جدول نوتیفیکیشن درج شده است مقایسه می کند و در ادامه نوتیفیکیشن ایجاد می شود. این همان بخشی هست که در SmartCattleCoreProcessor پیاده سازه شده است. در ادامه به توضیح دقیق تر این روند پرداخته خواهد شد.

پارامترهایی که برای ایجاد نوتیفیکیشن توسط سیستم بررسی می شوند به دو دسته کلی قابل تقسیم هستند. پارامترهایی که شرایط محیطی بهاربندها را توصیف می کنند. نظیر دما، رطوبت و THI. پارامترهایی که شرایط سلامت دام و یا سایر موارد مهم مربوط به آن را توصیف می کنند. مثل راه رفتن، نشستن، نشخوار کردن و غیره. تمامی مراحلی که برای ایجاد یک نوتیفیکیشن رخ می دهد، بین تمامی پارامترها یکسان هستند. لذا برای نمونه روند ایجاد نوتیفیکیشن برای فعالیت راه رفتن در ادامه توضیح داده خواهد شد. به طبع توضیحات داده شده قابل بسط به سایر پارامترهای توصیفی است. با در نظر گرفتن تغییرات پارامتر دمای بدن دام آغاز می کنیم.

در حال حاضر تغییرات دمای بدن دام هر دو دقیقه یک بار از طریق ربیت دریافت و در دیتابیس اسمارت کتل ذخیره می شود. هر بار که داده جدیدی ذخیره می شود، منجر به فراخوانی روند ذیل می گردد.

به جهت آن که پارامتر مورد بررسی دمای بدن دام است، لذا تمام آلارم های تنظیم شده برای دام از دیتابیس فراخوانی می شود. این آلارم ها برای تفکیک شدن از یکدیگر یک شناسه یکتا دارند. از این شناسه برای ساختن پارامتری استفاده می کنیم که آن را TagName می نامیم. دلیل و شیوه ساخت این پارامتر توضیح داده خواهد شد.

TagName = "#Tag\_Cattle\_Avg\_Temp\_" + \_CattleNotificationsSetting.ID + "\_" + CattleId;

عبارت Tag\_Cattle اشاره به این دارد که این تگ برای یک دام ساخته می شود. عبارت Avg\_Temp مشخص می کند که برای میانگین گیری از دمای بدن دام از این تگ استفاده خواهد شد.

عبارت CattleNotificationSetting.ID همان شناسه یکتایی هست که صحبت آن شد. در انتها شناسه یکتایی که برای مشخص کردن دام ها از یکدیگر استفاده می شود به TagName اضافه خواهد شد. پس به طور کلی می خواهیم یک TagName یکتا درست کنیم که میانگین دمای بدن یک دام مشخص را بتوان با آن مشخص کرد. این TagName قرار است در جدول CurrentValue نگه داری شود. این جدول تعدادی سطر و ستون دارد. یکی از ستون های آن همین TagName هست که آن را به صورت یکتا ساختیم. ستون بعدی مقداری هست که TagName به آن اشاره می کند و آن را Value می نامیم که در مثال ما می شود میانگین دمای بدن دام. ستون بعدی در جدول مورد بحث، اشاره به آخرین تاریخی دارد که تغییراتی در پارامتری رخ داده است. حالا وقت آن رسیده که تمام این دانش رو در کنار هم قرار بدیم و نتیجه رو ببینیم.

پس هر بار که داده جدیدی برای دمای بدن دام ثبت شد، به ازای تمام تنظیماتی که برای آلارم مرتبط با دام ثبت شده، TagName ویژه ای ساخته می شه. این TagName ویژه در جدول CurrentValue جستجو می شود. اگر پیدا نشد، ثبت می شود در غیر این صورت ممکن هست که آپدیت شود. قبل از این که به مساله آپدیت شدن بپردازیم در مورد مقادیری که قرار است ذخیره یا آپدیت شود صحبت کنیم. یکی از این مقادیر مربوط به آخرین تاریخی هست که در جدول مذکور اتفاقی رخ داده است. پس اگر درست همین الان TagName در جدول ذخیره یا آپدیت شود، ستون مربوط به تاریخ، مقدار همین لحظه را به خود می گیرد. اما مقدار ستون بعدی یا به عبارتی Value به چه صورت محاسبه می شود؟

یکی از پارامترهایی که در زمان ثبت یک آلارم کاربر مجبور به تنظیم آن است، پارامتر Window می باشد. به اندازه Window دقیقه/ساعت از دیتای دمای بدن دام توسط سیستم برداشته و سپس از آن ها میانگین گرفته می شود. مقدار حاصل در ستون Value از جدول CurrentValue قرار می گیرد.

گفته شد که اگر TagName ساخته شده قبلا در جدولCurrentValue وجود نداشت، به عنوان مقدار جدید ثبت می شود در غیر این صورت یا ممکن هست آپدیت شود یا خیر. یکی دیگر از پارامترهایی که کاربر برای ثبت یک آلارم مجبور به تعریف آن بود پارامتر Period هست. سیستم قبل از آپدیت کردن مقداری، آخرین تاریخ ثبت شده برای آن TagName را استخراج می کند. اگر فاصله بین زمان جاری با زمان ثبت شده مذکور بیش از مقدار Period باشد، مقدار جدید جایگزین مقدار قبلی Value می شود. در غیر این صورت عمل آپدیت اتفاق نمی افتد.

به این ترتیب می تونیم میانگین دمای بدن دام رو در یک بازه زمانی Window دقیقه/ساعت هر چند Period دقیقه/ساعت داشته باشیم بدون این که نیاز باشه در جدول بزرگی مثل CattleBodyTempreture با میلیون ها رکورد جستجو و محاسبه کنیم. در حقیقت با استفاده از عبارت یونیک TagName می توانیم به این داده دسترسی داشته باشیم. همانطور که در از سطور بالا برداشت شد، داده های محاسبه شده گذشته دور ریخته می شوند و همیشه آخرین محاسبات در جدول ذخیره شده است. سوالی که مطرح می شود این هست که اگر دام در گذشته دچار مشکلی شده باشد، با توجه به دور ریختن داده ها، نوتیفیکیشنی برای آن تولید نمی شود؟ پاسخ این سوال رو در ادامه بررسی خواهیم کرد.

همانطور که گفته شد اگر داده ای برای دمای بدن دام ثبت شود، بلافاصله سناریوی بالا روی آن اجرا می شود. به طور مشابه، هر بار که داده جدیدی در جدول CurrentValue ثبت می شود، سناریوی زیر به صورت خودکار اتفاق می افتد. قبل از آن که به شرح سناریو پرداخته شود لازم است تا سوالی که در بالا پرسیده شد پاسخ داده شود. با توجه به این که این سناریوها به صورت زنجیره وار به هم وصل هستند، نباید نگران نادیده گرفتن داده ای توسط سیستم بود. زیرا که اگر کار بخشی تمام شود، بخش بعدی به صورت خودکار وظایف خود را آغاز می کند. بخشی که قصد شرح آن را داریم به طور مستقیم در تولید نوتیفکیشن نقش دارد.

شرح سناریو را از این بخش آغاز کنیم که هر بار که داده جدیدی در جدول CurrentValue قرار می گیرد یا آپدیت می شود، برای بررسی به واحد تولید نوتیفیکیشن ارسال می شود. در این واحد ابتدا از روی TagName اطلاعاتی نظیر، شماره منحصر به فرد دام، نوع داده و شناسه آلارم که طبق آن مقدار میانگین محاسبه شده است استخراج می شود. به عنوان مثال TagName جدیدی در جدول CurrentValue آپدیت شده است که مربوط به دمای بدن دام هست. با توجه به اطلاعات استخراج شده، TagName های دیگری به صورت خودکار ساخته می شود. این TagName ها برای پارامتر راه رفتن، نشستن، نشخوار کردن و غیره ساخته می شود. پس از آن، تمامی این تگ ها در جدول CurrentValue مورد جستجو قرار می گیرند تا سایر پارامترهای مربوط به آن دام هم گردآوری شود. زیرا برای بررسی آن که پارامترهای جاری یک دام در محدوده غیرمجاز یک آلارم تعریف شده صدق می کند یا نه، به همه پارامترهای دام نیاز داریم. بعد از گردآوری همه مقادیر مورد نیاز برای تحلیل دام، تنظیماتی که با استفاده از آن این مقادیر در Window و Period مشخص محاسبه شده فراخوانی می شود. اگر پارامترهای مذکور برابر با مقدار پیش فرض خودشان باشند، در روند بررسی ایجاد نوتیفیکیشن نادیده گرفته می شوند. اگر پارامترهای دام در تنظیمات مربوط به آلارم صدق کند وارد مرحله بعدی خواهد شد در غیر این صورت نادیده گرفته می شود.

در مرحله بعدی، سیستم چک می کند که آیا قبلا برای این دام و با توجه به چنین پارامترهایی نوتیفیکیشنی ایجاد شده است یا خیر. برای این کار از یک TagName مخصوص استفاده می کند. این تگ همانند تگ جدول CurrentValue ساخته می شود ولی به جای عبارت ابتدایی آن، عبارتی قرار می گیرد که مشخص شود این نوتی برای دام می خواهد ایجاد شود یا برای بهاربند. پس ابتدا وجود نوتیفیکیشن برای تگ مذکور چک می شود. توضیح سناریو رو به این صورت پیش می بریم که تصور می کنیم برای تگ مذکور هنوز نوتیفیکیشنی ثبت نشده باشد. در این صورت در جدول Notification یک نوتی جدید با وضعیت ACTIVE،رسیدگی NO، تگ مربوطه و مقدار Snooze Time از پیش تعیین شده ذخیره می شود.

حالا برای پارامترهای مذکور و با توجه به تنظمیات تعیین شده، برای دام خاصی نوتیفیکیشنی ثبت شد. مجدد روند توضیح داده شده در سطور پیش تکرار می شود. اگر مجدد برای همین دام با همین تنظیمات آلارم بخواهد نوتیفیکیشنی ایجاد شود، ببینیم چه اتفاقی خواهد افتاد.

ابتدا یک تگ برای نوتیفیکیشن ساخته می شود. با توجه به این که برای این تگ قبلا نوتی ایجاد شده است، اولین داده ای که از آن استخراج می شود، زمان Snooze Time هست. اگر زمان ثبت شده برای تگ نوتی، بعلاوه زمان Snooze از زمان جاری کوچکتر باشد، روند ایجاد نوتی ادامه پیدا می کند در غیر این صورت متوقف می شود.

بیایید فرض کنیم روند تولید نوتی با توجه به Snooze Time ادامه پیدا می کند. در این صورت سه شرط در ابتدا بررسی می شود. مهمترین شرطی که اول از همه بررسی می شود این است که آیا قبلا این نوتیفیکیشن Act شده است یا خیر. اگر Act نشده باشد وضعیت آن چک می شود که آیا DEACTIVE هست یا خیر. یعنی آیا نوتی به صورت خودکار توسط سیستم غیرفعال شده است یا خیر. اگر این مورد هم صدق نکرد آخرین شرطی که چک می شود این است که آیا این نوتی توسط کاربر غیرفعال شده است یا خیر. حالا بیایید هر مرحله رو به صورت مجزا با هم چک کنیم.

نوتی از قبل Act شده باشد

وقتی کاربری یک نوتی را Act می کند، معنی اش آن است که از دید وی پرونده این مشکل بسته شده است. ولی اگر باز هم مشکل مشابه توسط سیستم تشخیص داده شده باشد، دست کم معنی آن این است که اقدامات صورت گرفته توسط کاربر منجر به حل کامل مشکل نشده است. پس باید یک نوتی جدید ایجاد شود تا کاربر از وجود مشکل با خبر شود. این موضوع به صورت سطر جداگانه ای در صندوق ورودی قابل مشاهده خواهد بود.

وضعیت نوتی DEACTIVE باشد

سیستم جدای از این که می تواند تشخیص دهد آیا شرایط ایجاد نوتی وجود دارد یا خیر، می تواند تشخیص دهد که این شرایط برطرف شده است یا خیر. در مورد روند بررسی این موضوع در آینده نزدیک صحبت خواهد شد. ولی اگر سیستم قبلا تشخیص داده باشد که مشکل مدتی هست که از بین رفته، ولی مجدد فعال شده است، تنها کاری که می کند این است که وضعیت نوتی از قبل ایجاد شده را به ACTIVE آپدیت می کند.

نوتی توسط یوزر غیرفعال شده باشد

ممکن است یوزر ترجیح دهد موقتا یک نوتی را غیرفعال کند. اگر مشکل همچنان پابرجا باشد، و کاربر هم نوتی را غیرفعال کرده باشد و Snooze Time گذشته باشد، نوتی مجدد به صورت خودکار فعال می شود.

تا این جا روند ایجاد نوتی بررسی شد. یک روند به صورت مستقل در کنار سناریوی بالا در جریان هست که در ادامه شرح داده خواهد شد. این روند زمانی اتفاق می افتد که شرایط دام یا بهاربند از نظر سیستم در حال حاضر مشکوک نیست ولی قبلا برای آن نوتی ثبت شده است.

نخستین پارامتری که اول از همه بررسی می شود مقدار Snooze Time هست. اگر آخرین داده ای که برای نوتی ایجاد شده ثبت شده از میزان Snooze Time بیشتر باشد و کاربر برای نوتی Act ثبت نکرده باشد و همچنین اگر کاربر نوتی را غیرفعال نکرده باشد، وضعیت نوتی به صورت خودکار به حالت غیرفعال سوییچ می شود.

در انتها باید به این نکته اشاره کرد که تمام این موارد در تاریخچه نوتی ها قابل پیگیری و ردیابی است.

مروری بر ساختار نقش ها و اکانت ها

این بخش با نسخه ویرایش شده تطبیق داده شود. همچنین برای آن معماری و فلوچارت رسم شود.

تست ایجاد مزرعه، اکانت و فارم

ایجاد اکانت

* ایجاد اکانت از نوع کارمند فارم
* بدون فارم

این بخش برای کاربران فارم مسدود شده است و فقط کاربر بک آفیس به آن دسترسی دارد. کاربر ایجاد شده از این راه، نمی تواند به نرم افزار اسمارت کتل لاگین شود. در اصل لاگین شدن وی مسدود شده است. زیرا هیچ فارمی برای وی هنوز ثبت نشده است. رل و پرمیژن های او نیز NULL هستند.

* انتخاب یک فارم

این بخش برای کاربران فارم مسدود شده است و فقط کاربر بک آفیس به آن دسترسی دارد. کاربری که از این طریق ایجاد می شود فقط به فارمی دسترسی دارد که توسط دراپ دان لیست انتخاب می شود. بعد از ایجاد فقط می تواند وارد همان فارم شود و رل وی توسط کاربر ایجاد کننده از طریق دراپ دان لیستی انتخاب شده است. لیست رل ها تنها می توانند شامل رل هایی باشند که قبلا برای آن فارم تعریف شده باشند. یعنی رل ها بین فارم ها مشترک نیست و مختص هر فارم هستند. لیست فارم هایی که کاربر می تواند برای آن اکانتی ایجاد کند بر اساس میزان دسترسی کاربر ایجاد کننده اکانت محدود شده است.

وقتی کارمند بک آفیس وارد بخش ایجاد اکانت می شود، علاوه بر امکان ایجاد اکانت، می تواند لیست اکانت هایی که برای فارم تا الان ایجاد شده است رو مشاهده کنه. از بین اکانت هایی که تا الان ایجاد شده است، تنها می تواند آن هایی را مشاهده کند که برای فارم هایی هستند که کاربر جاری اجازه دسترسی به آن ها را دارد. سایر اکانت ها از دید پنهان می ماند. همچنین کاربر بک آفیس جاری می تواند، اکانت های از نوع بک آفیس رو در کنار اکانت خودش مشاهده کنه. ولی امکان ویرایش و یا حذف اکانت خودش برای کاربر مسدود شده است. اکانت های مذکور زیرمجموعه وی محسوب می شوند ولی همتا نیستند.

* ایجاد اکانت از نوع کارمند بک آفیس
* این نوع اکانت برای کاربران فارم مسدود شده است. کاربر بک آفیس اگر اجازه ایجاد اکانت داشته باشه می تونه یک اکانت کارمند بک آفیس ایجاد کند. به این ترتیب که لیست رل مورد نظرش رو از دراپ دان انتخاب می کند. البته از بین تمام رل ها، فقط آن هایی را می تواند انتخاب کند که زیر مجموعه خودش محسوب می شود. سپس از بین لیست فارم هایی که تنها به آن دسترسی دارد، می تواند هر کدام را که بخواهد به اکانت جدید نسبت دهد. کاربر ایجاد شده بعد از این که لاگین می شود، باید در بخش انتخاب فارم، فارمی که می خواهد کارهای پشتیبانی آن را انجام دهد انتخاب کند در غیر این صورت به یک سری اکشن ها دسترسی دارد که الزاما نیاز به بودن در یک فارم خاص ندارد.

نکته ای که وجود داره اینه که، وقتی یک کارمند بک آفیس وارد نرم افزار می شه، می تونه لیست اکانت های زیر مجموعه خودش رو ببینه. ولی همتاهای خودش رو نمتونه ببینه. معنیش اینه که اکانت خودش رو نمی تونه توی لیست ببینه چون از دید نرم افزار خود کاربر، همتای خودش محسوب می شه. البته اگر کاری هم کنیم که بخواهد ببینه، به جهت این که اجازه حذف و ویرایش خودشون رو به کاربر جاری نمی دیم، فقط دیدن یا ندیدن اکانت خود، یک حالت بصری دارد از این که من به عنوان کاربر جاری در سیستم تعریف شده هستم چون خودم رو می بینم یا تعریف شده نیستم چون خودم رو نمی بینم ولی تونستم وارد بشم. به هر حال این موضوع فقط در مورد کارمند بک آفیس صدق می کنه. ولی در مورد کارمند فارم قضیه فرق می کنه.

* ایجاد اکانت توسط کارمند فارم

هر اکانتی که در این مرحله ایجاد شود متعلق به همان فارمی هست که کاربر در آن قرار دارد. همچنین نقش هایی که کاربر جاری می تواند برای اکانت جدید از دراپ دان لیست انتخاب کند، همتا یا زیر مجموعه کاربر محسوب می شود. در زیر بخش ایجاد فارم، کاربر تنها می تواند لیست اکانت های از قبل ایجاد شده ای را ببیند که یا همتای وی هستند یا زیر مجموعه او محسوب می شوند. رده بالاتر یا کارمند بک آفیس در این صفحه از دید وی پنهان می شوند. در لیست اکانت های از پیش تعریف شده، اکانت کاربر جاری هم وجود دارد. ولی امکان ویرایش و یا حذف برای وی مسدود شده است.

ورود به فارم

* ورود کارمند بک آفیس

وقتی کارمند بک آفیس وارد نرم افزار می شود، ابتدا به ساکن، مشخصات هیچ فارمی را نمی تواند مشاهده کند. یعنی اگر در این شرایط وارد لیست دام ها شود، هیچ دامی را نمی تواند ببیند. در این شرایط می تواند یک سری اکشن هایی که وظیفه بک آفیس هست انجام دهد. در مورد این اکشن ها در سایر بخش ها به صورت مفصل توضیح داده شده یا در سطور آتی توضیح داده خواهد شد. برای انجام کارهای پشتیبانی مربوط به یک فارم، باید حتما از قسمت سوییچ، یکی از فارم هایی که اجازه دسترسی به آن را دارد انتخاب نماید. در بخش سوییچ فارم، در مورد این بخش به تفضیل توضیح داده خواهد شد. پس در کل کارمند بک آفیس بعد از ورود به نرم افزار با دو سناریو رو به رو هست. کارهایی که مستقل از فارم می توان انجام داد و کارهایی که انجام آن وابسته به فارم است.

* ورود کارمند فارم

وقتی یک کارمند فارم وارد نرم افزار می شود، می تواند از تمامی بخش های نرم افزار با حداکثر میزان دسترسی که برای اکانت وی تعیین شده استفاده نماید. اگر کارمند فارم، دو نقش مختلف در دو فارم مختلف داشته باشد، هنگام ورود به فارم، یکی از فارم ها به انتخاب سیستم برای وی نمایش داده می شود. برای رفتن به فارم دیگر لازم نیست تا از نرم افزار خارج شود. کافی است در بخش سوییچ فارم، یکی از فارم هایی که قصد ورود به آن را دارد انتخاب نماید تا از آن پس در آن فارم قرار گیرد. برای رفتن به فارم های مختلف نیازی به وارد کردن دو ایمیل جدا با دو پسورد جدا نیست. هر دو فارم کاربر مذکور تنها با یک ایمیل و با یک پسورد قابل دسترس است.

اگر کارمند فارم، فارمی نداشته باشد و یا هنوز به وی مزرعه ای انتساب نشده باشد، وی برای ورود به نرم افزار با پیغام خطا مواجه می شود.

ایجاد فارم

* کارمند بک آفیس

وقتی کارمند بک آفیس وارد فارم می شود، اگر اجازه دسترسی به بخش ایجاد فارم داشته باشد می تواند این گزینه را ببیند و وارد آن بشود در غیر این صورت نمی تواند به این بخش دسترسی داشته باشد. یعنی اول چک می شود که آیا کارمند بک آفیس هست یا نه، اگر مثبت بود، سطح دسترسی وی بررسی می شود. اگر اجازه دسترسی داشت، آن وقت می تواند این بخش را ببیند و در آن اقدام به ایجاد فارم نماید.

وقتی کاربر مذکور وارد بخش ایجاد فارم می شود، از لیست فارم های ایجاد شده تنها می تواند آن هایی را ببیند که به آن اجازه دسترسی دارد. پس به این ترتیب می تواند مشخصات آن فارم ها را ببیند و یا فارمی را حذف و یا ویرایش نمیاد. برای ویرایش یک فارم توسط کاربر بک آفیس، باید در لیست فارم های ایجاد شده بر روی دکمه ویرایش کلیک نمیاد. وقتی وارد صفحه ویرایش می شود، می تواند عمده اطلاعات هویتی و نوعی فارم را ویرایش نماید. به عنوان مثال وی می تواند نوع یک فارم را از AssetType1 به AssetType2 تغییر دهد. ولی این اجازه تغییر نوع تایپ زمانی برای وی مقدور هست که، سطح دسترسی های داده شده به آن کارمند در سطح دسترسی های AssetType2 باشد. برای بررسی این موضوع، سیستم لیست سطح دسترسی های کارمند بک آفیس را از دیتابیس می خواند، همچنین لیست دسترسی های تعریف شده برای AssetType2 را نیز می خواند و سپس آن ها را با هم مقایسه می کند. برای این که کارمند بک آفیس بتواند AssetType2 را انتخاب کند یا باید تمام دسترسی های تعریف شده در AssetType2 را بدون کم و کاست داشته باشد یا علاوه بر آن بیش از دسترسی های لازم AssetType2 در اختیار داشته باشد. به زبان ریاضی بزرگتر مساوی با AssetType2 باشد.

همین موضوع در مورد ایجاد فارم نیز صدق می کند. کاربر بک آفیس فقط می تواند فارم هایی را درست کند که، AssetType های آن فارم کوچکتر مساوی باشند با اختیارات آن کارمند بک آفیس.

در بخش ایجاد مزرعه، نامی برای مزرعه توسط کارمند بک آفیس درج می شود، سپس اکانت و نوع مزرعه انتخاب می گردد. اکانت هایی که کارمند بک آفیس آن را انتخاب می کند دو حالت کلی دارد. یا تازه ایجاد شده اند و مزرعه ای به آن ها نسبت داده نشده. یا این که اکانت مورد نظر مزرعه دارد و قرار است مزرعه دیگری هم به لیست مزارع اون اضافه گردد. همچنین اکانت مذکور می تواند دو نوع کلی داشته باشد. یا اکانت از نوع کارمند مزرعه ویا اکانت از نوع کارمند بک آفیس. در ادامه این چهار حالت بررسی می شوند.

1. اکانت از نوع کارمند بک آفیس

در نرم افزار اسمارت کتل نمی توان برای اکانت های از نوع بک آفیس، فارمی تعریف کرد. این بدان معناست که کاربر بک آفیس، هنگام ایجاد فارم، نمی تواند ایمیل فردی را که کارمند بک آفیس است از دراپ دان لیست اکانت ها انتخاب نماید و فارمی را به وی منتسب نماید.

1. اکانت از نوع کارمند فارم

در حال حاضر هیچ محدودیت برای انتساب یک فارم به اکانت های از نوع کارمند فارم وجود ندارد.

1. اکانت فارمی ندارد و تازه ایجاد شده است

اگر کاربری فارمی نداشته باشد، نمی تواند وارد نرم افزار شود. پس انتساب فارم، از آن پس کاربر همانند یک کارمند فارم با نقشی که برای وی تعریف شده است به فارم دسترسی دارد.

1. قبلا فارمی به اکانت نسبت داده شده است

اگر قبلا فارمی برای اکانت مذکور ثبت شده باشد، فارم جدید به لیست فارم های وی اضافه می شود. به این ترتیب فارم منیجر آن فارم می تواند در بخش سوییچ فارم، فارم جدید خود را ببیند.

* کارمند فارم

وقتی کارمند فارم وارد بخش تنظیمات می شود، گزینه ایجاد فارم را نمی بیند. مهم نیست که چه میزان از سطح دسترسی داشته باشد، این گزینه ابدا برای وی نمایش داده نمی شود.

سوییچ بین فارم ها

وقتی کاربری وارد بخش سوییچ بین فارم های می شه، یک لیستی از فارم هایی که به آن دسترسی دارد را مشاهده می کند که در مقابل آن یک دکمه برای ورودی به آن فارم وجود دارد. رفتار این دکمه برای دو نوع کاربر مختلف یعنی کارمند بک آفیس و کارمند فارم متفاوت هست. وقتی کارمند یک فارم به نرم افزار لاگین می کند، حتما در یکی از فارم هایی که به آن دسترسی دارد قرار دارد. یعنی امکان ندارد که کارمند فارمی وارد نرم افزار اسمارت کتل شود و داخل فارمی نباشد. نقش وی هم متناسب با فارمی هست که در آن قرار دارد. در لیست فارم ها در صفحه سوییچ بین فارم ها، دکمه ورود به فارم جاری با رنگ متفاوت مشخص شده است و روی آن درج شده Signed In. کاربر وقتی روی آن کلیک می کند اتفاق خاصی نمی افتد. ولی با کلیک بر روی سایر دکمه ها می تواند وارد فارم دلخواه خود شود. ولی وقتی کارمند بک آفیس وارد می شود قضیه قدری متفاوت هست. حتما و حتما وقتی کارمند بک آفیس وارد اسمارت کتل می شود، داخل هیچ فارمی قرار ندارد. پس در صفحه سوییچ بین فارم ها، تمام وضعیت دکمه های ورود به فارم در حالت عادی قرار دارد. وقتی کارمند بک آفیس روی دکمه ورودی کلید می کند و وارد فارمی می شود، وضعیت آن دکمه به Sign Out تغییر پیدا می کند. یعنی کارمند بک آفیس در مراجعه بعدی به صفحه سوییچ بین فارم ها، دو کار می تواند انجام دهد. یا روی یک فارم دیگر کلیک کند و وارد آن شود و یا این که روی دکمه Sign Out کلیک کند و مستقل از هر فارمی شود.

* کارمند بک آفیس

وقتی یک کارمند بک آفیس وارد بخش سوییچ بین فارم های می شه، تنها می تونه لیست فارم هایی رو ببینه که اجازه دسترسی به آن ها رو داره. وقتی کاربر مذکور روی دکمه SignIn کلیک می کنه، از این به بعد وارد اون فارم می شه ولی ورودش چند نکته داره. نکته اول اینه که میزان دسترسی اون کاربر به فارم، همان میزان دسترسی هست که کاربر مذکور از قبل داشته. چون دلیل ورود یک کارمند بک آفیس به فارم انجام کارهای پشتیبانی هست، اگر بخواهد دسترسی محدود داشته باشه ممکن هست نتونه مشکل فارم رو پس از ورود حل کنه. ولی دسترسی های همین کاربر هم بعد از ورود به فارم جدید به بعضی بخش ها محدود هست. محدودیت های مذکور به شرح ذیل هست(به یاد داشته باشید که داریم دسترسی های یک کاربربک آفیس رو بعد از وارد شدن به یک فارم بررسی می کنیم):

* وقتی کاربر بک آفیس وارد بخش ایجاد نقش ها می شود، می تواند همانند یک کارمند فارم نقش ایجاد کند و نمی تواند مانند یک کارمند بک آفیس از دراپ دان لیست، نوع نقش ویا مزرعه ای را انتخاب کند. لیست پرمیژن هایی که به وی برای انتساب به نقش نمایش داده می شود برابر است با حداکثر میزان دسترسی که رییس اون فارم داره. پس ملاحظه ای که این جا صورت گرفته به این صورت هست که، کاربر بک آفیس در این حالت، در لیست پرمیژن ها، تمام پرمیژن هایی که یک کارمند بک آفیس همتای خودش دارد را ندارد. همچنین محدودیت هایی که یک کارمند فارم ممکن هست داشته باشد را ندارد. یعنی کارمند فارم ممکنه بعضی پرمیژن ها رو ببینه و بعضی ها رو نبینه. در نهایت تمام چیزی که وی از پرمیژن ها می بینه، همانی است که فارم منیجر می بینه. این طوری یک کارمند بک آفیس هم نمی تونه باعث بشه که نقشی بالاتر از تجهیزاتی که به یک فارم فروخته شده اعطا کنه و همچنین اینقدر هم نقشش محدود نشده که نتونه کار موثری در رابطه با پشتیبانی انجام بده.

در ادامه کاربر بک آفیس، از لیست نقش های از قبل تعریف شده، فقط می تواند نقش هایی را ببیند که برای آن فارمی هست که در حال حاضر لاگین شده. نکته ای که وجود دارد این هست که از بین نقش هایی که برای وی نمایش داده می شود، تنها می تواند نقش هایی را ببیند که زیر مجموعه وی باشند. با توجه به این که نقش های کارمندان بک آفیس به نحوی تعیین می شوند که اشراف کامل به یک یا چند فارم داشته باشند، پس تمام نقش های تعریف شده در آن فارم زیر مجموعه وی محسوب می شوند و در نتیجه تمام نقش ها را می تواند ببیند. نکته جالبی که وجود دارد این هست که اگر فارم منیجر وارد این صفحه بشود، نقش خود را می بیند ولی نمی تواند خود را ویرایش کند. ولی کارمند بک آفیس وقتی وارد این صفحه می شود، علاوه بر این که نقش فارم منیجر را می تواند ببیند، وی را می تواند ویرایش هم نماید.

* کارمند فارم

وقتی کارمند فارم وارد بخش سوییچ می شه، به تعداد فارم هایی که در آن عضو هست، در جدول فارم ها، گزینه SignIn می بیند. ستون های این جدول رو مشخصات فارم تشکیل داده. کاربر جاری در یکی از آن ها به عنوان مثال نقش FarmManager رو داره و در دیگری نقش یک دامپزشک. روی هر کدام که کلید کند، می تواند وارد فارم مذکور شود. بر خلاف کارمند بک آفیس که پس از ورودش به فارم همچنان نقشش ثابت می ماند، کارمند فارم بعد از ورود به یک فارم، همان نقشی رو به خودش می گیره که در آن فارم برای وی تعریف شده است. مثلا اگر در فارم الف به عنوان منیجر تعریف شده باشد، بعد از سوییچ به آن، نقش منیجر رو می گیره. اگر در فارم ب به عنوان دامپزشک تعریف شده باشد، بعد از سوییچ با فارم ب، نقش دامپزشک به اون تعلق می گیرد. ممکن است در اسمارت کتل برای فارم های مختلف چندین نقش دامپزشک تعریف شده باشد، ولی در زمان یک کاربر به یک فارم، دقیقا همان نقشی را می گیرد که در فارم مقصد برای وی تعریف شده است. جالب است بدانید تمام این ها تنها با یک ایمیل و پسورد مدیریت می شود. یعنی لازم نیست تا کاربر ایمیل و پسورد دیگری برای ورود به فارم دیگری تعریف یا وارد کند.

ایجاد رل

* ایجاد نقش توسط کارمند بک آفیس

وقتی کارمند بک آفیس وارد بخش ایجاد نقش می شه، از سه نوع نقش فقط می تواند دو نوع آن را ایجاد کند. نقش از نوع کارمند فارم، نقش از نوع جنس فارم. نقش سومی که کارمند بک آفیس نمی تواند آن را ایجاد کند، نقش از نوع کارمند فارم هست. دلیل این که نمی تواند نقشی برای فارم خاصی ایجاد کند این هست که، سیستم را محدود به اول، انتخاب فارم، سپس ایجاد نقش کردیم. یعنی اگر کارمند بک آفیس می خواهد مثلا برای فارم رضایی نقشی ایجاد کند، اول باید وارد فارم رضایی شود سپس اقدام به ایجاد نقش برای آن فارم کند. پس می توان به طور کلی گفت کارمند بک آفیس سه نوع نقش می تواند ایجاد کند که دوتای آن را تنها زمانی که به عنوان کارمند بک آفیس وارد شده می تواند ایجاد کند و سومی را زمانی که به عنوان کارمند فارم وارد فارم خاصی شده. در انتها وی می تواند دو نوع لیست در کادر زیر ایجاد نقش ببینید. اولی مربوط به نقش های ایجاد شده از نوع فارم هست و دومی مربوط به نقش های ایجاد شده برای کارمندان بک آفیس هست. نقش های ایجاد شده برای کارمندان فارم را وقتی با اکانت بک آفیس وارد اسمارت کتل شده نمی تواند ببیند. اگر نقشی را بخواهد مشاهده کند می تواند وارد فارم تحت دسترسی اش بشود و نقش مورد نظرش را ببیند، حذف و یا ویرایش نماید. کارمند بک آفیس از لیستی که در این صفحه می بینید، تنها می تواند نقش هایی را ببیند که زیر مجموعه وی محسوب می شوند. کارمند بک آفیس خودش رو در لیست نقش های تعریف شده برای کارمندان می تونه ببینه ولی نمی تونه ویرایش کنه.

* ایجاد نقش توسط کارمند فارم

وقتی کارمند فارم وارد بخش ایجاد نقش می شود، می تواند لیست نقش هایی که در فارم از قبل تعریف شده رو مشاهده کند با این شرط که اون نقش ها، زیرمجموعه نقش کاربر جاری باشند. همچنین کاربر فعلی می تواند نقش خود را در لیست نقش ها مشاهده کند و ببینید به کدام بخش ها دسترسی دارد. ولی نمی تواند نقش خود را حذف یا ویرایش نماید. هر نقشی که کاربر فعلی ایجاد کند، مختص همان فارم هست و جای دیگر قابل استفاده نیست. همچنین این نقش را تنها می توان به یک اکانت نسبت داد و به فارم قابل انتساب نیست. همچنین این کاربر هیچ دراپ دان لیستی برای انتخاب گزینه ای ندارد. وقتی کاربر جاری می خواهد نقشی تعریف کند، لیست نقش هایی که برای وی نمایش داده می شود تا انتخاب کند، همان هایی هست که خودش به آن ها دسترسی دارد.

سناریوهای مختلف که باید روی آن فکر شود.

* وقتی یک کاربر بک آفیس یک اکانت ایجاد می کند، بلافاصله بعد از ایجاد اکانت، نمی تواند آن را در لیست اکانت های ایجاد شده ببیند. زیرا لیست اکانت های ایجاد شده بر اساس اجازه دسترسی کارمند بک آفیس چیده شده است. مثلا کاربر بک آفیس به فارم های شماره یک و دو دسترسی دارد. پس می تواند تمام اکانت های ایجاد شده این دو فارم را ببیند. ولی وقتی اکانت جدیدی توسط وی ایجاد می شود، متعلق به هیچ فارمی نیست پس برای وی نمایش داده نمی شود.
* ایجاد کردن مزرعه تنها بر عهده کارمند بک آفیس هست و سایر کاربران نمی توانند این اکشن را انجام دهند. برای ایجاد مزرعه باید نوع مزرعه توسط کارمند بک آفیس انتخاب شود. انواع مزارعی که به کارمند بک آفیس نمایش داده می شود تنها بستگی به نقش وی دارد. از این رو، برخی به وی نمایش داده می شود و برخی دیگر نه. ولی این موضوع در مورد اکانت ها رعایت نمی شود. کارمند بک آفیس بدون هیچ گونه محدودیتی می تواند به لیست تمام اکانت ها دسترسی داشته باشد و برای هر کدام به دلخواه مزرعه ای ایجاد نماید.

فصل چهار

شرح API ها

مقدمه

در این فصل در مورد API های پیاده سازی شده به طور مفصل شرح داده خواهد شد. ساختار توضیح داده شده به این صورت است که ابتدا آدرس API به همراه پروتکل ارتباطی آن بیان می شود. به عنوان مثال: POST: <http://www.domainname.com> یعنی که که پروتکل ارتباطی مورد نیاز برای برقراری ارتباط با این API پست می باشد که بر روی آدرس مذکور فعال هست.

در انتهای URL سرویس شرح داده شده ابتدا SubDirectory های سرویس و بعد از آن پارامترهایی که باید برای آن ارسال شود درج می شود. پارامترها برای آن که از SubDirectory ها مجزا شوند، در داخل آکولاد قرار داده شدند. پس از بیان ساختار سرویس ها، عملکرد آن و پارامترهای مورد استفاده شرح داده می شوند. توضیحات مربوط به پیاده سازی سرویس ها در فصل 4 بیان شده اند.

|  |
| --- |
| POST: http://79.175.133.194/SmartCattle/APIs/EnvSensors/List/{FarmId}/{FreeStallid} |
| از این API برای دریافت لیست سنسورهای یک بهاربند خاص استفاده می شود. |
| FarmId  شناسه فارم جاری |
| FreeStalId  شناسه بهاربندی که می خواهیم سنسورهای محیطی آن را استخراج کنیم. |

|  |
| --- |
| POST: http://localhost:5763/EnvSensors/THIValue |
| دریافت داده های مربوط به نمودار THI و رطوبت و دما با استفاده از وب سرویس میسر می شود. خروجی این سرویس سه پارامتر فیزیکی شامل دما، رطوبت و THI را بر می گرداند. |
| {  "FarmId": 2,  "FreeStallId": 32,  "StartTime": "2017-12-08 11:40",  "EndTime":"2017-12-08 11:41",  "MACs":  [  {  "Address":"d1:13:a3:c5:d1:be",  "Step": 2  },  {  "Address":"e3:ad:eb:31:e0:ef",  "Step": 20  }  ]    } |
| FarmId  شناسه فارم |
| FreeStalId  شناسه بهاربند |
| StartTime  زمان آغاز داده های جمع آوری شده |
| EndTime  زمان پایان داده های جمع آوری شده |
| MACs  پارامتر فوق یک لیست از مک آدرس ها است. بدین معنی که می توان به عنوان مثال دو مک را در یک ریکوئست ارسال کرد و به ازای هر یک داده ها را در یک Response به صورت جداگانه و json دریافت کرد. |
| Address  شماره مکی که قصد دریافت داده از آن را داریم. |
| Step  توالی زمانی داده ها را مشخص می کند. |

|  |
| --- |
| POST: http://79.175.133.194/APIs/CattleSensors/Activity |
| دریافت داده های مربوط به اکتیویتی دام |
| {  "FarmId": 3,  "CattleId": 2033,  "StartTime": "2017-12-20 09:07",  "EndTime": "2017-12-25 21:07",  "Step": 2  } |
| FarmId  شناسه فارم |
| CattleId  شناسه دامی که می خواهیم اکتیویتی آن را استخراج کنیم. |
| StartTime  زمان آغاز داده های جمع آوری شده |
| EndTime  زمان پایان داده های جمع آوری شده |
| Step  توالی زمانی داده ها را مشخص می کند. |

|  |
| --- |
| POST: http://79.175.133.194/APIs/EnvSensors/List |
| با استفاده از این API می شه متوجه شد که برای یک بهاربند چندتا سنسور محیطی نصب شده است. |
| {  "FarmId" : 3,  "FreeStallId": 5  } |
| FarmId  شناسه فارم |
| FreeStalId  شناسه بهاربند |

فصل پنج

ساختار دیتابیس

مقدمه

در این فصل در مورد ساختار دیتابیس توضیح داده خواهد شد. شیوه توضیح به این صورت است که ابتدا نام تمامی جداولی که در پیاده سازی این پروژه مورد استفاده قرار گرفته بیان می شود و در مورد هر یک به صورت مختصر توضیحی درج می شود. سپس در ادامه فیلدهای هر یک از جداول به همراه کوئری که برای ساخت آن نیاز است بیان می گردد.

لیست جدول های دیتابیس

در جدول زیر لیست جدول های مورد استفاده برای پروژه اسمارت کتل را مشاهده می کنید.

|  |  |  |
| --- | --- | --- |
| ID | Name | Comment |
| 1 | EnvSensors | لیست سنسورهای محیطی برای اندازه گیری دما و رطوبت |
| 2 | ActivityStateTbl | فعالیت دام ها نظیر نشستن، راه رفتن، نشخوار کردن و غیره |
| 3 | CattleScoreTbl | ثبت امتیاز های هر دام |
| 4 | TempretureTbl | دمای بدن دام ها |
| 5 | SensorTbl | سنسورهای وصل شده به دام برای تشخیص فعالیت و دمای بدن |
| 6 | CattlePositionTbl | موقعیت مکانی دام ها |
| 7 | CattleTbl | لیست دام هایی که برای یک دام ثبت می شوند. |
| 8 | CattleGroupTbl | گروه بندی دام ها |
| 9 | EnvTHITbl | داده های سنسور محیطی شامل دما و رطوبت و میزان THI در این جدول ذخیره می شوند. |
| 10 | FarmTbl | لیست فارم هایی که درBackOffice ساخته شده اند. این لیست به Email هایی تخصیص داده می شوند. همچنین شناسه فارم ها برای لینک با سایر جدول ها استفاده می شود. |
| 11 | FreeStallTbl | لیست بهاربندها |
| 12 | UserInfo | داده های مربوط به کاربرا در این بخش ذخیره می شود. |
| 13 | UserPermissions | دسترسی های تعریف شده در برنامه |
| 14 | UserRoles | تعریف نقش ها |
| 15 | Notifications | نوتیفیکیشن هایی که تولید می شوند در این جا ذخیره می شوند |
| 16 | FreeStallNotificationsSetting | تنظیمات مربوط به نوتیفیکیشن بهاربندها |
| 17 | CattleNotificationsSetting | تنظیمات مربوط به نوتیفیکیشن دام ها |

در ادامه تمامی کدهای مورد نیاز برای تولید جداول درج شده اند. همانطور که می دانید برای ساختن این جداول نیاز هست تا تک تک فیلدها در SQL تعریف شوند. ولی از این پس فقط با دانستن کدهای زیر و اجرای کردن هر کدام می توان، تمامی جدول ها را خیلی سریع ساخت.

جدول EnvSensors

|  |
| --- |
| USE [smartCattle]  GO  SET ANSI\_NULLS ON  GO  SET QUOTED\_IDENTIFIER ON  GO  CREATE TABLE [SmartCattle].[EnvSensors](  [id] [int] IDENTITY(1,1) NOT NULL,  [FreeStallId] [int] NULL,  [FarmId] [int] NULL,  [Lat] [float] NULL,  [Lng] [float] NULL,  [MAC] [nvarchar](50) NULL  ) ON [PRIMARY]  GO |

جدول ActivityStateTbl

|  |
| --- |
| USE [smartCattle]  GO  SET ANSI\_NULLS ON  GO  SET QUOTED\_IDENTIFIER ON  GO  CREATE TABLE [SmartCattle].[ActivityStateTbl](  [ID] [int] IDENTITY(1,1) NOT NULL,  [jsonedActivities] [nvarchar](max) NULL,  [Sitting] [decimal](18, 2) NOT NULL,  [Standing] [decimal](18, 2) NOT NULL,  [Walking] [decimal](18, 2) NOT NULL,  [Eating] [decimal](18, 2) NOT NULL,  [Rumination] [decimal](18, 2) NOT NULL,  [Drinking] [decimal](18, 2) NOT NULL,  [cattleId] [int] NOT NULL,  [date] [datetime2](7) NOT NULL,  [FarmID] [int] NOT NULL,  [LastRecievedId] [bigint] NOT NULL,  [UserId] [nvarchar](max) NULL,  CONSTRAINT [PK\_SmartCattle.ActivityStateTbl] PRIMARY KEY CLUSTERED  (  [ID] ASC  )WITH (PAD\_INDEX = OFF, STATISTICS\_NORECOMPUTE = OFF, IGNORE\_DUP\_KEY = OFF, ALLOW\_ROW\_LOCKS = ON, ALLOW\_PAGE\_LOCKS = ON) ON [PRIMARY]  ) ON [PRIMARY] TEXTIMAGE\_ON [PRIMARY]  GO  ALTER TABLE [SmartCattle].[ActivityStateTbl] WITH CHECK ADD CONSTRAINT [FK\_SmartCattle.ActivityStateTbl\_SmartCattle.CattleTbl\_cattleId] FOREIGN KEY([cattleId])  REFERENCES [SmartCattle].[CattleTbl] ([ID])  ON DELETE CASCADE  GO  ALTER TABLE [SmartCattle].[ActivityStateTbl] CHECK CONSTRAINT [FK\_SmartCattle.ActivityStateTbl\_SmartCattle.CattleTbl\_cattleId]  GO  ALTER TABLE [SmartCattle].[ActivityStateTbl] WITH CHECK ADD CONSTRAINT [FK\_SmartCattle.ActivityStateTbl\_SmartCattle.FarmTbl\_FarmID] FOREIGN KEY([FarmID])  REFERENCES [SmartCattle].[FarmTbl] ([ID])  GO  ALTER TABLE [SmartCattle].[ActivityStateTbl] CHECK CONSTRAINT [FK\_SmartCattle.ActivityStateTbl\_SmartCattle.FarmTbl\_FarmID]  GO |

جدول CattleScoreTbl

|  |
| --- |
| USE [smartCattle]  GO  SET ANSI\_NULLS ON  GO  SET QUOTED\_IDENTIFIER ON  GO  CREATE TABLE [SmartCattle].[CattleScoreTbl](  [ID] [int] IDENTITY(1,1) NOT NULL,  [item] [int] NOT NULL,  [value] [float] NOT NULL,  [CattleId] [int] NOT NULL,  [Date] [datetime2](7) NOT NULL,  [UserId] [nvarchar](max) NULL,  CONSTRAINT [PK\_SmartCattle.CattleScoreTbl] PRIMARY KEY CLUSTERED  (  [ID] ASC  )WITH (PAD\_INDEX = OFF, STATISTICS\_NORECOMPUTE = OFF, IGNORE\_DUP\_KEY = OFF, ALLOW\_ROW\_LOCKS = ON, ALLOW\_PAGE\_LOCKS = ON) ON [PRIMARY]  ) ON [PRIMARY] TEXTIMAGE\_ON [PRIMARY]  GO  ALTER TABLE [SmartCattle].[CattleScoreTbl] WITH CHECK ADD CONSTRAINT [FK\_SmartCattle.CattleScoreTbl\_SmartCattle.CattleTbl\_CattleId] FOREIGN KEY([CattleId])  REFERENCES [SmartCattle].[CattleTbl] ([ID])  ON DELETE CASCADE  GO  ALTER TABLE [SmartCattle].[CattleScoreTbl] CHECK CONSTRAINT [FK\_SmartCattle.CattleScoreTbl\_SmartCattle.CattleTbl\_CattleId]  GO |

جدول TempretureTbl

|  |
| --- |
| USE [smartCattle]  GO  SET ANSI\_NULLS ON  GO  SET QUOTED\_IDENTIFIER ON  GO  CREATE TABLE [SmartCattle].[TempretureTbl](  [ID] [int] IDENTITY(1,1) NOT NULL,  [value] [float] NOT NULL,  [point] [nvarchar](max) NULL,  [cattleId] [int] NOT NULL,  [date] [datetime2](7) NOT NULL,  [LastRecievedId] [bigint] NOT NULL,  [FarmID] [int] NOT NULL,  [UserId] [nvarchar](128) NULL,  [FreeStall] [int] NULL,  [dateStr] [nvarchar](max) NULL,  CONSTRAINT [PK\_SmartCattle.TempretureTbl] PRIMARY KEY CLUSTERED  (  [ID] ASC  )WITH (PAD\_INDEX = OFF, STATISTICS\_NORECOMPUTE = OFF, IGNORE\_DUP\_KEY = OFF, ALLOW\_ROW\_LOCKS = ON, ALLOW\_PAGE\_LOCKS = ON) ON [PRIMARY]  ) ON [PRIMARY] TEXTIMAGE\_ON [PRIMARY]  GO |

جدول SensorTbl

|  |
| --- |
| USE [smartCattle]  GO  SET ANSI\_NULLS ON  GO  SET QUOTED\_IDENTIFIER ON  GO  CREATE TABLE [SmartCattle].[SensorTbl](  [ID] [int] IDENTITY(1,1) NOT NULL,  [MacAddress] [nvarchar](max) NULL,  [cattleId] [int] NULL,  [lastTransmitDate] [datetime2](7) NOT NULL,  [activationDate] [datetime2](7) NOT NULL,  [linkingDate] [datetime2](7) NOT NULL,  [lastSyncDate] [datetime2](7) NOT NULL,  [antennaId] [int] NOT NULL,  [antennaName] [nvarchar](max) NULL,  [status] [int] NOT NULL,  [softwareVersion] [nvarchar](max) NULL,  [FarmID] [int] NOT NULL,  [UserId] [nvarchar](max) NULL,  CONSTRAINT [PK\_SmartCattle.SensorTbl] PRIMARY KEY CLUSTERED  (  [ID] ASC  )WITH (PAD\_INDEX = OFF, STATISTICS\_NORECOMPUTE = OFF, IGNORE\_DUP\_KEY = OFF, ALLOW\_ROW\_LOCKS = ON, ALLOW\_PAGE\_LOCKS = ON) ON [PRIMARY]  ) ON [PRIMARY] TEXTIMAGE\_ON [PRIMARY]  GO  ALTER TABLE [SmartCattle].[SensorTbl] WITH CHECK ADD CONSTRAINT [FK\_SmartCattle.SensorTbl\_SmartCattle.FarmTbl\_FarmID] FOREIGN KEY([FarmID])  REFERENCES [SmartCattle].[FarmTbl] ([ID])  GO  ALTER TABLE [SmartCattle].[SensorTbl] CHECK CONSTRAINT [FK\_SmartCattle.SensorTbl\_SmartCattle.FarmTbl\_FarmID]  GO |

جدول CattlePositionTbl

|  |
| --- |
| USE [smartCattle]  GO  SET ANSI\_NULLS ON  GO  SET QUOTED\_IDENTIFIER ON  GO  CREATE TABLE [SmartCattle].[CattlePositionTbl](  [ID] [int] IDENTITY(1,1) NOT NULL,  [Latitude] [float] NOT NULL,  [Longitude] [float] NOT NULL,  [cattleId] [int] NOT NULL,  [date] [datetime2](7) NOT NULL,  [LastRecievedId] [bigint] NOT NULL,  [FarmID] [int] NOT NULL,  [FreeStall] [int] NOT NULL,  CONSTRAINT [PK\_SmartCattle.CattlePositionTbl] PRIMARY KEY CLUSTERED  (  [ID] ASC  )WITH (PAD\_INDEX = OFF, STATISTICS\_NORECOMPUTE = OFF, IGNORE\_DUP\_KEY = OFF, ALLOW\_ROW\_LOCKS = ON, ALLOW\_PAGE\_LOCKS = ON) ON [PRIMARY]  ) ON [PRIMARY]  GO  ALTER TABLE [SmartCattle].[CattlePositionTbl] WITH CHECK ADD CONSTRAINT [FK\_SmartCattle.CattlePositionTbl\_SmartCattle.CattleTbl\_cattleId] FOREIGN KEY([cattleId])  REFERENCES [SmartCattle].[CattleTbl] ([ID])  ON DELETE CASCADE  GO  ALTER TABLE [SmartCattle].[CattlePositionTbl] CHECK CONSTRAINT [FK\_SmartCattle.CattlePositionTbl\_SmartCattle.CattleTbl\_cattleId]  GO  ALTER TABLE [SmartCattle].[CattlePositionTbl] WITH CHECK ADD CONSTRAINT [FK\_SmartCattle.CattlePositionTbl\_SmartCattle.FarmTbl\_FarmID] FOREIGN KEY([FarmID])  REFERENCES [SmartCattle].[FarmTbl] ([ID])  GO  ALTER TABLE [SmartCattle].[CattlePositionTbl] CHECK CONSTRAINT [FK\_SmartCattle.CattlePositionTbl\_SmartCattle.FarmTbl\_FarmID]  GO |

جدول CattleTbl

|  |
| --- |
| USE [smartCattle]  GO  SET ANSI\_NULLS ON  GO  SET QUOTED\_IDENTIFIER ON  GO  CREATE TABLE [SmartCattle].[CattleTbl](  [ID] [int] IDENTITY(1,1) NOT NULL,  [age] [int] NOT NULL,  [preg] [int] NOT NULL,  [milkAvg] [float] NOT NULL,  [healthStatus] [int] NOT NULL,  [animalNumber] [int] NOT NULL,  [heatStatus] [int] NOT NULL,  [birthDate] [datetime2](7) NOT NULL,  [Dim] [int] NOT NULL,  [fertilityStatus] [int] NOT NULL,  [lactationNumber] [int] NOT NULL,  [InseminationCount] [int] NOT NULL,  [lastInseminationDate] [datetime2](7) NULL,  [lastCalvingDate] [datetime2](7) NULL,  [calvingCount] [int] NOT NULL,  [CattleGroupId] [int] NULL,  [FreeStallId] [int] NULL,  [FarmID] [int] NOT NULL,  [UserId] [nvarchar](128) NULL,  [CattleHerd\_ID] [int] NULL,  CONSTRAINT [PK\_SmartCattle.CattleTbl] PRIMARY KEY CLUSTERED  (  [ID] ASC  )WITH (PAD\_INDEX = OFF, STATISTICS\_NORECOMPUTE = OFF, IGNORE\_DUP\_KEY = OFF, ALLOW\_ROW\_LOCKS = ON, ALLOW\_PAGE\_LOCKS = ON) ON [PRIMARY]  ) ON [PRIMARY]  GO |

جدول CattleGroupTbl

|  |
| --- |
| USE [smartCattle]  GO  SET ANSI\_NULLS ON  GO  SET QUOTED\_IDENTIFIER ON  GO  CREATE TABLE [SmartCattle].[CattleGroupTbl](  [ID] [int] IDENTITY(1,1) NOT NULL,  [name] [nvarchar](max) NULL,  [Description] [nvarchar](max) NULL,  [order] [nvarchar](max) NULL,  [code] [nvarchar](max) NULL,  [FarmID] [int] NOT NULL,  [UserId] [nvarchar](128) NULL,  CONSTRAINT [PK\_SmartCattle.CattleGroupTbl] PRIMARY KEY CLUSTERED  (  [ID] ASC  )WITH (PAD\_INDEX = OFF, STATISTICS\_NORECOMPUTE = OFF, IGNORE\_DUP\_KEY = OFF, ALLOW\_ROW\_LOCKS = ON, ALLOW\_PAGE\_LOCKS = ON) ON [PRIMARY]  ) ON [PRIMARY] TEXTIMAGE\_ON [PRIMARY]  GO  ALTER TABLE [SmartCattle].[CattleGroupTbl] WITH CHECK ADD CONSTRAINT [FK\_SmartCattle.CattleGroupTbl\_SmartCattle.AspNetUsers\_UserId] FOREIGN KEY([UserId])  REFERENCES [SmartCattle].[AspNetUsers] ([Id])  GO  ALTER TABLE [SmartCattle].[CattleGroupTbl] CHECK CONSTRAINT [FK\_SmartCattle.CattleGroupTbl\_SmartCattle.AspNetUsers\_UserId]  GO  ALTER TABLE [SmartCattle].[CattleGroupTbl] WITH CHECK ADD CONSTRAINT [FK\_SmartCattle.CattleGroupTbl\_SmartCattle.FarmTbl\_FarmID] FOREIGN KEY([FarmID])  REFERENCES [SmartCattle].[FarmTbl] ([ID])  GO  ALTER TABLE [SmartCattle].[CattleGroupTbl] CHECK CONSTRAINT [FK\_SmartCattle.CattleGroupTbl\_SmartCattle.FarmTbl\_FarmID]  GO |

جدول EnvSensors

|  |
| --- |
| USE [smartCattle]  GO  SET ANSI\_NULLS ON  GO  SET QUOTED\_IDENTIFIER ON  GO  CREATE TABLE [SmartCattle].[EnvSensors](  [id] [int] IDENTITY(1,1) NOT NULL,  [FreeStallId] [int] NULL,  [FarmId] [int] NULL,  [Lat] [float] NULL,  [Lng] [float] NULL,  [MAC] [nvarchar](50) NULL  ) ON [PRIMARY]  GO |

جدول FarmTbl

|  |
| --- |
| USE [smartCattle]  GO  SET ANSI\_NULLS ON  GO  SET QUOTED\_IDENTIFIER ON  GO  CREATE TABLE [SmartCattle].[FarmTbl](  [ID] [int] IDENTITY(1,1) NOT NULL,  [name] [nvarchar](max) NULL,  [LogoUrl] [nvarchar](max) NULL,  [email] [nvarchar](max) NULL,  [website] [nvarchar](max) NULL,  [Latitude] [nvarchar](max) NULL,  [Longitude] [nvarchar](max) NULL,  [UserId] [nvarchar](max) NULL,  [MapGeoJson] [nvarchar](max) NULL,  [MapFilePath] [nvarchar](max) NULL,  [UerIdentify] [int] NULL,  CONSTRAINT [PK\_SmartCattle.FarmTbl] PRIMARY KEY CLUSTERED  (  [ID] ASC  )WITH (PAD\_INDEX = OFF, STATISTICS\_NORECOMPUTE = OFF, IGNORE\_DUP\_KEY = OFF, ALLOW\_ROW\_LOCKS = ON, ALLOW\_PAGE\_LOCKS = ON) ON [PRIMARY]  ) ON [PRIMARY] TEXTIMAGE\_ON [PRIMARY]  GO  ALTER TABLE [SmartCattle].[FarmTbl] WITH CHECK ADD CONSTRAINT [FK\_FarmTbl\_UserInfo] FOREIGN KEY([UerIdentify])  REFERENCES [SmartCattle].[UserInfo] ([ID])  GO  ALTER TABLE [SmartCattle].[FarmTbl] CHECK CONSTRAINT [FK\_FarmTbl\_UserInfo]  GO |

جدول FreeStallTbl

|  |
| --- |
| USE [smartCattle]  GO  SET ANSI\_NULLS ON  GO  SET QUOTED\_IDENTIFIER ON  GO  CREATE TABLE [SmartCattle].[FreeStallTbl](  [ID] [int] IDENTITY(1,1) NOT NULL,  [name] [nvarchar](max) NULL,  [Description] [nvarchar](max) NULL,  [code] [int] NOT NULL,  [location] [geography] NULL,  [FarmID] [int] NOT NULL,  [GroupID] [int] NULL,  [UserId] [nvarchar](128) NULL,  CONSTRAINT [PK\_SmartCattle.FreeStallTbl] PRIMARY KEY CLUSTERED  (  [ID] ASC  )WITH (PAD\_INDEX = OFF, STATISTICS\_NORECOMPUTE = OFF, IGNORE\_DUP\_KEY = OFF, ALLOW\_ROW\_LOCKS = ON, ALLOW\_PAGE\_LOCKS = ON) ON [PRIMARY]  ) ON [PRIMARY] TEXTIMAGE\_ON [PRIMARY]  GO  ALTER TABLE [SmartCattle].[FreeStallTbl] WITH CHECK ADD CONSTRAINT [FK\_SmartCattle.FreeStallTbl\_SmartCattle.AspNetUsers\_UserId] FOREIGN KEY([UserId])  REFERENCES [SmartCattle].[AspNetUsers] ([Id])  GO  ALTER TABLE [SmartCattle].[FreeStallTbl] CHECK CONSTRAINT [FK\_SmartCattle.FreeStallTbl\_SmartCattle.AspNetUsers\_UserId]  GO  ALTER TABLE [SmartCattle].[FreeStallTbl] WITH CHECK ADD CONSTRAINT [FK\_SmartCattle.FreeStallTbl\_SmartCattle.CattleGroupTbl\_GroupID] FOREIGN KEY([GroupID])  REFERENCES [SmartCattle].[CattleGroupTbl] ([ID])  GO  ALTER TABLE [SmartCattle].[FreeStallTbl] CHECK CONSTRAINT [FK\_SmartCattle.FreeStallTbl\_SmartCattle.CattleGroupTbl\_GroupID]  GO  ALTER TABLE [SmartCattle].[FreeStallTbl] WITH CHECK ADD CONSTRAINT [FK\_SmartCattle.FreeStallTbl\_SmartCattle.FarmTbl\_FarmID] FOREIGN KEY([FarmID])  REFERENCES [SmartCattle].[FarmTbl] ([ID])  ON DELETE CASCADE  GO  ALTER TABLE [SmartCattle].[FreeStallTbl] CHECK CONSTRAINT [FK\_SmartCattle.FreeStallTbl\_SmartCattle.FarmTbl\_FarmID]  GO |

جدول UserInfo

|  |
| --- |
| USE [smartCattle]  GO  SET ANSI\_NULLS ON  GO  SET QUOTED\_IDENTIFIER ON  GO  CREATE TABLE [SmartCattle].[UserInfo](  [ID] [int] IDENTITY(1,1) NOT NULL,  [Name] [nvarchar](max) NULL,  [Family] [nvarchar](max) NULL,  [Email] [nvarchar](max) NULL,  [Password] [nvarchar](max) NULL,  [Role] [nvarchar](max) NULL,  [Permissions] [nvarchar](max) NULL,  [FarmId] [int] NULL,  CONSTRAINT [PK\_UserInfo] PRIMARY KEY CLUSTERED  (  [ID] ASC  )WITH (PAD\_INDEX = OFF, STATISTICS\_NORECOMPUTE = OFF, IGNORE\_DUP\_KEY = OFF, ALLOW\_ROW\_LOCKS = ON, ALLOW\_PAGE\_LOCKS = ON) ON [PRIMARY]  ) ON [PRIMARY] TEXTIMAGE\_ON [PRIMARY]  GO |

جدول UserPermissions

|  |
| --- |
| USE [smartCattle]  GO  SET ANSI\_NULLS ON  GO  SET QUOTED\_IDENTIFIER ON  GO  CREATE TABLE [SmartCattle].[UserPermissions](  [ID] [int] IDENTITY(1,1) NOT NULL,  [Controller] [nvarchar](max) NULL,  [Action] [nvarchar](max) NULL,  [Comment] [nvarchar](max) NULL,  [UniqueId] [nvarchar](max) NULL  ) ON [PRIMARY] TEXTIMAGE\_ON [PRIMARY]  GO |

جدول UserRoles

|  |
| --- |
| USE [smartCattle]  GO  SET ANSI\_NULLS ON  GO  SET QUOTED\_IDENTIFIER ON  GO  CREATE TABLE [SmartCattle].[UserRoles](  [ID] [int] IDENTITY(1,1) NOT NULL,  [Name] [nvarchar](max) NULL,  [Permissions] [nvarchar](max) NULL,  [Comment] [nvarchar](max) NULL  ) ON [PRIMARY] TEXTIMAGE\_ON [PRIMARY]  GO |

جدول Notifications

|  |
| --- |
| USE [smartCattle]  GO  SET ANSI\_NULLS ON  GO  SET QUOTED\_IDENTIFIER ON  GO  CREATE TABLE [SmartCattle].[Notifications](  [ID] [int] IDENTITY(1,1) NOT NULL,  [Topic] [nvarchar](max) NULL,  [Comment] [nvarchar](max) NULL,  [FarmID] [int] NULL,  [RoleName] [nvarchar](max) NULL,  [CreatedDate] [nvarchar](max) NULL,  [Status] [nvarchar](max) NULL,  [NotificationType] [nvarchar](max) NULL,  [Snooze] [int] NULL,  [TagName] [nvarchar](max) NULL,  [SnoozeMsg] [nvarchar](max) NULL  ) ON [PRIMARY] TEXTIMAGE\_ON [PRIMARY]  GO |

جدول FreeStallNotificationsSetting

|  |
| --- |
| USE [smartCattle]  GO  SET ANSI\_NULLS ON  GO  SET QUOTED\_IDENTIFIER ON  GO  CREATE TABLE [SmartCattle].[FreeStallNotificationsSetting](  [ID] [int] IDENTITY(1,1) NOT NULL,  [FarmId] [int] NULL,  [GroupName] [nvarchar](max) NULL,  [Topic] [nvarchar](max) NULL,  [Roles] [nvarchar](max) NULL,  [Comment] [nvarchar](max) NULL,  [PeroidTime] [int] NULL,  [WindowTime] [int] NULL,  [TempMin] [float] NULL,  [TempMax] [float] NULL,  [HumMin] [float] NULL,  [HumMax] [float] NULL,  [THIMin] [float] NULL,  [THIMax] [float] NULL,  [CreateDate] [datetime2](7) NULL,  [ActivationState] [nvarchar](max) NULL,  [SnoozeTime] [nvarchar](max) NULL  ) ON [PRIMARY] TEXTIMAGE\_ON [PRIMARY]  GO |

جدول CattleNotificationsSetting

|  |
| --- |
| USE [smartCattle]  GO  /\*\*\*\*\*\* Object: Table [SmartCattle].[CattleNotificationsSetting] Script Date: 4/12/2018 10:05:39 PM \*\*\*\*\*\*/  SET ANSI\_NULLS ON  GO  SET QUOTED\_IDENTIFIER ON  GO  CREATE TABLE [SmartCattle].[CattleNotificationsSetting](  [ID] [int] IDENTITY(1,1) NOT NULL,  [FarmId] [int] NULL,  [GroupName] [nvarchar](max) NULL,  [Topic] [nvarchar](max) NULL,  [Roles] [nvarchar](max) NULL,  [Comment] [nvarchar](max) NULL,  [PeroidTime] [int] NULL,  [WindowTime] [int] NULL,  [CattleTempMin] [float] NULL,  [CattleTempMax] [float] NULL,  [SittingMin] [float] NULL,  [SittingMax] [float] NULL,  [WalkingMin] [float] NULL,  [WalkingMax] [float] NULL,  [RuminationMin] [float] NULL,  [RuminationMax] [float] NULL,  [DrinkingMin] [float] NULL,  [DrinkingMax] [float] NULL,  [EatingMin] [float] NULL,  [EatingMax] [float] NULL,  [StandingMin] [float] NULL,  [StandingMax] [float] NULL,  [CreateDate] [datetime2](7) NULL,  [ActivationState] [nvarchar](max) NULL,  [SnoozeTime] [nvarchar](max) NULL  ) ON [PRIMARY] TEXTIMAGE\_ON [PRIMARY]  GO |

فصل شش

توضیح اکشن های برنامه

مقدمه

در این فصل به توضیح خط به خط کدها پرداخته می شود. ساختاری که طبق آن این فصل شکل خواهد گرفت بدین شرح است. از mvc برای طراحی اسمارت کتل استفاده شده است. mvc مخفف کلمات Model View Controller هست. در واقع MVC بر روی معماری های چند لایه ای جهت جداسازی قسمت های مختلف برنامه است. هر یک از بخش های معماری MVC یعنی Model و View و controller را به شکل زیر تعریف می کنیم.

Model

در mvc مدل ها همان کلاس های برنامه هستند که شبیه به دیتابیس هستند ولی می توانند متفاوت و بیشتر باشند. همچنین Model وظیفه تایید داده ها را برعهده دارد تا اطلاعات درست درون آنها قرار بگیرد و برای دیتابیس ارسال شود.

View

این بخش اطلاعات را به کاربر نشان می دهد، ویو کاری به منطق برنامه ندارد و فقط یک model دریافت می کند و آن را به هر شکلی که نیاز باشد به کاربر نشان می دهد، این model هم توسط کنترلر ایجاد شده است و برای view رسال شده است. نکته مهمی که در بخش View باید مد نظر داشت این است که این لایه مسئول کنترل صحت داده های وارد شده از طریق کاربر و همچنین مسئول صحت داده های نشان داده شده به کاربر نیست . در واقع این بخش با داده های خام کار میکند و مسولیت صحت داده با model است.

Controller

این بخش همانگونه که از اسم آن مشخص است، یک بخش کنترل کننده است، ساده بگویم controller بیشتر مواقع دو جور کار می کند :

controller اطلاعات را از دیتابیس در قالب model هایی می گیرد و بعد پردازش خود را روی آن انجام می دهد و اطلاعات را در قالب model جدیدی به view می فرستد تا ویو هر طور خواست این اطلاعات را به کاربر نشان دهد.مثل فرض کنید که ما برای صفحه پروفایل یک view برای موبایل داریم و یک view برای ویندوز، حال کنترولر بنا به تشخیص که کاربر با موبایل است یا نه یک view را انتخاب می کند تا برای کاربر ارسال شود و می بینید که یک اطلاعات مشخص در دو view متفاوت ظاهر می شود و view (ظاهر) از منطق برنامه جدا است.

controller اطلاعات را از مدلی که از view برای او ارسال کرده است دریافت می کند و پس انجام اقداماتی آن را در دیتابیس ذخیره می کند و view دیگری را به کاربر می فرستد تا ببیند. به زبان ساده Controller ها بخش اصلی از سناریوی یک برنامه هستند. بنابراین کدهای نرم افزار اسمارت کتل بر اساس Controller ها به ترتیب حروف الفبا شرح داده خواهد شد.

AccountController

وظیفه کنترلر Account، بررسی معتبر بودن کاربر و لاگین کردن وی به اسمارت کتل است. این کنترل با استفاده از اکشن Login صلاحیت کاربر را برای ورودی به نرم افزار بررسی می کند. در ادامه به توضیح اکشن های کنترلر AccountController پرداخته می شود.

|  |  |  |
| --- | --- | --- |
|  |  |  |
|  | [HttpPost]  [AllowAnonymous]  [ValidateAntiForgeryToken]  public async Task<ActionResult> Login(LoginViewModel model, string returnUrl) |  |
|  |  |  |

از این متد برای بررسی صحت یوزر و پسورد وارد شده توسط کاربر استفاده می شود. اگر کاربر یوزر و پسورد اشتباهی را وارد کند زمان و آی کاربر در دیتابیس ذخیره می شود. اگر برای بار دوم پسورد اشتباهی وارد کند، مدت زمان وارد کردن دو پسورد با هم مقایسه می شود. اگر زمان آن ها زیر 3 ثانیه باشد، اون نام کاربری غیر فعال می شود. در غیر این صورت کاربر می تواند تا 5 بار پسورد اشتباه وارد کند. در بار 5ام نام کاربری غیر فعال می شود.

|  |  |  |
| --- | --- | --- |
|  |  |  |
|  | [HttpPost]  [AllowAnonymous]  [ValidateAntiForgeryToken]  public async Task<ActionResult> Register(RegisterViewModel model) |  |
|  |  |  |

از این اکشن برای ثبت نام کاربر استفاده می شود. ابتدا اطلاعات وارد شده کاربر به یک مدل تبدیل می شود و سپس این مدل از نظر وجود موارد امنیتی بررسی می شود. اگر مشکلی وجود نداشت داده ها در دیتابیس ذخیره می شود. البته در مورد رعایت موارد امنیتی به صورت مفصل در فصل های بعدی توضیح داده شده است.

CattleController

یکی از مهمترین کنترلر های اسمارت کتل CattleController است. در ادامه در مورد اکشن های تشکیل دهنده آن صحبت خواهد شد.

فصل چهار

امنیت

مقدمه

در این فصل در مورد انواع حملات به برنامه های تحت وب و راه های مقابله با آن صحبت خواهد شد. همچنین سعی می شود به طور دقیق در مورد تست های نفوذ توضیح داده شود.

حمله SQL Injection

یک حمله SQL Injection شامل وارد کردن و یا تزریق یک کوئری SQL از طریق اطلاعات ورودی از سوی کاربر به برنامه می باشد.یک exploit موفق که منجر به تزریق SQL بشود می تواند اطلاعات حساس را از دیتابیس بخواهند و یا اطلاعات را حذف و یا ویرایش نماید. کار دیگری که می تواند انجام دهد این است که بتواند DBMS را خاموش کند یا این که محتویات یک فایل بر روی سیستم DBMS را فراخوانی کند و در برخی حالات هم می تواند فرمان هایی را برای سیستم عامل صادر کند.

برای مثال صفحه لاگین اسمارت کتل رو نظر بگیرید. صفحات Login معمولا دارای فیلدهایی هستند که دو مقدار ID و Password را از کاربر گرفته و سپس با استفاده از یک دستور SQL آن را پردازش می کنند. نمونه ساده ای از این دستور به صورت زیر است:

|  |  |  |
| --- | --- | --- |
|  |  |  |
|  | SELECT COUNT(UserID) FROM tblUsers WHERE UserID=''' & UserID.Text & ''' AND Pass=''' & Password.Text |  |
|  |  |  |

در این مثال UserID و Password دو کنترل TextBox هستند که مقادیر آنها بایستی مورد پردازش قرار گیرد. حال فرض می کنیم کاربر مقادیر را به صورت Pooya و 123 وارد نماید، در این صورت جمله SQL به صورت زیر تولید می شود:

|  |  |  |
| --- | --- | --- |
|  |  |  |
|  | SELECT COUNT(UserID) FROM tblUsers WHERE UserID='Pooya' AND Pass=’123’ |  |
|  |  |  |

خوب تا اینحا مشکلی وجود ندارد. حال فرض کنید که هکر به جای کلمه کاربری خود عبارت زیر را وارد نماید:

|  |  |  |
| --- | --- | --- |
|  |  |  |
|  | ' OR 1=-1 |  |
|  |  |  |

تکه کدی که در بالا درج شده دقیقا باید به همین شکل باشد. یعنی وجود تک کوتیشن و مساوی عبارت نامعادل یک مساوی یک منفی می بایست به همین ترتیب درج شود. در این صورت عبارت SQL زیر تولید خواهد شد:

|  |  |  |
| --- | --- | --- |
|  |  |  |
|  | SELECT COUNT(UserID) FROM tblUsers WHERE UserID='' OR 1=1 – AND PASS='' |  |
|  |  |  |

کاراکتر - در SQL علامت توضیحات است و عبارت پس از آن دیگر پردازش نمی شود لذا هنگام پردازش دستور چون همواره 1=1 است و با توجه به OR به کار رفته، بدون توجه به خالی بودن مقدار مقابل UserID ، کوئری از نظر دیتابیس کافی تلقی شده و کلیه رکوردهای بانک اطلاعاتی برگردانده می شود و Attacker اصطلاحا Authenticate می شود. از این نوع دستورات که به دستورات SQL Injection معروف هستند زیاد هست و راه های مختلفی هم برای مقابله با آن ها وجود دارد.

ساختار شی گرای ASP.NET که اسمارت کتل بر اساس آن نوشته شده است و امکانات این ساختار به برنامه نویسان امکان مانور بیشتری را داده است. حال ما کد کامل اصلاح شده را در زیر آورده و سپس توضیحات آن را ذکر می کنیم:

|  |  |  |
| --- | --- | --- |
|  |  |  |
|  | Dim strSQL As String = ''SELECT COUNT(UserID) FROM tblUsers WHERE UserID=@UserID AND Password=@Password''  Dim cmndCheck As OleDbCommand = New OleDbCommand(strSQL, \_Connection) cmndCheck.Parameters.Add(''@UserID'', UserID.Text);  cmndCheck.Parameters.Add(''@Password'', Password.Text);  cmndCheck.Connection.Open()  Dim IsValid As Integer = cmndCheck.ExecuteScalar()  If IsValid > 0  '... Some Code here... User is authenticated  Else  '... Some Code here...  User is not aututorized to view the page End If |  |
|  |  |  |

تغییر در همان خط اول یعنی دستور SQL مشخص است این بار به جای اینکه مقادیر دریافتی از فیلدها با Single Quote به خود پردازشگر دستور داده شود مقادیر با پارامترهای تولید شده توسط آبجکت OleDbCommand جایگزین می شود و در آخر cmndCheck پردازش شده چنانچه مقدار تولید شده توسط ExecuteScalar بزرگتر از صفر باشد بدین معنی است که حداقل یک رکورد با شرایط مورد نظر ما پیدا شده است. (متغیر Connection\_ که باید برابر ConnectionString اسمارت کتل باشد) تفاوت اصلی این روش نسبت به روش ناامن قبلی این است که در اینجا مقادیر به صورت کاملا پارامتری شده پردازش می شوند نه به صورت قطعاتی از یک رشته حرفی که در واقع دام اصلی SQL Injection به شمار می آید. در واقع ما در اینجا از تکنیکهای رشته سازی با استفاده از & یا + اجتناب کرده ایم.

نکته مهم دیگری که باید حتما مورد توجه قرار گیرد این است که یکی از راههایی که هکرها برای حمله به نرم افزار استفاده می کنند خطاسازی صوری در زمان اجرای Application و مطالعه اطلاعات خطای دریافت شده است. برای جلوگیری از این اتفاق ساختار بعضی از قسمتهای کد در Application از روال خطایابی Customize شده استفاده می شود. به عنوان مثال وقتی کاربری پسورد خود را اشتباه وارد می کند، به جای آن که به وی گفته شود که پسورد اشتباه است، پیامی مبتنی بر عدم دسترسی کاربر برای وی به نمایش در میاید. به این ترتیب هکر نمی تواند متوجه شود که آیا نام کاربری را اشتباه وارد کرده است یا کلمه عبور.

مثال ها و راه حل های مختلفی برای حملات Injection وجود دارد. برای بررسی بیشتر مثال ها می توان به سایت های زیر مراجعه کرد.

|  |  |  |
| --- | --- | --- |
|  |  |  |
|  | [www.sqlsecurity.com](http://www.sqlsecurity.com)  [www.nextgenss.com](http://www.nextgenss.com) |  |
|  |  |  |

چنین سیستمی در ابتدای پیاده سازی اسمارت کتل در نظر گرفته شده بود. این سیستم خوبی ها و بدی های مختلف دارد. یکی از مزایای آن سادگی در فهم و پیاده سازی کوئری های ساخته شده است ولی از بدی های آن می توان به عدم پشتیبانی از رنج وسیعی از تزریق ها نام برد. برای بهبود حفاظت اسمارت کتل در مقابل حمله های مبتنی بر Injection، از First Code Entity و landa Expression ها استفاده شده. به این ترتیب که کوئری های مورد نیاز به صورت Landa به ORM داده می شود و سپس توسط ORM کار مورد نظر بر روی دیتابیس انجام می شود.

حملاتی از نوع Variable Manipulation و Buffer Overflow

علاوه برحملات SQL Injection عليه برنامه هاي تحت وب، دو دسته دیگر از حملات با نام های Variable Manipulation و Buffer Overflow وجود دارد. در اين تكنيك، مهاجم سعي بر تغيير متغيرهاي موجود در برنامه دارد. در نتيجه اين تغييرات، منطق برنامه دچار مشكل مي شود. مثال كلاسيك اين نوع حمله eShoplifting نام دارد كه در آن مهاجم با دسترسي به يك يا چند متغير قيمت در وب سايت فروش آنلاين و تغيير مقدار آنها، منطق برنامه را فريب داده و منجر به محاسبه اشتباه قيمت محصول توسط برنامه مي شود. در نتيجه مهاجم مي تواند اقدام به خريد يك آيتم گران قيمت با قيمت بسيار پايين كند. در اكثر موارد برنامه قادر به تشخيص تغيير قيمت نيست و روند كار در حالت عادي انجام خواهد شد. براي بررسي وجود چنين رخنه پذيري در اسمارت کتل به این صورت عمل شد:

كد HTML صفحات اسمارت کتل بررسی شد و متغیرهایی که موارد مهم نظیر نام جداول و یا آی دی برخی سطرها را در خود نگه داری می کردند رمزگذاری شدند. شیوه رمزگذاری به این صورت است که هربار که از سمت View درخواستی به یکی از کنترلرها ارسال می شود، زمان جاری از کنترلر Utile دریافت می شود. سپس این زمان با یک عدد رندم و یک delimiter با دیتای مورد نظر ترکیب و در نهایت نتیجه انکریپت می شود. داده انکریپت شده به کنترلر مورد نظر ارسال می شود و در آن جا ابتدا دکریپت می شود و سپس با delimiter بخش های مختلف داده از هم مجزا می شود. ابتدا زمان چک می شود. اگر اختلاف زمان داخل پکت از زمان جاری بیش از ده ثانیه باشد نشان می دهد که هکر قصد دارد با ارسال یک پکت Sniff شده به دفعات مختلف، به سیستم نفوذ کند. این یعنی هکر تونسته یک پکت سالم رو از سیستم پیدا کنه ولی نتونسته الگوریتم انکریپشن آن را مورد هدف قرار دهد. با توجه به این که Threshold ده ثانیه ای برای معتبر بودن پکت ها در نظر گرفته شده، هر پکت ساخته شده پس از سپری شدن زمان اعتبارش دیگر هیچ ارزشی ندارد.

یکی دیگر از حملاتی که در اسمارت کتل بررسی شد Buffer overflow نام دارد. همانطور كه مي دانيد، يك برنامه حجم مشخصي از حافظه را اشغال مي كند. درصورتي كه اطلاعاتي بيشتر از فضاي اختصاص يافته به برنامه (حتي باندازه يك بايت)، وارد حافظه شود، منجر به سرريزي خواهد شد. در زير به بيان دو نوع رايج سرريز به نام هاي" سرريزي پشته" و "سرريزي هيپ" پرداخته می شود:

Stack Overflow پشته، بخشي از حافظه است. سیستم اطلاعاتي را كه نمي تواند در رجيسترهاي خود ذخيره كند، در پشته ذخيره مي كند. در اين تكنيك مهاجم با آگاهي از عدم كنترل نوع و سايز مقادير ورودي يك تابع توسط برنامه، اقدام به ارسال مقادير بيشتر از فضاي حافظه و نهايتا بازنويسي آدرس تابع مخرب خود به جاي آدرس بازگشت تابع اصلي مي كند. تابع مخرب عبارتست از هر آن که چه منجر به خطر پذيري برنامه و سرور شود، مثلا در محيط لينوكس دستور ساده EXEC("sh") منجر به باز شدن پنجره دستورات و خطرپذيري بسيار براي سرور مي شود، حال وقتي برنامه قفل مي شود، اقدام به بازيابي توسط آدرس بازگشت مي كند و چون اين آدرس به دستور مخرب مشخص شده از سوي مهاجم تغيير كرده است، منجر به اجراي دستور مخرب خواهد شد.

در حمله از نوع Heap Overflow وقتي يك برنامه با حجم بزرگي از اطلاعات نيازمند پردازش مواجه مي شود، بخشي از حافظه به نام هيپ به منظور مديريت اطلاعات فوق در نظر گرفته مي شود. در زبان های سطح پایین، برای هندل کردن چنین موضوعی، لازم است تا برنامه نویس میزان حافظه مورد نیاز را خودش تعیین کند ولی در زبان های برنامه نویسی سطح بالا نظیر سی شارپ که اسمارت کتل با آن نوشته شده است، برنامه نویس نیازی به تنظیم میزان حافظه ندارد مگر آن که ضرورتی در کار تشخیص داده شود. زیرا چنین زبان های برنامه نویسی به صورت خودکار و با توجه به برنامه نوشته شده، میزانی از حافظه را به کد تخیصص می دهند. در نرم افزار نوشته شده اسمارت کتل تمامی کدهایی که به نحوی به صورت دستی حافظه ای را به کد تخصیص می دادند حذف شدند و کد جدیدی به جای آن ها درج شد.

حملات از نوع CSRF

نوع دیگری از حملات که باید در برنامه‌های وب به آن‌ها دقت داشت به نام CSRF یا Cross site request forgery معروف هستند. برای مثال فرض کنید کاربری قبل از اینکه بتواند در نرم افزار کار خاصی را انجام دهد، نیاز به اعتبار سنجی داشته باشد. پس از لاگین شخص و ایجاد کوکی و سشن معتبر، همین شخص به سایت دیگری مراجعه می‌کند که در آن مهاجمی بر اساس وضعیت جاری اعتبار سنجی او مثلا لینک حذف کاربری یا افزودن اطلاعات جدیدی را به برنامه ارائه می‌دهد. چون سشن شخص و کوکی مرتبط به سایت اول هنوز معتبر هستند و شخص سایت را نبسته است، «احتمال» اجرا شدن درخواست مهاجم بالا است. خصوصا اگر از مرورگرهای قدیمی استفاده کند. بنابراین نیاز است بررسی شود آیا درخواست رسیده واقعا از طریق فرم‌های برنامه ما صادر شده است یا اینکه شخصی از طریق سایت دیگری اقدام به جعل درخواست‌ها کرده است.

CSRF یا Cross Site Request Forgery به صورت خلاصه به این معنا است که شخص مهاجم اعمالی را توسط کاربر و با سطح دسترسی وی بر روی سایت انجام دهد و اطلاعات مورد نظر خود را استخراج کرده و به هر سایتی که تمایل دارد ارسال کند. این‌کار عموما با تزریق کد در صفحه صورت می‌گیرد.

برای مقابله با این نوع خطاها ابتدا داخل برنامه اسمارت کتل از متد Html.AntiForgeryToken استفاده شد. کار این متد ایجاد یک فیلد مخفی با مقداری منحصربفرد بر اساس اطلاعات سشن جاری کاربر است، به علاوه ارسال یک کوکی خودکار تا بتوان از تطابق اطلاعات اطمینان حاصل کرد. در مرحله بعد فیلتر ValidateAntiForgeryToken جهت بررسی مقدار token دریافتی به متد ثبت اطلاعات اضافه شد. سپس مقدار دریافتی از فیلد مخفی از قبل درج شده در فرم با مقدار موجود در کوکی سایت بررسی و تطابق داده شد. اگر این مقادیر تطابق نداشته باشند، یک پیام عدم دسترسی صادر شده و از پردازش اطلاعات رسیده جلوگیری می‌شود.

علاوه بر این‌ها حین استفاده از متد و فیلتر یاد شده، از یک salt مجزا نیز به ازای هر فرم، استفاده می شود. به این ترتیب tokenهای تولید شده در فرم‌های مختلف سایت یکسان نخواهند بود.

به علاوه باید دقت داشت که ValidateAntiForgeryToken فقط با فعال بودن کوکی‌ها در مرورگر کاربر کار می‌کند و اگر کاربری پذیرش کوکی‌ها را غیرفعال کرده باشد، قادر به ارسال اطلاعاتی به برنامه نخواهد بود. همچنین این فیلتر تنها در حالت HttpPost قابل استفاده است. در جاهایی که با HttpGet ساده کار می‌کند،‌ تبدیل به HttpPost شدند تا میزان امنیت برنامه بهبود یابد. از HttpGet فقط برای گزارشگیری و خواندن اطلاعات از برنامه استفاده می شود و نه ثبت اطلاعات.

حمله XSS

در این نوع حمله هکر به نرم افزارهای تحت وب از طریق اسکریپت نویسی حمله می‌کند و به دنبال فیلد های ورود اطلاعاتی می‌گردد و در صورت عدم اعتبارسنجی یا Validation فیلدهای ورودی می‌تواند به آن نرم افزار حمله کند. هدف اصلی این نوع حمله بدست آودن اطلاعات از کاربران و کلاینت هایی است که به سرور متصل می‌شوند. بیشتر اطلاعاتی که از طریق XSS بدست می آید از طریق Cookie هایی است که کلاینت بر روی مرورگر خود بر جای می گذارد. در حملات XSS ، کوکی مرورگر قربانی به سرقت میره. برای آن که در اسمارت کتل بتوان با چنین حمله هایی مقابله کرد چند کار صورت گرفت. در ابتدا فیلدهای ورود اطلاعات regular expression اعتبارسنجی می شود و به کاربران اجازه داده نمی شود که کد مخرب وارد کنند. همچنین در قدم بعدی با استفاده از متدهای جلوگیری از تزریق کدهای HTML و JavaScript، جلوی Injection گرفته می شود. یکی دیگر از کارهایی که انجام ی شود استفاده از سشن ها به جای کوکی هاست. در هر جای برنامه که از کوکی استفاده شد، تمام تبدیل به سشن شدند و همچنین با استفاده از یک Salt تبدیل به یک کد یونیک شدند.

فصل چهار

تنظیمات مربوط به سرور

ایجاد حساب کاربری FTP

برای آپلود فایل های پروژه بر روی سرور راه های بسیاری وجود دارد. یکی از راه های استاندارد و اصولی استفاده از پورتکل FTP است. برای ایجاد پروتکل طبق روش زیر عمل می کنیم.

۱- فعال کردن رول‌های WebServer (IIS) و FTP Server

1. Server Manager را اجرا کنید
2. به Manage > Add Roles and Features بروید.
3. Role-based or feature-based installation را انتخاب کنید.
4. سرور خود را انتخاب کنید
5. تیک Web Server (IIS) را بزنید
6. یک پاپ آپ باز می‌شود که باید از بین گزینه‌های موجود تیک Include management tools (if applicable) را بزنید.
7. دکمه Add Features را بزنید
8. دوباره پایین‌تر آمده و تیک‌های FTP server ، FTP Service و FTP Extensibility را هم بزنید.
9. در این بین Next ها را زده و در نهایت Install را بزنید و پس از اتمام پنجره را ببندید.

۲- ساخت کاربر FTP

1. Server Manager را اجرا کنید
2. در زیر مجموعه Tools گزینه Computer Management را بزنید
3. روی Local Users and Groups کلیک کنید. در صورتی که این گزینه را مشاهده نمی‌کنید باید کاربران خود را روی دامین تعریف کنید (از طریق Active Directory Computer and users در منوی Tools در Server Manager)
4. روی Users کلیک کنید
5. وسط صفحه کلیک راست کرده و New User… را بزنید.
6. اطلاعات کاربر را وارد کرده و Create را بزنید
7. هر تعداد کاربر می‌خواهید ایجاد کنید.

۳- تنظیمات کلی FTP روی IIS

یک سری تنظیمات کلی باید روی IIS خود اعمال کنید تا بتوانید از FTP استفاده کنید. این تنظیمات ساده‌است. کافیست مراحل زیر را انجام دهید:

1. در Server Manager به Tools بروید.
2. روی Internet Information Services (IIS) manager کلیک کنید
3. از پنل سمت چپ روی سرور دابل‌کلیک کنید (ریشه درخت در زیر Start Page)
4. اگر پاپ‌آپی باز شد در باره Microsoft Web Platform، تیک Do not show this message را بزنید و No رو بزنید.
5. در پنل وسط FTP Authentication را دابل‌کلیک کنید
6. اگر می‌خواهید همه بتوانند به FTP دسترسی داشته‌باشند (بدون نیاز به نام کاربری و کلمه عبور) روی Anonymous Authentication کلیک‌راست کرده و Enable را بزنید.
7. اگر می‌خواهید به کاربرانی که در مرحله ۲ بالا ایجاد کردید دسترسی دهید روی Basic Authentication کلیک‌راست کرده و Enable را بزنید.
8. دوباره از پنل سمت چپ روی سرور دابل‌کلیک کنید
9. در پنل وسط FTP Authorization Rules را دابل‌کلیک کنید.
10. تمام قوانین موجود را حذف کنید (در صورت وجود و در صورت نیاز)
11. بعد از حذف تمام قوانید وسط صفحه کلیک‌راست کرده و Add Allow Rule… را بزنید.
12. گزینه Specified users را انتخاب کنید.
13. در کادر مربوطه کاربرانی که در مرحله قبل تعریف کردی را به صورتی که با کاما از هم جدا شده‌اند وارد کنید.
14. سطح دسترسی‌های Read و Write مد نظر خود را به کاربران انتخاب شده بدهید.
15. روی OK کلیک کنید
16. مراحل ۸ تا ۱۵ بالا را به تعداد دلخواه خود تکرار کنید و کاربران بیشتری با سطح دسترسی‌های دلخواهد اضافه کنید.

۴- ساخت وب‌سایت FTP

1. یک فولدر در کامپیوتر خود ایجاد کنید
2. آدرس آن را کپی کنید. یا اینکه یک فولدر که می‌خواهید روی آن دسترسی FTP ایجاد کنید را باز کرده و آدرس آن را کپی کنید.
3. Server Manager را باز کنید
4. به Tools بروید
5. روی Internet Information Services (IIS) Manager کلیک کنید
6. از پنل سمت چپ روی سرور کلیک‌راست کنید (ریشه درخت در زیر Start Page)
7. Add FTP Site را بزنید
8. یک نام برای سایت خودوارد کنید مثلا Armandar FTP Site
9. آدرسی که در بالا (مرحله ۲) کپی کرده بودید در بخش Physical path وارد کنید
10. Next کنید
11. در لیست IP ها IP سرور خود را انتخاب کنید
12. پورت ۲۱ به صورت پیشفرض انتخاب شده می‌توانید پورت دیگری را انتخاب کنید
13. مطمئن شوید تیک Start FTP site automatically خورده شده است.
14. گزینه No SSL را انتخاب کنید. (اگر گواهینامه SSL ندارید)
15. Next کنید.
16. تیک Basic را بزنید تا به کاربرای که مشخص می‌کنید دسترسی به این وب‌سایت FTP داده‌شود.
17. در بخش Allow access گزینه Specified Users را انتخاب کنید
18. در محل مربوطه نامهای کاربری مدنظر خود را وارد کنید.
19. تیک‌های Read و Write را متناسب به نیاز خود انتخاب کنید.
20. Finish را بزنید.

۵- تنظیمات Firewall در IIS

1. در Server Manager به Tools بروید
2. روی Internet Information Services (IIS) manager کلیک کنید
3. از پنل سمت چپ روی سرور دابل‌کلیک کنید (ریشه درخت در زیر Start Page)
4. در پنل وسط FTP Firewall Support را دابل‌کلیک کنید
5. در بخش Data Channel Port Range از 0-0 برای استفاده جهت پورت‌های پیشفرض استفاده کنید
6. یا آن را در صورت نیاز خود تغییر دهید.
7. Apply کنید
8. IIS را ببندید

۶- تنظیمات Firewall در ویندوز

به صورت پیشفرض قواعد مربوط جهت دسترسی به FTP در زمانی که رول مربوط به آن را نصب می‌کنید به صورت اتوماتیک اضافه می‌شود. با این وجود من برای اینکه در صورت پیدا کردن مشکل بتوانید راهی جهت رفع آن پیدا کنید راهکار مربوط به افزودن تنظیمات مربوط به FTP در فایروال ویندوز را به شما آموزش می‌دهم.

1. Server Manager را باز کنید
2. در پنل سمت چپ روی Local Server کلیک کنید
3. در پنل سمت راست لینک مقابل Windows Firewall را کلیک کنید.
4. وقتی فایروال باز شد در پنل سمت چپ روی Advanced Settings کلیک کنید
5. وقتی باز شد Inbound Rules را کلیک کنید.
6. در پنل سمت راست چک کنید ببینید یک قاعده با نام FTP Server (FTP Traffic-In) وجود دارد
7. روی آن دابل کلیک کنید
8. در تب General چک کنید تیک Enabled خورده است.
9. به تب Protocols and Ports بروید
10. پروتوکل باید روی TCP و پورت باید روی ۲۱ باشد. (یا پورتی که شما انتخاب کرده بودید)
11. به تب Advanced بروید
12. مطمئن شوید پروفایل‌های Domain ، Private و Public تیک خورده‌اند.
13. Ok کنید.
14. مراحل ۷ تا ۱۳ را برای قاعده با نام FTP Server Passive (FTP Passive Traffic-In) هم بررسی کنید. با این تفاوت که پورت برای این قاعده باید 1024-65535 باشد.
15. مراحل ۷ تا ۱۳ را برای قاعده با نام FTP Server Secure (FTP SSL Traffic-In) هم بررسی کنید. با این تفاوت که پورت برای این قاعده باید 990 باشد.
16. در پنل سمت چپ Outbound Rules را کلیک کنید.
17. مراحل ۷ تا ۱۳ را برای قاعده با نام FTP Server (FTP Traffic-Out) هم بررسی کنید. با این تفاوت که پورت برای این قاعده باید 20 باشد.
18. مراحل ۷ تا ۱۳ را برای قاعده با نام FTP Server Secure (FTP SSL Traffic-Out) هم بررسی کنید. با این تفاوت که پورت برای این قاعده باید 989 باشد.
19. تمام پنجره‌ها را ببندید.

بعد از آن که مراحل بالا طی شدند امکان دارد که هنوز برای وصل شدن با FTP مشکلی وجود داشته باشد. دو کار لازم هست که انجام شود. اولی ایجاد یک Virtual Directory برای وب سایتی که ایجاد شده و تخصیص FTPUSER ایجاد شده به آن. کار دوم که به مراتب مهمتر هست Share پوشه ایجاد شده و دادن دسترسی به FTPUSER ایجاد شده. برای این کار روی پوشه کلیک راست کنید. در پنجره باز شده روی Edit کلیک کنید. در پنجره Permission for Folder Name بر روی Add کلیک شود. در پنجره Select Users بر روی Advance کلیک شود و یوزری که برای FTP ساختید رو پیدا کنید. سپس آن را انتخاب کنید و تمام گزینه ها را اوکی کنید. به یاد داشته باشید که پرمیژن ها برای Read, Read & Execute, and List Folder Contents می بایست فعال باشند.

7- آپلود Build اسمارت کتل بر روی FTP با استفاده از Git

فصل پنج

رفع مشکلات حین برنامه نویسی

مقدمه

حین برنامه نویسی مشکلاتی پیش میاد که اغلب وقت زیادی را برای برطرف شدن صرف می کند. در این فصل به بررسی علل پیشامد و شیوه برطرف کردن مشکلات پرداخته می شود. ساختار این فصل به این صورت هست که ابتدا شرحی در مورد مشکل پیش آمده داده می شود سپس دلیل و در انتها راه حل آن بیان می شود. پس هر مشکل از سه بخش مجزا از هم تشکیل شده است.

|  |
| --- |
| مشکل یک |
| وقتی متد جدیدی نوشته می شود، در هر بار اجرا برای بار نخست کار نمی کند ولی برای دفعات بعدی اجرا می شود. مثالی که می توان در مورد این مشکل مطرح کرد این است که، فرض کنیم توسعه دهنده دکمه ای را برای انجام کاری در صفحه قرار داده. هر بار که برنامه اجرا می شود و کاربر بر روی دکمه مورد نظر کلیک می کند، اتفاقی نمی افند. ولی اگر مجدد روی دکمه کلیک کند، اکشن تعریف شده برای آن دکمه اجرا می شود. |
| دلیل رخدادن مشکل یک |
| دلیل به وجود آمدن چنین مشکلی این است که، بخشی از اکشن های مربوط به آن دکمه در دیتابیس مربوط به بررسی دسترسی ها ذکر شده و بخش دیگر همچنان توسط برنامه نویس بدون تعریف باقی مانده. معمولا این مشکل برای آن دسته از ماژول هایی پیش میاید که جز Partial View ها باشند. پس اگر مشکلی که مشاهده می کنید برای یک ماژول Partial هست و هنوز برای آن دسترسی در دیتابیس تعریف نکردید، راه حل ذیل، مشکل شما را حل خواهد کرد. |
| راه حل مشکل یک |
| در دیتابیس ActionControllerTbl نام اکشن و کنترلر اخیری که ایجاد شده است را درج نمایید. اگر ماژول مربوطه از نوع Partial هست فیلد PartialView را True و در غیر این صورت False قرار دهید و مجدد برنامه را اجرا نمایید. به این ترتیب باید مشکل شما حل شده باشد. |

|  |
| --- |
| مشکل دو |
| گاهی پیش میاد که لازم هست در سیستم لوکال، برای ربیت برنامه جدیدی نوشت یا تغییراتی در برنامه های مربوط به ربیت ایجاد کرد. ولی مساله این جاست که موقع کار روی سیستم لوکال، عموما برنامه با پیام خطایی مبنی بر عدم وجود سرور یا آدرس ربیت مواجه می شود. |
| دلیل رخدادن مشکل دو |
| فعال نبودن سرور ربیت روی سیستم لوکال باعث بروز چنین پیام خطایی می شود. |
| راه حل مشکل دو |
| برای فعال کردن سرور روی سیستم لوکال ابتدا به محل نصب ربیت روی سیستم خود مراجعه کنید. معمولا ربیت در مسیر زیر نصب می شود.  C:\Program Files\RabbitMQ Server\rabbitmq\_server-3.6.4  سپس وارد پوشه sbin شوید و بر روی فایل زیر دابل کلیک کنید تا اجرا شود.  rabbitmq-server.bat  اگر بعد از اجرای فایل بالا، در پنجره ظاهر شده با پیام خطایی مواجه شدید ابتدا rabbitmq-service.bat را فعال کنید تا سرویس ربیت فعال شود و سپس مجدد فایل مربوط به فعال سازی سرور را اجرا نمایید. توجه داشته باشید که سرور و کلاینتی که برای ربیت می نویسید از سروری که در بالا فعال کردید مجزا هستند. |

|  |
| --- |
| مشکل سه |
| روی گیت، یک آدرس وب هوک ست شده است به نحوی که بلافاصله بعد از Push شدن، باعث آپدیت شدن اسمارت کتل روی سایت می شه. موقع کدنویسی ممکن هست کد CMD رو اجرا نکنه. |
| دلیل رخدادن مشکل سه |
| معمولا IIS به برنامه های ASP اجازه نمی دهد تا یک فایل اجرایی را اجرا کنند. برای همین جلوی اجرا شدن آن ها را می گیرد. |
| راه حل مشکل سه |
| توی Pool مربوط به وب سایتی که برنامه در آن اجرا می شود مقدار LoadUserProfile باید True باشد. همچنین خط زیر نیز باید به برنامه اضافه شود.  proc.StartInfo.LoadUserProfile = true; |

|  |
| --- |
| مشکل چهار |
|  |
| دلیل رخدادن مشکل چهار |
|  |
| راه حل مشکل چهار |
|  |

فصل شش

ضمیمه

1

IPARS DB

IPARS

SmartCattle DB

SmartCattle

Create a new account

Use available accounts to create a new farm

2

Call “createSubproject” API to get the “spID” and “Data queue” and “Equipment queue” names

Call “CreatSubproject” API

Generate “spID” “Data queue” and “Equipment queue” names

Get spID and queue names for the new sub-project

If approved by admin

3

Import cattle list

\* Generate initial “config” files and two initial mapTypes\*\*: physical & grid map, create zoning and positioning services with status set to false

5

Run RabbitMQ manually

Generate Gridmap according to the pre-installation phase of positioning system setup

SmartCattle map setup tool on RabbitMQ

Import KML \*\*\* and Gridmap for the new farm or any other new mapType which has been defined in the panel

7

Import a device list for the new farm using “import equipment” or add new devices by selecting available devices and assigning them to the current sub-project

4

Send equipment changes on RabbitMQ

Import or update equipment

Send body temperature and activity data and environment data on RabbitMQ

6

Perform training for zoning and positioning according to the procedure

Only Environmental data is accepted and other cattle-related data is left on the rabbit until assignment of the sensors to the cattle is done at the smart cattle side!

Generate positioning and zoning models and maps

Activate zoning & positioning services by sending DGN list and training data to the relating services

Send services output data + activity +body temperature +environmental sensors data on RabbitMQ

Sensor to cattle assignment\*\*\*\*

Begin reception of data on Rabbit

Positioning system setup diagram



Notes:

\* Config files contain information needed in generating positioning and zoning models. They can be accessed and modified using “config management” menu in IPARS panel.

\*\* Currently we use two map types: physical and grid map. If new maps are needed, (like what we have in InJust) we can define new map type using panel.

\*\*\* KML map is currently provided by Melika and after upload is formatted as a jeojson format.

\*\*\*\* Data of each remain on Rabbit until the sensor is assigned to a cattle

Needs Review:

1. How a project is created and approved?
2. Create sub-project API detail?
3. Where the sub-project should be approved?
4. RabbitMQ detailed structure?
5. Currently RabbitMQ is activated manually, It should run automatically
6. Why we have divided Body temperature and activity of each cattle and put them in different collections?
7. Redis structure?