

بِسْمِ اللَّهِ الرَّحْمَنِ الرَّحِيمِ

داکیومنت ERD

استاد مربوطه: دکتر سجاد پیراهش

ارائه دهنده: زهرا پور عباس

نیم سال اول تحصیلی:

۱۴۰۲-۱۴۰۳

سرفصل

.....شناخت سیستم	۱
.....تکنولوژی ها	۱
.....Django	۱
.....Docker	۱
.....Mysql	۱
.....Markdown	۲
.....Figma	۲
.....draw.io	۲
.....شناخت محیط	۳
..... موجودیت های محیط و توضیحات دیاگرام	۴
..... نوع کاربر	۴
..... کاربر	۴
..... حوزه	۴
..... ساختمان	۴
..... سیستم عامل	۴
..... ورژن سیستم عامل	۴
..... کامپیوتر	۴
..... دسته کالا	۵
..... دسته ویژگی کالا	۵
..... ویژگی های کالا	۵
..... مقادیر پیش فرض ویژگی های کالا	۵
..... کالا	۵
..... کالای تحویلی	۵
..... وضعیت تیکت	۵
..... موضوع در خواستی تیکت	۶
..... تیکت	۶

بروزرسانی.....	۶
جایه جایی.....	۶
اسفاط	۶
تعمیر داخلی.....	۶
سند ارسال به بیرون	۷
نصبیات.....	۷
نرم افزاری.....	۷
پلمر.....	۷
تصویر دیاگرام سیستم.....	۸
جداول دیاگرام.....	۹
کدهای جداول دیاگرام.....	۲۳
User Flow.....	۳۳

شناخت سیستم

تکنولوژی ها

Django

جنگو یک فریم ورک وب سطح بالا پایتون است که توسعه سریع وب سایت های امن و قابل نگهداری را امکان پذیر می کند. جنگو که توسط توسعه دهندگان با تجربه ساخته شده است، از بسیاری از مشکلات توسعه وب مراقبت می کند، بنابراین می توانید بدون نیاز به اختراع مجدد چرخ، روی نوشتن برنامه خود تمرکز کنید و مجموعه ای از ابزارها و کتابخانه ها را برای کمک به شما در ساخت سریع و کارآمد برنامه های وب ارائه می دهد. اگرچه جنگو تا حدودی از توسعه فرانت‌اند پشتیبانی می کند، اما تمرکز اصلی آن بر روی بک اند است. و همچنین یک فریمورک همه کاره است که می تواند برای ایجاد برنامه های کاربردی در حوزه های مختلف استفاده شود. شرکت‌ها در حال حاضر از جنگو برای ایجاد انواع برنامه‌های کاربردی مانند سیستم‌های مدیریت محتوا، سایت‌های شبکه‌های اجتماعی، و پلتفرم‌های محاسباتی علمی و غیره استفاده می کنند. یکی دیگر از عواملی که به موقوفیت مدام جنگو کمک می کند، مقیاس پذیری آن است. جنگو توانایی خود را در مدیریت وب سایت های پریازدید و برنامه های پیچیده ثابت کرده است. معماری این فریم ورک، با جداسازی واضح نگرانی ها و مازولات بودن، توسعه دهندگان را قادر می سازد تا پایگاه های کد مقیاس پذیر و قابل نگهداری بسازند.

Docker

Docker یک پلتفرم نرم افزاری است که به شما امکان می دهد تا برنامه ها را به سرعت بسازید، آزمایش کنید و اجرا کنید. Docker نرم افزار را در واحدهای استانداردی به نام کانتینرها بسته بندی می کند که همه چیزهایی را که نرم افزار برای اجرا نیاز دارد از جمله کتابخانه ها، ابزارهای سیستم، کد و زمان اجرا دارد. به عبارت دیگر Docker توسط توسعه دهندگان استفاده می شود تا به آنها کمک کند استقرار برنامه های کاربردی در داخل کانتینرها را خود کار کنند. یک محیط سازگار توسط داکر ارائه شده است تا نرم افزار بتواند در چندین محیط محاسباتی اجرا شود. ارسال، ساخت و اجرای برنامه ها به دلیل داکر آسان تر می شود و عملیات استقرار را automate کنند. این یک جایگزین مناسب و مقرون به صرفه برای ماشین های مجازی مبتنی بر Hypervisor ارائه می دهد، بنابراین می توانید از ظرفیت سرور خود برای دستیابی به اهداف تجاری خود استفاده کنید. Docker برای محیط های با چگالی بالا و برای استقرارهای کوچک و متوسط که در آن باید کارهای بیشتری با منابع کمتر انجام دهید، عالی است.

Mysql

Mysql یک سیستم مدیریت پایگاه داده رابطه ای، منبع باز، ایده آل برای برنامه های کوچک و بزرگ، یار سریع، قابل اعتماد، مقیاس پذیر و آسان برای استفاده، رایگان، کراس پلتفرم است. MySQL توسط Oracle Corporation توسعه، توزیع و پشتیبانی می شود. Mysql به برنامه نویسان اجازه می دهد تا از SQL برای ایجاد، اصلاح و استخراج داده ها از پایگاه داده رابطه ای استفاده کنند. با عادی سازی داده ها در ردیف ها و ستون های جداول، MySQL به یک سیستم ذخیره سازی داده مقیاس پذیر و در عین حال انعطاف پذیر با یک رابط کاربر پسند تبدیل می شود که می تواند داده های زیادی را مدیریت کند.

Markdown

یک زبان نشانه گذاری است که به کاربران اجازه می دهد در متن ساده بنویسند و سپس آن را به متن فرمت شده مانند HTML تبدیل کنند. برای اهداف مختلفی استفاده می شود، از جمله: مستندات فنی، وبلاگ نویسی و پیام های فوری، نرم افزار مشارکتی، فایل های Readme، یادگیری آسان، قابل حمل و سازگار، انعطاف پذیر و قابل تنظیم، قابل خواندن و نگهداری است. با این حال، Markdown همچنین دارای معایبی است، از جمله: استاندارد نشده و برای توسعه دهندهان عالی نیست. ما در این پژوهه برای نوشتن فایل Readme استفاده کردیم.

Figma

یک ابزار طراحی محبوب است که برای ایجاد، اشتراک گذاری و آزمایش طرح ها برای محصولات دیجیتال استفاده می شود. این یک ابزار مبتنی بر مرورگر است که به کاربران اجازه می دهد بدون در نظر گرفتن پلتفرم یا موقعیت مکانی خود، به صورت بلاذرگ روی یک فایل با هم کار کنند. دلیل برای استفاده مردم از Figma عبارت است از همکاری، مبتنی بر مرورگر، گرافیک برداری، اشتراک گذاری. ما در این پژوهه از آن برای کشیدن user flow استفاده نمودیم.

draw.io

مردم از draw.io که با نام Draw.io با محصولات Atlassian Confluence و Jira ادغام Diagrams.net نیز شناخته می شود. می شود و به کاربران امکان می دهد نمودارهای خود را جاسازی کنند. به صورت آفلاین و روی دستگاه های تلفن همراه کار می کند. می تواند نمودارهای ذخیره شده را در فضای ابری یا در فضای ذخیره سازی شبکه در یک مرکز داده نگهداری کند. همچنین با سرویس های ابری مانند GitHub، GitLab.com، Google Drive، OneDrive، Dropbox، اینفوگرافیک ها، نقشه های ذهنی، نقشه های داستانی، جدول زمانی، فلوچارت ها، وايرفريم ها، نمودارهای UML و نمودارهای سازمانی. ما از این ابزار برای کشیدن ERD diagram استفاده کرده ایم.

شناخت محیط

این سیستم با هدف مدیریت تمام دارایی های یک سازمان نوشته شده است. دارایی هر سازمان اعم از کامپیوتر، مونیتور، کیبورد، پرینتر، قطعات شبکه، قطعات داخلی کامپیوتر و ... است. ما همچنین عملیاتی را مختص به هر یک از نوع دارایی داریم. اسقاط برای تمام کالاهای به غیر از قطعات داخلی را شامل می شود. بروزرسانی و تعمیر داخلی مختص قطعات داخلی است. جابجایی تمام کالاهای به غیر از قطعات داخلی را شامل می شود. نصیبات فقط بر روی کامپیوتر انجام می شود. سند ارسال به بیرون برای مونیتور و پرینتر می باشد.

ما در این سیستم اطلاعات کاربران دارنده هر دارایی در هر حوزه و ساختمن را نگهداری می کنیم. هر کاربر ساده می تواند برای دارایی ها تیکت ثبت نماید و پشتیبان برای دارایی های خود و کاربران زیر دستش. همچنین هر تیکت با موضوع های مختلف امکان ارجاع به کاربر بالا دست را دارد و می تواند وضعیت آن را تغییر دهد.

هر کامپیوتر سیستم عامل های مختلف با ورژن های متنوع را داراست. قبل از تحويل کامپیوتر به کاربر باید پلomp شود. هر دارایی تحويلی ما دارای سریال منحصر بفرد می باشد.

ما در این سیستم می توانیم تمام دارایی ها، تیکت ها، عملیات انجام شده بر روی هر دارایی به همراه خود دارایی اش را مشاهده نماییم.

موجودیت های محیط و توضیحات دیاگرام

نوع کاربر

این موجودیت سطح های دسترسی ما به سیستم را نگهداری می کند، ما سه سطح دسترسی اعم از /ادمین، پشتیبان و کاربر ساده داریم.

کاربر

تمام کاربران ما اعم از /ادمین، پشتیبان و کاربر ساده در سیستم را نگهداری می کند. این موجودیت رابطه های N:1 با موجودیت نوع کاربری، N:1 با موجودیت کاربر تحت عنوان کاربر پشتیبان ، N:N با حوزه و ساختمان را با عنوان محل کار، رابطه N:N را با ساختمان با عنوان طبقات تحت اختیار ساختمان، رابطه N:N را با تیکت با عنوان تیکت های ارجاعی دارد

حوزه

تمام حوزه های موجود در دانشگاه را در سیستم را نگهداری می کند.

ساختمان

تمام ساختمان های موجود در دانشگاه را در سیستم را نگهداری می کند.

سیستم عامل

تمام سیستم عامل های موجود در بازار را در سیستم را نگهداری می کند.

ورژن سیستم عامل

تمام ورژن های مختص به هر یک از سیستم عامل های موجود در بازار را در سیستم را نگهداری می کند. این موجودیت رابطه N:1 با موجودیت سیستم عامل دارد.

کامپیوتر

تمام اطلاعات کامپیوتر های مختلف در خود نگهداری می کند. این موجودیت رابطه های N:1 با موجودیت حوزه با عنوان حوزه دارنده، N:1 با موجودیت ساختمان با عنوان ساختمان دارنده،N:1 با موجودیت کاربر با عنوان متعلق، N:1 با موجودیت کاربر با عنوان کاربر ثبت کننده، N:1 با موجودیت کاربر با عنوان ویرایش کننده، N:N با نصبیات جهت نگهداری اطلاعات مربوط به نصبیات انجام شده بروی آن، N:N با جابجایی جهت نگهداری اطلاعات مربوط به جابجایی انجام شده کامپیوتر، با موجودیت ورژن سیستم عامل یه عنوان سیستم عامل کامپیوتر دارد.

دسته کالا

تمام اطلاعات دسته کالا را در خود نگهداری می کند. این موجودیت رابطه N:N با دسته ویژگی کالا و ویژگی های کالا را با عنوان محل کار را دارد.

دسته ویژگی کالا

تمام اطلاعات دسته ویژگی کالا را در خود نگهداری می کند.

ویژگی های کالا

تمام اطلاعات ویژگی های کالا را در خود نگهداری می کند.

مقادیر پیش فرض ویژگی های کالا

تمام اطلاعات مقادیر پیش فرض ویژگی های کالا را در خود نگهداری می کند. این موجودیت رابطه از نوع ضعیف با ویژگی کالا تحت عنوان مقادیر پیش فرض هر ویژگی های کالا را دارد.

کالا

تمام اطلاعات مقادیر کالا را در خود نگهداری می کند. این موجودیت رابطه های 1:N با موجودیت دسته کالا با عنوان بودن در زیر دسته کالا مشخص، N:N با ویژگی های کالا جهت نگهداری اطلاعات مربوط به ویژگی های کالا را دارد.

کالای تحویلی

تمام اطلاعات دارای ها را بغیر از کامپیوتر را در خود نگهداری می کند. این موجودیت رابطه های 1:N با موجودیت حوزه با عنوان حوزه دارنده، N:N با موجودیت ساختمان با عنوان ساختمان دارنده، N:N با نسبیات جهت نگهداری اطلاعات مربوط به نسبیات انجام شده بروی آن، N:N با جابجایی جهت نگهداری اطلاعات مربوط به جابجایی انجام شده کامپیوتر، N:N با موجودیت کالا تحت عنوان تحویل نمونه کالا با سریال منحصرفرد، N:N با موجودیت بروزرسانی با عنوان کالای تحویلی جایگزین در هر بروزرسانی، N:N با موجودیت کامپیوتر با عنوان تحویل کالای تحویلی داخل کامپیوتر، N:N با موجودیت کاربر با عنوان متعلق در صورت قطعه داخلی نبودن را دارد.

وضعیت تیکت

تمام وضعیت های ممکن در تیکت را در خود نگهداری می کند.

موضوع در خواستی تیکت

تمام موضوع های در خواستی ممکن در تیکت را در خود نگهداری می کند.

تیکت

تمام اطلاعات مربوط به تیکت های ثبت شده برای هر کامپیوتر و کالای تحویلی را در خود نگهداری می کند. ۱:N با موجودیت کاربر با عنوان کاربر ثبت کننده، ۱:N با موجودیت وضعیت تیکت، ، ۱:N با موجودیت موضوع در خواستی تیکت، ۱:N با موجودیت کامپیوتر تحت عنوان کامپیوتر درج شونده در تیکت، ۱:N با موجودیت کالای تحویلی تحت عنوان کالای تحویلی درج شونده در تیکت را دارد.

بروزرسانی

تمام اطلاعات مربوط به دارایی های بروزرسانی های انجام شده را در درون خود نگهداری می کند. این موجودیت رابطه های ۱:N با موجودیت کاربر با عنوان کاربر ثبت کننده، ۱:N با موجودیت کاربر با عنوان ویرایش کننده، ۱:N با موجودیت کاربر با عنوان کاربر دارنده، ۱:۱ با موجودیت تیکت به عنوان در خواست بروزرسانی در تیکت، ۱:N با موجودیت کالای تحویلی تحت عنوان قطعات داخلی تعویض شونده را دارد.

جابه جایی

تمام اطلاعات مربوط به دارایی های جایجا شده را در درون خود نگهداری می کند. این موجودیت رابطه های ۱:N با موجودیت کاربر با عنوان کاربر ثبت کننده، ۱:N با موجودیت کاربر با عنوان ویرایش کننده، ۱:N با موجودیت کاربر با عنوان کاربر انتقال دهنده، ۱:۱ با موجودیت تیکت به عنوان در خواست جابه جایی در تیکت را دارد.

اسقاط

تمام دارایی های اسقاط شده را در درون خود نگهداری می کند. این موجودیت رابطه های ۱:N با موجودیت کاربر ثبت کننده، ۱:N با موجودیت کاربر با عنوان ویرایش کننده، ۱:N با موجودیت کاربر با عنوان کاربر دارنده، ۱:۱ با موجودیت تیکت به عنوان در خواست اسقاط در تیکت، ۱:۱ با موجودیت کامپیوتر با عنوان کامپیوتر اسقاط شونده، ۱:۱ با موجودیت کالای تحویلی با عنوان کالای تحویلی اسقاط شونده را دارد.

تعمیر داخلی

تمام تعمیر های داخلی انجام شده بر روی قطعات داخلی هر کامپیوتر را در خود نگهداری می کند. این موجودیت رابطه های ۱:N با موجودیت کاربر با عنوان کاربر ثبت کننده، ۱:N با موجودیت کاربر با عنوان ویرایش کننده، ، ۱:N با موجودیت تیکت به عنوان در خواست تعمیر داخلی در تیکت، ۱:N با موجودیت کالای تحویلی با عنوان تعمیر های داخلی انجام شده بر روی قطعات داخلی را دارد.

سنده ارسال به بیرون

تمام ارسال های به بیرون انجام شده مختص هر مونیتور و پرینتر را در خود نگهداری می کند. این موجودیت رابطه های N:1 با موجودیت کاربر با عنوان کاربر ثبت کننده، 1:N با موجودیت کاربر با عنوان ویرایش کننده، 1:N با موجودیت کاربر با عنوان کاربر دارنده، 1:N با موجودیت تیکت به عنوان در خواست ارسال به بیرون در تیکت، 1:N با موجودیت کالای تحويلی با عنوان کالای تحويلی ارسال شونده به بیرون را دارد.

نصبیات

تمام نصبیات انجام شده بر روی همه کامپیوتر ها را درون خود نگهداری می کند. این موجودیت رابطه های N:N با موجودیت نرم افزار را با عنوان نرم افزار های نصب شده در هر نصبیات با درج کارایی یا همان علت نصب، 1:1 با موجودیت تیکت به عنوان در خواست نصب در تیکت، 1:N با موجودیت کاربر با عنوان کاربر ثبت کننده، 1:N با موجودیت کاربر با عنوان ویرایش کننده را دارد.

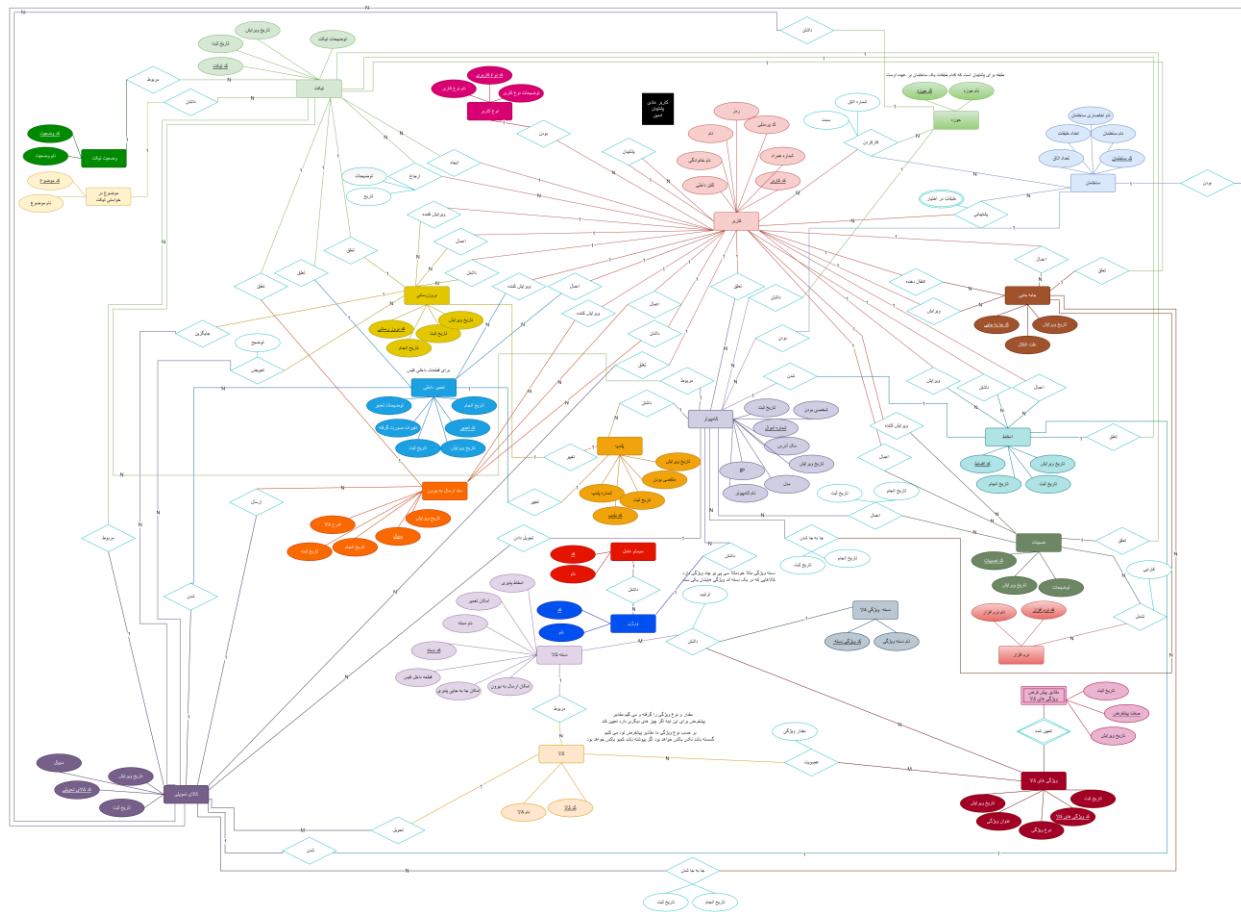
نرم افزاری

نگهدارنده لیستی از نرم افزار های موجود است که در هر نصبیات حداقل یکی از آنها نصب می شود.

پلمنپ

اطلاعات مربوط به پلمنپ های کامپیوتر را در خود نگهداری می کند که بغیر از یکی از آنها بقیه منقضی شده اند. این موجودیت رابطه 1:N با موجودیت کامپیوتر با عنوان کامپیوتر دارنده، 1:1 با موجودیت بروزرسانی برای ثبت اینکه در بروزرسانی تغییر کرده، 1:1 با موجودیت بروزرسانی برای ثبت اینکه در تعمیر داخلی تغییر کرده را دارد.

تصویر دیاگرام سیستم



جداول دیاگرام

نوع کاربری						
Row	Attribute name	Data Type	PK	Null	Unique	FK
1	کد نوع کاربری	INT	✓		✓	
2	عنوان نوع کاربری	NVARCHAR(255)				
3	توضیحات نوع کاربری	TEXT				
4	تاریخ ایجاد	DATETIME				
5	تاریخ ویرایش	DATETIME				

کاربر						
Row	Attribute name	Data Type	PK	Null	Unique	FK
1	کد کاربر	INT	✓		✓	
2	نام	NVARCHAR(255)				
3	نام خانوادگی	NVARCHAR(255)				
4	شماره همراه	NVARCHAR(11)		✓		
5	تلفن داخلی	NVARCHAR(11)				
6	کد کاربر ثبت کننده	INT		✓		✓
7	کد کاربر ویرایش کننده	INT		✓		✓
8	تاریخ ایجاد	DATETIME				
9	تاریخ ویرایش	DATETIME				
10	کد پرسنلی	INT			✓	
11	رمز	NVARCHAR(255)				
12	کد نوع کاربری	INT				✓
13	کد کاربر پشتیبان	INT				✓

حوزه						
Row	Attribute name	Data Type	PK	Null	Unique	FK
1	کحوزه	INT	✓		✓	
2	نام حوزه	NVARCHAR(255)				
3	کد کاربر ثبت کننده	INT				✓
4	کد کاربر ویرایش کننده	INT		✓		✓
5	تاریخ ایجاد	DATETIME				
6	تاریخ ویرایش	DATETIME				

ساختمان						
Row	Attribute name	Data Type	PK	Null	Unique	FK
1	کد ساختمان	INT	✓		✓	
2	نام ساختمان	NVARCHAR(255)				
3	نام اختصاری ساختمان	NVARCHAR(255)				
4	تعداد طبقات	INT				
5	کد کاربر ثبت کننده	INT				✓
6	کد کاربر ویرایش کننده	INT				✓
7	تاریخ ایجاد	DATETIME				
8	تاریخ ویرایش	DATETIME				
9	تعداد آتاق	INT				

طبقات در اختیار پشتیبان ساختمان						
Row	Attribute name	Data Type	PK	Null	Unique	FK
1	کد کاربری	INT				✓
2	کد ساختمان	INT	✓		✓	✓
3	طبقه در اختیار	INT				

کریبران کار کننده در هر حوزه ساختمان						
Row	Attribute name	Data Type	PK	Null	Unique	FK
1	کد کاربر	INT	✓		✓	✓
2	کد ساختمان	INT				
3	کد حوزه	INT				
4	سمت	NVARCHAR(255)				
5	شماره اتاق	INT				

سیستم عامل						
Row	Attribute name	Data Type	PK	Null	Unique	FK
1	کد سیستم عامل	INT	✓		✓	
2	نام	NVARCHAR(255)				
3	کد کاربر ثبت کننده	INT				✓
4	کد کاربر ویرایش کننده	INT		✓		✓
5	تاریخ ویرایش	DATETIME				
6	تاریخ ثبت	DATETIME				

ورژن سیستم عامل						
Row	Attribute name	Data Type	PK	Null	Unique	FK
1	کد ورژن سیستم عامل	INT	✓		✓	
2	نام ورژن سیستم عامل	NVARCHAR(255)				
3	کد کاربر ثبت کننده	INT				✓
4	کد کاربر ویرایش کننده	INT		✓		✓
5	تاریخ ثبت	DATETIME				
6	تاریخ ویرایش	DATETIME				
7	کد سیستم عامل	INT				✓

کامپیوٹر						
Row	Attribute name	Data Type	PK	Null	Unique	FK
1	شماره اموال کامپیوٹر	INT	✓		✓	
2	نام کامپیوٹر	NVARCHAR(255)				
3	تاریخ ثبت	DATETIME				
4	مدل	NVARCHAR(255)				
5	مک آدرس	TEXT				
6	IP	NVARCHAR			✓	
7	کد کاربر متعلق	INT			✓	
8	کد ورژن سیستم عامل	INT			✓	
9	شخصی بودن	BOOLEAN				
10	تاریخ ویرایش	DATETIME				
11	کد کاربر ویرایش کننده	INT	✓		✓	
12	کد کاربر ثبت کننده	INT			✓	
13	کد کاربر حوزه	INT			✓	
14	کد کاربر ساختمان	INT				✓

دسته کالا							
Row	Attribute name	Data Type	PK	Null	Unique	FK	
1	کد دسته	INT	✓		✓		
2	نام دسته	NVARCHAR(255)					
3	قطعه داخل کیس	BOOLEAN					
4	امکان ارسال به بیرون	BOOLEAN					
5	اسقاط پذیری	BOOLEAN					
6	امکان جایی پذیری	BOOLEAN					
7	امکان تعمیر	BOOLEAN					
8	کد کاربر ثبت کننده	INT				✓	
9	کد کاربر ویرایش کننده	INT		✓		✓	
10	تاریخ ویرایش	DATETIME					
11	تاریخ ثبت	DATETIME					

دسته ویژگی کالا							
Row	Attribute name	Data Type	PK	Null	Unique	FK	
1	کد ویژگی دسته	INT	✓		✓		
2	نام دسته ویژگی	NVARCHAR(255)					
3	کد کاربر ویرایش کننده	INT		✓		✓	
4	کد کاربر ثبت کننده	INT				✓	
5	تاریخ ویرایش	DATETIME					
6	تاریخ ثبت	DATETIME					

ویژگی های کالا						
Row	Attribute name	Data Type	PK	Null	Unique	FK
1	کد ویژگی های کالا	INT	✓		✓	
2	عنوان ویژگی	NVARCHAR(255)				
3	نوع ویژگی	INT				
4	کد کاربر ویرایش کننده	INT		✓		✓
5	کد کاربر ثبت کننده	INT				✓
6	تاریخ ویرایش	DATETIME				
7	تاریخ ثبت	DATETIME				

دسته کالای مرتبه به ویژگی های کالای عضو دسته ویژگی کالا						
Row	Attribute name	Data Type	PK	Null	Unique	FK
1	کد دسته کالا	INT				✓
2	کد ویژگی کالا	INT	✓		✓	✓
3	کد دسته ویژگی	INT				✓
4	ترتیب	INT				

مقدار پیش فرض ویژگی های کالا						
Row	Attribute name	Data Type	PK	Null	Unique	FK
1	کد ویژگی کالا	INT				✓
2	صفت پیشفرض	NVARCHAR(255)	✓		✓	
3	کد کاربر ویرایش کننده	INT		✓		✓
4	کد کاربر ثبت کننده	INT				✓
5	تاریخ ویرایش	DATETIME				
6	تاریخ ثبت	DATETIME				

کالا						
Row	Attribute name	Data Type	PK	Null	Unique	FK
1	کد کالا	INT	✓		✓	
2	نام کالا	NVARCHAR(255)				
3	کد دسته کالا	INT				✓
4	کد کاربر ویرایش کننده	INT		✓		✓
5	کد کاربر ثبت کننده	INT				✓
6	تاریخ ویرایش	DATETIME				
7	تاریخ ثبت	DATETIME				

اختصاص ویژگی به کالا (عضویت کالا در ویژگی های کالا)						
Row	Attribute name	Data Type	PK	Null	Unique	FK
1	کد کالا	INT				✓
2	کد ویژگی کالا	INT	✓		✓	✓
3	مقدار ویژگی	NVARCHAR(255)				

کالای تحویلی						
Row	Attribute name	Data Type	PK	Null	Unique	FK
1	کد کالای تحویلی	INT	✓		✓	
2	تاریخ ثبت	DATETIME				
3	کد کاربر متعلق	INT		✓		✓
4	کد کامپیوتر تحویلی	INT		✓		✓
5	کد بروزرسانی	INT		✓		✓
6	تاریخ ویرایش	DATETIME				
7	کد کاربر ویرایش کننده	INT		✓		✓
8	کد کاربر ثبت کننده	INT				✓
9	کد ساختمان	INT		✓		✓
10	کد حوزه	INT		✓		✓
11	سریال	NVARCHAR(255)			✓	
12	کد کالا	INT				✓

وضعیت تیکت						
Row	Attribute name	Data Type	PK	Null	Unique	FK
1	کد وضعیت	INT	✓		✓	
2	نام وضعیت	NVARCHAR(255)				
3	تاریخ ایجاد	DATETIME				
4	تاریخ ویرایش	DATETIME				

موضوع در خواستی تیکت						
Row	Attribute name	Data Type	PK	Null	Unique	FK
1	کد موضوع	INT	✓		✓	
2	نام موضوع	NVARCHAR(255)				
3	تاریخ ایجاد	DATETIME				
4	تاریخ ویرایش	DATETIME				

تیکت						
Row	Attribute name	Data Type	PK	Null	Unique	FK
1	کد تیکت	BIGINT	✓		✓	
2	کد وضعیت	INT				✓
3	کد موضوع در خواستی	INT				✓
4	کد کاربر ثبت کننده	INT				✓
5	کد کالای تحویلی مربوطه	INT		✓		✓
6	کد کامپیووتر مربوطه	INT		✓		✓
7	توضیح تیکت	TEXT				
8	تاریخ ایجاد	DATETIME				
9	تاریخ ویرایش	DATETIME				

تیکت ارجاعی کاربران						
Row	Attribute name	Data Type	PK	Null	Unique	FK
1	کد کاربر	INT	✓		✓	✓
2	کد تیکت	INT				✓
3	تاریخ	DATETIME				
4	توضیحات	TEXT				

بروزرسانی						
Row	Attribute name	Data Type	PK	Null	Unique	FK
1	کد بروز رسانی	INT	✓		✓	
2	تاریخ ثبت	DATETIMETIME				
3	کد کاربر دارنده بروزرسانی	INT			✓	
4	کد کاربر ثبت کننده	INT			✓	
5	کد تیکت مرتبط	INT			✓	
6	تاریخ انجام	DATETIMETIME				
7	کد کاربر ویرایش کننده	INT		✓		✓
8	تاریخ ویرایش	DATETIMETIME				

کالای تحويلی تعویضی در بروزرسانی						
Row	Attribute name	Data Type	PK	Null	Unique	FK
1	کد بروزرسانی	INT			✓	
2	کد کالای تحويلی	INT	✓		✓	✓
3	توضیحات	TEXT				

جا به جایی						
Row	Attribute name	Data Type	PK	Null	Unique	FK
1	کد جا به جایی	INT	✓		✓	
2	عنوان انتقال	NVARCHAR(255)				
3	کد کاربر ثبت کننده	INT				✓
4	کد کاربر ویرایش کننده	INT		✓		✓
5	تاریخ ویرایش	DATETIMETIME				
6	کاربر متعلق قبلی	INT				✓
7	کد تیکت مرتبط	INT		✓		✓

جایه جایی کالاهای تحویلی						
Row	Attribute name	Data Type	PK	Null	Unique	FK
1	کد کالای تحویلی	INT	✓		✓	✓
2	کد جایه جایی	INT				✓
3	از تاریخ	DATETIME				
4	تا تاریخ	DATETIME		✓		

جایه جایی کالاهای کامپیووتر						
Row	Attribute name	Data Type	PK	Null	Unique	FK
1	کد کامپیووتر	INT	✓		✓	✓
2	کد جایه جایی	INT				✓
3	تاریخ ایجاد	DATETIME				
4	تاریخ ثبت	DATETIME				

اسقاط						
Row	Attribute name	Data Type	PK	Null	Unique	FK
1	کد اسقاط	INT	✓		✓	
2	تاریخ ثبت	DATETIME				
3	کد کاربر متعلق	INT				✓
4	کاربر ثبت کننده	INT				✓
5	کد تیکت مرتبط	INT				✓
6	کد کالای تحویلی مربوطه	INT	✓		✓	
7	کد کامپیووتر مربوطه	INT	✓		✓	
8	تاریخ انجام	DATETIME				
9	تاریخ ویرایش	DATETIME				
10	کد کاربر ویرایش کننده	INT	✓		✓	

تعمیر داخلی						
Row	Attribute name	Data Type	PK	Null	Unique	FK
1	کد تعمیر	INT	✓		✓	
2	توضیحات تعمیر	TEXT				
3	تغیرات صورت گرفته	TEXT				
4	تاریخ ثبت	DATETIME				
5	تاریخ انجام	DATETIME				
6	کد تیکت مربوطه	INT			✓	
7	کد کاربر ثبت کننده	INT			✓	
8	کد کالای تحویلی	INT			✓	
9	کد کاربر ویرایش کننده	INT		✓		✓
10	تاریخ ویرایش	DATETIME				

سند ارسال به بیرون						
Row	Attribute name	Data Type	PK	Null	Unique	FK
1	سریال سند	INT	✓		✓	
2	شرح کالا	NVARCHAR				
3	کد کالای تحویلی	INT			✓	
4	کد کاربر ثبت کننده	INT			✓	
5	کاربر دارنده سند ارسال بیرون	INT			✓	
6	کد تیکت مربوطه	INT			✓	
7	تاریخ ثبت	DATETIMETIME				
8	تاریخ انجام	DATETIMETIME				
9	تاریخ ویرایش	DATETIMETIME				
10	کد کاربر ویرایش کننده	INT		✓		✓

عملیات نصب						
Row	Attribute name	Data Type	PK	Null	Unique	FK
1	کد نصبیات	INT	✓		✓	
2	توضیحات	TEXT				
3	کد نیکت مرتبط	INT				✓
4	کاربر ثبت کننده	INT				✓
5	تاریخ ویرایش	DATETIME				
6	کد کاربر ویرایش کننده	INT		✓		✓

نرم افزار						
Row	Attribute name	Data Type	PK	Null	Unique	FK
1	کد نرم افزار	INT	✓		✓	
2	نام نرم افزار	NVARCHAR(255)				
3	کاربر ثبت کننده	INT				✓
4	تاریخ ویرایش	DATETIME				
5	تاریخ ثبت	DATETIME				
6	کد کاربر ویرایش کننده	INT		✓		✓

نرم افزار مشمول در هر عملیات نصب						
Row	Attribute name	Data Type	PK	Null	Unique	FK
1	کد نصبیات	INT				✓
2	کد نرم افزار	INT	✓		✓	✓
3	کارایی	TEXT				

عملیات نصب اعمالی در کامپیوتر							
Row	Attribute name	Data Type	PK	Null	Unique	FK	
1	کد نصبیات	INT	✓		✓		
2	کد کامپیوتر	INT				✓	
3	تاریخ انجام	DATETIME					
4	تاریخ ثبت	DATETIMETIME					

پلیس							
Row	Attribute name	Data Type	PK	Null	Unique	FK	
1	کد پلیس	INT	✓		✓		
2	شماره پلیس	NVARCHAR(7)			✓		
3	تاریخ ثبت	DATETIME					
4	منقضی بودن	BOOLEAN					
5	کد کامپیوتر	INT				✓	
6	کد بروز رسانی	INT	✓		✓		
7	کد کاربر ویرایش کننده	INT	✓		✓		
8	کد تعمیر داخلی	INT	✓		✓		
9	تاریخ ویرایش	DATETIME					
10	کد کاربر ثبت کننده	INT				✓	

کدهای جداول دیاگرام

```
CREATE TABLE IF NOT EXISTS `userrole`(
    `UserRoleID` int NOT NULL AUTO_INCREMENT,
    `UserName` varchar (255) NOT NULL,
    `UserRoleDescription` text NOT NULL,
    `UserRoleCreateTime` timestamp NOT NULL DEFAULT CURRENT_TIMESTAMP,
    `UserRoleUpdateTime` timestamp NOT NULL DEFAULT CURRENT_TIMESTAMP ON UPDATE CURRENT_TIMESTAMP,
    UNIQUE KEY `UserRoleID`(`UserRoleID`);
```

```
INSERT INTO userrole(userrolename,userroledescription)
values
    ('admin','access every thing'),
    ('supporter','access every thing inside the regin'),
    ('usual user','access just information about own');
```

```
INSERT INTO userrole(userrolename,userroledescription)
values
    ('admin','access every thing'),
    ('supporter','access every thing inside the regin'),
    ('usual user','access just information about own');

CREATE TABLE IF NOT EXISTS `user`(
    `UserId` int NOT NULL AUTO_INCREMENT,
    `UserPersonalId` int NOT NULL,
    `UserName` varchar (255) NOT NULL,
    `UserLastName` varchar (255) NOT NULL,
    `UserPhoneNumber` varchar (11) DEFAULT NULL,
    `UserLandLinePhoneNumber` varchar(11) DEFAULT NULL,
    `CreatorUserId` int DEFAULT NULL,
    `UpdaterUserId` int DEFAULT NULL,
    `UserSupportId` int NULL,
    `UserCreateTime` timestamp NOT NULL DEFAULT CURRENT_TIMESTAMP,
    `UserUpdateTime` timestamp NOT NULL DEFAULT CURRENT_TIMESTAMP ON UPDATE CURRENT_TIMESTAMP,
    `UserPassword` varchar(128) NOT NULL,
    `UserRoleID` int NOT NULL,
    `last_login` datetime(6) DEFAULT NULL,
    `is_active` tinyint(1) DEFAULT '1',
    `is_anonymous` tinyint(1) DEFAULT '1',
    `is_authenticated` tinyint(1) DEFAULT '0',
    PRIMARY KEY(`UserId`),
    UNIQUE KEY `UserId`(`UserId`),
    UNIQUE KEY `UserPersonalId`(`UserPersonalId`),
    KEY `CreatorUserId`(`CreatorUserId`),
    KEY `UpdaterUserId`(`UpdaterUserId`),
    KEY `UserRoleID`(`UserRoleID`),
    CONSTRAINT `user_ibfk_1` FOREIGN KEY(`CreatorUserId`) REFERENCES `user`(`UserId`) ON UPDATE CASCADE,
    CONSTRAINT `user_ibfk_2` FOREIGN KEY(`UpdaterUserId`) REFERENCES `user`(`UserId`) ON UPDATE CASCADE,
    CONSTRAINT `user_ibfk_3` FOREIGN KEY (`UserSupportId`) REFERENCES `user`(`UserId`) ON UPDATE CASCADE,
    CONSTRAINT `UserRoleID` FOREIGN KEY(`UserRoleID`) REFERENCES `userrole`(`UserRoleID`)ON UPDATE CASCADE
);

INSERT INTO
user(userpersonalid,username,lastname,phonenum,landlinephonenum,passwd,roleid,is_active,
is_anonymous,is_authenticated)
VALUES
-- (1,'admin','admin','09021780622','0416666605','admin123',1,True,True,False);
-- (1,'admin','admin','09021780622','0416666605','pbkdf2_sha256$72000$TeBWIwDa4dnsBuIFifCcq$uPfUyR0fDsgUYCbv@mHefij
GnaL2fcuVANyjsgtBE=',1,True,True,False);
```

```
CREATE TABLE IF NOT EXISTS `area` (
    `AreaId` int NOT NULL AUTO_INCREMENT,
    `AreaName` varchar(255) NOT NULL,
    `CreatorUserId` int NOT NULL,
    `UpdaterUserId` int DEFAULT NULL,
    `AreaCreateTime` timestamp NOT NULL DEFAULT CURRENT_TIMESTAMP,
    `AreaUpdateTime` timestamp NOT NULL DEFAULT CURRENT_TIMESTAMP ON UPDATE CURRENT_TIMESTAMP,
    PRIMARY KEY (`AreaId`),
    UNIQUE KEY `AreaId`(`AreaId`),
    KEY `CreatorUserId`(`CreatorUserId`),
    KEY `UpdaterUserId`(`UpdaterUserId`),
    CONSTRAINT `area_ibfk_1` FOREIGN KEY (`CreatorUserId`) REFERENCES `user`(`UserId`) ON UPDATE CASCADE,
    CONSTRAINT `area_ibfk_2` FOREIGN KEY (`UpdaterUserId`) REFERENCES `user`(`UserId`) ON UPDATE CASCADE
);
```

```
CREATE TABLE IF NOT EXISTS `building` (
    `BuildingId` int NOT NULL AUTO_INCREMENT,
    `BuildingName` varchar(255) NOT NULL,
    `BuildingAbbreviationName` varchar(255) NOT NULL,
    `CreatorUserId` int NOT NULL,
    `UpdaterUserId` int DEFAULT NULL,
    `BuildingCreateTime` timestamp NOT NULL DEFAULT CURRENT_TIMESTAMP,
    `BuildingUpdateTime` timestamp NOT NULL DEFAULT CURRENT_TIMESTAMP ON UPDATE CURRENT_TIMESTAMP,
    `BuildingFloorCount` int NOT NULL,
    `BuildingRoomCount` int NOT NULL,
    PRIMARY KEY (`BuildingId`),
    UNIQUE KEY `BuildingId`(`BuildingId`),
    KEY `CreatorUserId`(`CreatorUserId`),
    KEY `UpdaterUserId`(`UpdaterUserId`),
    CONSTRAINT `building_ibfk_1` FOREIGN KEY (`CreatorUserId`) REFERENCES `user` (`UserId`) ON UPDATE CASCADE,
    CONSTRAINT `building_ibfk_2` FOREIGN KEY (`UpdaterUserId`) REFERENCES `user` (`UserId`) ON UPDATE CASCADE
);$#SCADE
```

```
CREATE TABLE IF NOT EXISTS `availablefloortousersupport` (
    `BuildingId` int NOT NULL,
    `UserSupportId` int NOT NULL,
    `AvailableFloor` int NOT NULL,
    PRIMARY KEY (`UserSupportId`, `BuildingId`, `AvailableFloor`),
    KEY `BuildingId`(`BuildingId`),
    CONSTRAINT `availablefloortousersupport_ibfk_1` FOREIGN KEY (`UserSupportId`) REFERENCES `user` (`UserId`) ON UPDATE CASCADE,
    CONSTRAINT `availablefloortousersupport_ibfk_2` FOREIGN KEY (`BuildingId`) REFERENCES `building` (`BuildingId`) ON UPDATE CASCADE
);
```

```
CREATE TABLE IF NOT EXISTS `userlocationinbuildingarea` (
    `BuildingId` int NOT NULL,
    `UserId` int NOT NULL,
    `AreaId` int NOT NULL,
    `UserOfficial` varchar(255) NOT NULL,
    `RoomNumber` int NOT NULL,
    PRIMARY KEY (`UserId`, `BuildingId`, `AreaId`),
    KEY `BuildingId`(`BuildingId`),
    KEY `AreaId`(`AreaId`),
    CONSTRAINT `userlocationinbuildingarea_ibfk_1` FOREIGN KEY (`UserId`) REFERENCES `user` (`UserId`) ON UPDATE CASCADE,
    CONSTRAINT `userlocationinbuildingarea_ibfk_2` FOREIGN KEY (`BuildingId`) REFERENCES `building` (`BuildingId`) ON UPDATE CASCADE,
    CONSTRAINT `userlocationinbuildingarea_ibfk_3` FOREIGN KEY (`AreaId`) REFERENCES `area` (`AreaId`) ON UPDATE CASCADE
);
```

```
CREATE TABLE IF NOT EXISTS `operationsystem` (
    `OperationsystemId` int NOT NULL AUTO_INCREMENT,
    `OperationsystemName` varchar(255) NOT NULL,
    `CreatorUserId` int NOT NULL,
    `UpdaterUserId` int DEFAULT NULL,
    `OperationsystemCreateTime` timestamp NOT NULL DEFAULT CURRENT_TIMESTAMP,
    `OperationsystemUpdateTime` timestamp NOT NULL DEFAULT CURRENT_TIMESTAMP ON UPDATE CURRENT_TIMESTAMP,
    PRIMARY KEY (`OperationsystemId`),
    UNIQUE KEY `OperationsystemId`(`OperationsystemId`),
    KEY `operationsystem_ibfk_1`(`CreatorUserId`),
    KEY `operationsystem_ibfk_2`(`UpdaterUserId`),
    CONSTRAINT `operationsystem_ibfk_1` FOREIGN KEY (`CreatorUserId`) REFERENCES `user` (`UserId`) ON UPDATE CASCADE,
    CONSTRAINT `operationsystem_ibfk_2` FOREIGN KEY (`UpdaterUserId`) REFERENCES `user` (`UserId`) ON UPDATE CASCADE
);
```

```
CREATE TABLE IF NOT EXISTS `operationsystemversion` (
    `OperationSystemVersionId` int NOT NULL AUTO_INCREMENT,
    `OperationSystemVersionName` varchar(255) NOT NULL,
    `CreatorUserId` int NOT NULL,
    `UpdaterUserId` int DEFAULT NULL,
    `OperationSystemId` int NOT NULL,
    `OperationSystemVersionCreateTime` timestamp NOT NULL DEFAULT CURRENT_TIMESTAMP,
    `OperationSystemVersionUpdateTime` timestamp NOT NULL DEFAULT CURRENT_TIMESTAMP ON UPDATE CURRENT_TIMESTAMP,
    PRIMARY KEY (`OperationSystemVersionId`),
    UNIQUE KEY `OperationSystemVersionId` (`OperationSystemVersionId`),
    KEY `operationsystemversion_ibfk_1` (`CreatorUserId`),
    KEY `operationsystemversion_ibfk_2` (`UpdaterUserId`),
    KEY `operationsystemversion_ibfk_3` (`OperationSystemId`),
    CONSTRAINT `operationsystemversion_ibfk_1` FOREIGN KEY (`CreatorUserId`) REFERENCES `user` (`UserId`) ON UPDATE CASCADE,
    CONSTRAINT `operationsystemversion_ibfk_2` FOREIGN KEY (`UpdaterUserId`) REFERENCES `user` (`UserId`) ON UPDATE CASCADE,
    CONSTRAINT `operationsystemversion_ibfk_3` FOREIGN KEY (`OperationSystemId`) REFERENCES `operationsystem` (`OperationSystemId`) ON UPDATE CASCADE
);
```

```
CREATE TABLE IF NOT EXISTS `computer` (
    `ComputerPropertyNumber` int NOT NULL,
    `ComputerName` varchar(255) NULL,
    `ComputerModel` varchar(255) NOT NULL,
    `ComputerIP` varchar(15) NOT NULL,
    `ComputerMacAddress` text NOT NULL,
    `ComputerIsPersonal` tinyint(1) NOT NULL,
    `ComputerCreateTime` timestamp NOT NULL DEFAULT CURRENT_TIMESTAMP,
    `ComputerUpdateTime` timestamp NOT NULL DEFAULT CURRENT_TIMESTAMP ON UPDATE CURRENT_TIMESTAMP,
    `CreatorUserId` int NOT NULL,
    `UpdaterUserId` int DEFAULT NULL,
    `OwnerUserId` int DEFAULT NULL,
    `OperationSystemVersionId` int NOT NULL,
    `AreaId` int DEFAULT NULL,
    `BuildingId` int DEFAULT NULL,
    PRIMARY KEY (`ComputerPropertyNumber`),
    UNIQUE KEY `ComputerIP` (`ComputerIP`),
    UNIQUE KEY `ComputerPropertyNumber` (`ComputerPropertyNumber`),
    KEY `computer_ibfk_1` (`CreatorUserId`),
    KEY `computer_ibfk_2` (`UpdaterUserId`),
    KEY `computer_ibfk_3` (`OwnerUserId`),
    KEY `computer_ibfk_4` (`OperationSystemVersionId`),
    KEY `computer_ibfk_5` (`AreaId`),
    KEY `computer_ibfk_6` (`BuildingId`),
    CONSTRAINT `computer_ibfk_1` FOREIGN KEY (`CreatorUserId`) REFERENCES `user` (`UserId`) ON UPDATE CASCADE,
    CONSTRAINT `computer_ibfk_2` FOREIGN KEY (`UpdaterUserId`) REFERENCES `user` (`UserId`) ON UPDATE CASCADE,
    CONSTRAINT `computer_ibfk_3` FOREIGN KEY (`OwnerUserId`) REFERENCES `user` (`UserId`) ON UPDATE CASCADE,
    CONSTRAINT `computer_ibfk_4` FOREIGN KEY (`OperationSystemVersionId`) REFERENCES `operationsystemversion` (`OperationSystemVersionId`) ON UPDATE CASCADE,
    CONSTRAINT `computer_ibfk_5` FOREIGN KEY (`AreaId`) REFERENCES `area` (`AreaId`) ON UPDATE CASCADE,
    CONSTRAINT `computer_ibfk_6` FOREIGN KEY (`BuildingId`) REFERENCES `building` (`BuildingId`) ON UPDATE CASCADE
);
```

```
CREATE TABLE IF NOT EXISTS `goodsgroup` (
    `GoodsGroupId` int NOT NULL AUTO_INCREMENT,
    `GoodsGroupName` varchar(255) NOT NULL,
    `IsPartInsideComputer` tinyint(1) NOT NULL,
    `IsAllowedToSendOut` tinyint(1) NOT NULL,
    `IsAllowedToBeAborted` tinyint(1) NOT NULL,
    `IsAllowedToMove` tinyint(1) NOT NULL,
    `IsPossibleForRepair` tinyint(1) NOT NULL,
    `GoodsGroupCreateTime` timestamp NOT NULL DEFAULT CURRENT_TIMESTAMP,
    `GoodsGroupUpdateTime` timestamp NOT NULL DEFAULT CURRENT_TIMESTAMP ON UPDATE CURRENT_TIMESTAMP,
    `CreatorUserId` int NOT NULL,
    `UpdaterUserId` int DEFAULT NULL,
    PRIMARY KEY (`GoodsGroupId`),
    UNIQUE KEY `GoodsGroupId` (`GoodsGroupId`),
    KEY `goodsgroup_ibfk_1` (`CreatorUserId`),
    KEY `goodsgroup_ibfk_2` (`UpdaterUserId`),
    CONSTRAINT `goodsgroup_ibfk_1` FOREIGN KEY (`CreatorUserId`) REFERENCES `user` (`UserId`) ON UPDATE CASCADE,
    CONSTRAINT `goodsgroup_ibfk_2` FOREIGN KEY (`UpdaterUserId`) REFERENCES `user` (`UserId`) ON UPDATE CASCADE
);
```

```
CREATE TABLE IF NOT EXISTS `attributeCategory` (
    `AttributeCategoryId` int NOT NULL AUTO_INCREMENT,
    `AttributeName` varchar(255) NOT NULL,
    `AttributeCategoryCreateTime` timestamp NOT NULL DEFAULT CURRENT_TIMESTAMP,
    `AttributeCategoryUpdateTime` timestamp NOT NULL DEFAULT CURRENT_TIMESTAMP ON UPDATE CURRENT_TIMESTAMP,
    `CreatorUserId` int NOT NULL,
    `UpdaterUserId` int DEFAULT NULL,
    PRIMARY KEY (`AttributeCategoryId`),
    UNIQUE KEY `AttributeCategoryId` (`AttributeCategoryId`),
    KEY `attributeCategory_ibfk_1` (`CreatorUserId`),
    KEY `attributeCategory_ibfk_2` (`UpdaterUserId`),
    CONSTRAINT `attributeCategory_ibfk_1` FOREIGN KEY (`CreatorUserId`) REFERENCES `user` (`UserId`) ON UPDATE CASCADE,
    CONSTRAINT `attributeCategory_ibfk_2` FOREIGN KEY (`UpdaterUserId`) REFERENCES `user` (`UserId`) ON UPDATE CASCADE
);
```

```
CREATE TABLE IF NOT EXISTS `goodsattributes` (
    `GoodsAttributesId` int NOT NULL AUTO_INCREMENT,
    `GoodsAttributesTitle` varchar(255) NOT NULL,
    `GoodsAttributesType` int NOT NULL,
    `GoodsAttributesCreateTime` timestamp NOT NULL DEFAULT CURRENT_TIMESTAMP,
    `GoodsAttributesUpdateTime` timestamp NOT NULL DEFAULT CURRENT_TIMESTAMP ON UPDATE CURRENT_TIMESTAMP,
    `CreatorUserId` int NOT NULL,
    `UpdaterUserId` int DEFAULT NULL,
    PRIMARY KEY (`GoodsAttributesId`),
    UNIQUE KEY `GoodsAttributesId` (`GoodsAttributesId`),
    KEY `goodsattributes_ibfk_1` (`CreatorUserId`),
    KEY `goodsattributes_ibfk_2` (`UpdaterUserId`),
    CONSTRAINT `goodsattributes_ibfk_1` FOREIGN KEY (`CreatorUserId`) REFERENCES `user` (`UserId`) ON UPDATE CASCADE,
    CONSTRAINT `goodsattributes_ibfk_2` FOREIGN KEY (`UpdaterUserId`) REFERENCES `user` (`UserId`) ON UPDATE CASCADE,
    CONSTRAINT `goodsattributes_chk_1` CHECK (((`GoodsAttributesType` = 1) OR (`GoodsAttributesType` = 2)))
);
```

```
CREATE TABLE IF NOT EXISTS `goodsgroup_attributecategory_goodsattributes_order` (
    `GoodsAttributesId` int NOT NULL,
    `AttributeCategoryId` int NOT NULL,
    `GoodsGroupId` int NOT NULL,
    `Order` int NOT NULL,
    PRIMARY KEY (`GoodsAttributesId`, `GoodsGroupId`),
    UNIQUE KEY `order` (`Order`),
    KEY `fk_attributecategory_order` (`AttributeCategoryId`),
    KEY `fk_goodsgroup_order` (`GoodsGroupId`),
    CONSTRAINT `fk_attributecategory_order` FOREIGN KEY (`AttributeCategoryId`) REFERENCES `attributeCategory` (`AttributeCategoryId`) ON UPDATE CASCADE,
    CONSTRAINT `fk_goodsattributes_order` FOREIGN KEY (`GoodsAttributesId`) REFERENCES `goodsattributes` (`GoodsAttributesId`) ON UPDATE CASCADE,
    CONSTRAINT `fk_goodsgroup_order` FOREIGN KEY (`GoodsGroupId`) REFERENCES `goodsgroup` (`GoodsGroupId`) ON UPDATE CASCADE
);
```

```
CREATE TABLE IF NOT EXISTS `goodsattributesdefaultValue` (
    `GoodsAttributesId` int NOT NULL,
    `DefaultAttributes` varchar(255) CHARACTER SET utf8mb3 COLLATE utf8mb3_general_ci NOT NULL,
    `DefaultAttributesCreateTime` timestamp NOT NULL DEFAULT CURRENT_TIMESTAMP,
    `DefaultAttributesUpdateTime` timestamp NOT NULL DEFAULT CURRENT_TIMESTAMP ON UPDATE CURRENT_TIMESTAMP,
    `CreatorUserId` int DEFAULT NULL,
    `UpdaterUserId` int DEFAULT NULL,
    PRIMARY KEY (`GoodsAttributesId`, `DefaultAttributes`),
    KEY `goodsattributesdefaultValue_ibfk_1` (`CreatorUserId`),
    KEY `goodsattributesdefaultValue_ibfk_2` (`UpdaterUserId`),
    CONSTRAINT `goodsattributesdefaultValue_ibfk_1` FOREIGN KEY (`CreatorUserId`) REFERENCES `user` (`UserId`) ON UPDATE CASCADE,
    CONSTRAINT `goodsattributesdefaultValue_ibfk_2` FOREIGN KEY (`UpdaterUserId`) REFERENCES `user` (`UserId`) ON UPDATE CASCADE
);
```

```

CREATE TABLE IF NOT EXISTS `goods` (
  `GoodsId` int NOT NULL AUTO_INCREMENT,
  `GoodsName` varchar(255) CHARACTER SET utf8mb3 COLLATE utf8mb3_general_ci NOT NULL,
  `GoodsGroupId` int NOT NULL,
  `GoodsCreateTime` timestamp NOT NULL DEFAULT CURRENT_TIMESTAMP,
  `GoodsUpdateTime` timestamp NOT NULL DEFAULT CURRENT_TIMESTAMP ON UPDATE CURRENT_TIMESTAMP,
  `CreateUserId` int NOT NULL,
  `UpdateUserId` int DEFAULT NULL,
  PRIMARY KEY (`GoodsId`),
  UNIQUE KEY `GoodsId`(`GoodsId`),
  KEY `goods_ibfk_1` (`CreateUserId`),
  KEY `goods_ibfk_2` (`UpdateUserId`),
  KEY `goods_ibfk_3` (`GoodsGroupId`),
  CONSTRAINT `goods_ibfk_1` FOREIGN KEY (`CreateUserId`) REFERENCES `user` (`UserId`) ON UPDATE CASCADE,
  CONSTRAINT `goods_ibfk_2` FOREIGN KEY (`UpdateUserId`) REFERENCES `user` (`UserId`) ON UPDATE CASCADE,
  CONSTRAINT `goods_ibfk_3` FOREIGN KEY (`GoodsGroupId`) REFERENCES `goodsgroup` (`GoodsGroupId`) ON UPDATE CASCADE
);

```

```

CREATE TABLE IF NOT EXISTS `assinedattributestogoods` (
  `GoodsId` int NOT NULL ,
  `GoodsAttributesId` int NOT NULL ,
  `AttributeValue` varchar(255) CHARACTER SET utf8mb3 COLLATE utf8mb3_general_ci NOT NULL,
  PRIMARY KEY (`GoodsId`,`GoodsAttributesId`),
  KEY `assinedattributestogoods_ibfk_1` (`GoodsAttributesId`),
  CONSTRAINT `assinedattributestogoods_ibfk_1` FOREIGN KEY (`GoodsAttributesId`) REFERENCES `goodsattributes` (`GoodsAttributesId`) ON UPDATE CASCADE,
  CONSTRAINT `assinedattributestogoods_ibfk_2` FOREIGN KEY (`GoodsId`) REFERENCES `goods` (`GoodsId`) ON UPDATE CASCADE
);

```

```

CREATE TABLE IF NOT EXISTS `deliveredgoods` (
  `DeliveredGoodsId` int NOT NULL AUTO_INCREMENT,
  `DeliveredGoodsSerial` int NOT NULL,
  `DeliveredGoodsCreateTime` timestamp NOT NULL DEFAULT CURRENT_TIMESTAMP,
  `DeliveredGoodsUpdateTime` timestamp NOT NULL DEFAULT CURRENT_TIMESTAMP ON UPDATE CURRENT_TIMESTAMP,
  `CreateUserId` int DEFAULT NULL,
  `UpdateUserId` int DEFAULT NULL,
  `OwnerUserId` int DEFAULT NULL,
  `UpdaterUserId` int DEFAULT NULL,
  `RelatedComputerPropertyNumber` int DEFAULT NULL,
  `GoodsId` int NOT NULL,
  `AreaId` int DEFAULT NULL,
  `BuildingId` int DEFAULT NULL,
  PRIMARY KEY (`DeliveredGoodsId`),
  UNIQUE KEY `DeliveredGoodsSerial` (`DeliveredGoodsSerial`),
  KEY `deliveredgoods_ibfk_1` (`CreateUserId`),
  KEY `deliveredgoods_ibfk_2` (`UpdateUserId`),
  KEY `deliveredgoods_ibfk_3` (`OwnerUserId`),
  KEY `deliveredgoods_ibfk_4` (`UpdaterUserId`),
  KEY `deliveredgoods_ibfk_5` (`RelatedComputerPropertyNumber`),
  KEY `deliveredgoods_ibfk_6` (`GoodsId`),
  KEY `deliveredgoods_ibfk_7` (`AreaId`),
  KEY `deliveredgoods_ibfk_8` (`BuildingId`),
  CONSTRAINT `deliveredgoods_ibfk_1` FOREIGN KEY (`CreateUserId`) REFERENCES `user` (`UserId`) ON UPDATE CASCADE,
  CONSTRAINT `deliveredgoods_ibfk_2` FOREIGN KEY (`UpdateUserId`) REFERENCES `user` (`UserId`) ON UPDATE CASCADE,
  CONSTRAINT `deliveredgoods_ibfk_3` FOREIGN KEY (`OwnerUserId`) REFERENCES `user` (`UserId`) ON UPDATE CASCADE,
  CONSTRAINT `deliveredgoods_ibfk_4` FOREIGN KEY (`UpdaterUserId`) REFERENCES `user` (`UserId`) ON UPDATE CASCADE,
  CONSTRAINT `deliveredgoods_ibfk_5` FOREIGN KEY (`RelatedComputerPropertyNumber`) REFERENCES `computer` (`ComputerPropertyNumber`) ON UPDATE CASCADE,
  CONSTRAINT `deliveredgoods_ibfk_6` FOREIGN KEY (`GoodsId`) REFERENCES `goods` (`GoodsId`) ON UPDATE CASCADE,
  CONSTRAINT `deliveredgoods_ibfk_7` FOREIGN KEY (`AreaId`) REFERENCES `area` (`AreaId`) ON UPDATE CASCADE,
  CONSTRAINT `deliveredgoods_ibfk_8` FOREIGN KEY (`BuildingId`) REFERENCES `building` (`BuildingId`) ON UPDATE CASCADE
);

```

```

CREATE TABLE IF NOT EXISTS `ticketstatus` (
  `TicketStatusId` int NOT NULL AUTO_INCREMENT,
  `TicketStatusName` varchar(255) CHARACTER SET utf8mb3 COLLATE utf8mb3_general_ci NOT NULL,
  `TicketStatusCreateTime` timestamp NOT NULL DEFAULT CURRENT_TIMESTAMP,
  `TicketStatusUpdateTime` timestamp NOT NULL DEFAULT CURRENT_TIMESTAMP ON UPDATE CURRENT_TIMESTAMP,
  PRIMARY KEY (`TicketStatusId`),
  UNIQUE KEY `TicketStatusId` (`TicketStatusId`)
);

INSERT INTO ticketstatus(TicketStatusName)
VALUES
  ('أنتظار'),
  ('تم حل المشكلة'),
  ('مكتملة'),
  ('مقدمة'),
  ('مغلقة');

```

```

CREATE TABLE IF NOT EXISTS `ticketsubject` (
  `TicketSubjectId` int NOT NULL AUTO_INCREMENT,
  `TicketSubjectName` varchar(255) CHARACTER SET utf8mb3 COLLATE utf8mb3_general_ci NOT NULL,
  `TicketSubjectCreateTime` timestamp NOT NULL DEFAULT CURRENT_TIMESTAMP,
  `TicketSubjectUpdateTime` timestamp NOT NULL DEFAULT CURRENT_TIMESTAMP ON UPDATE CURRENT_TIMESTAMP,
  UNIQUE KEY `TicketSubjectId` (`TicketSubjectId`)
);

INSERT INTO ticketsubject (TicketSubjectName)
VALUES
  ('جذب'),
  ('النفاذ'),
  ('الإرادة'),
  ('غير داخل'),
  ('طلبات'),
  ('أهتمامات');

```

```

CREATE TABLE IF NOT EXISTS `ticket` (
  `TicketId` int NOT NULL AUTO_INCREMENT,
  `TicketStatusId` int NOT NULL,
  `TicketSubjectId` int NOT NULL,
  `CreatorUserId` int NOT NULL,
  `UpdaterUserId` int DEFAULT NULL,
  `TicketCreateTime` timestamp NOT NULL DEFAULT CURRENT_TIMESTAMP,
  `TicketUpdateTime` timestamp NOT NULL DEFAULT CURRENT_TIMESTAMP ON UPDATE CURRENT_TIMESTAMP,
  `TicketDescription` text NOT NULL,
  `DeliveredGoodsId` int DEFAULT NULL,
  `ComputerPropertyNumber` int DEFAULT NULL,
  PRIMARY KEY (`TicketId`),
  KEY `ticket_ibfk_1` (`TicketStatusId`),
  KEY `ticket_ibfk_2` (`TicketSubjectId`),
  KEY `ticket_ibfk_3` (`CreatorUserId`),
  KEY `ticket_ibfk_4` (`DeliveredGoodsId`),
  KEY `ticket_ibfk_5` (`ComputerPropertyNumber`),
  CONSTRAINT `ticket_ibfk_1` FOREIGN KEY (`TicketStatusId`) REFERENCES `ticketstatus` (`TicketStatusId`) ON UPDATE CASCADE,
  CONSTRAINT `ticket_ibfk_2` FOREIGN KEY (`TicketSubjectId`) REFERENCES `ticketsubject` (`TicketSubjectId`) ON UPDATE CASCADE,
  CONSTRAINT `ticket_ibfk_3` FOREIGN KEY (`CreatorUserId`) REFERENCES `user` (`UserId`) ON UPDATE CASCADE,
  CONSTRAINT `ticket_ibfk_4` FOREIGN KEY (`DeliveredGoodsId`) REFERENCES `deliveredgoods` (`DeliveredGoodsId`) ON UPDATE CASCADE,
  CONSTRAINT `ticket_ibfk_5` FOREIGN KEY (`ComputerPropertyNumber`) REFERENCES `computer` (`ComputerPropertyNumber`) ON UPDATE CASCADE
);

```

```

CREATE TABLE IF NOT EXISTS `usersreferredticket` (
  `ReferredUserId` int NOT NULL,
  `TicketId` int NOT NULL,
  `ReferredTicketDate` datetime NOT NULL,
  `ReferredTicketDescription` text NOT NULL,
  PRIMARY KEY (`ReferredUserId`, `TicketId`),
  KEY `usersreferredticket_ibfk_2` (`TicketId`),
  CONSTRAINT `usersreferredticket_ibfk_1` FOREIGN KEY (`ReferredUserId`) REFERENCES `user` (`UserId`) ON UPDATE CASCADE,
  CONSTRAINT `usersreferredticket_ibfk_2` FOREIGN KEY (`TicketId`) REFERENCES `ticket` (`TicketId`) ON UPDATE CASCADE
);

```

```

create table if not exists `updater`(
  `UpdaterId` INT NOT NULL AUTO_INCREMENT,
  `UpdaterCreateTime` timestamp NOT NULL DEFAULT CURRENT_TIMESTAMP,
  `UpdaterUpdateTime` timestamp NOT NULL DEFAULT CURRENT_TIMESTAMP ON UPDATE CURRENT_TIMESTAMP,
  `UpdaterDoneTime` date not null,
  `CreatorUserId` int NOT NULL,
  `UpdaterUserId` int DEFAULT NULL,
  `OwnerUserId` int DEFAULT NULL,
  PRIMARY KEY(`UpdaterId`),
  UNIQUE KEY `UpdaterId`(`UpdaterId`),
  CONSTRAINT `updater_ibfk_1` FOREIGN KEY(`CreatorUserId`) REFERENCES `user`(`UserId`) ON UPDATE CASCADE,
  CONSTRAINT `updater_ibfk_2` FOREIGN KEY(`UpdaterUserId`) REFERENCES `user`(`UserId`) ON UPDATE CASCADE,
  CONSTRAINT `updater_ibfk_3` FOREIGN KEY(`OwnerUserId`) REFERENCES `user`(`UserId`) ON UPDATE CASCADE
);
ALTER TABLE `updater`
ADD `TicketId` INT NOT NULL,
ADD KEY `updater_ibfk_4` (`TicketId`),
ADD CONSTRAINT `updater_ibfk_4` FOREIGN KEY (`TicketId`) REFERENCES `ticket` (`TicketId`) ON UPDATE CASCADE;

```

```
CREATE TABLE IF NOT EXISTS `replacementdeliveygoodsinupdate` (
  `UpdaterId` int NOT NULL,
  `DeliveredGoodsId` int NOT NULL,
  `ReplacementDescription` text NOT NULL,
  PRIMARY KEY (`UpdaterId`),`DeliveredGoodsId`),
  KEY `replacementdeliveygoodsinupdate_ibfk_2` (`DeliveredGoodsId`),
  CONSTRAINT `replacementdeliveygoodsinupdate_ibfk_1` FOREIGN KEY (`UpdaterId`) REFERENCES `updater`(`UpdaterId`)
  ON UPDATE CASCADE,
  CONSTRAINT `replacementdeliveygoodsinupdate_ibfk_2` FOREIGN KEY (`DeliveredGoodsId`) REFERENCES `deliveredgoods`(`DeliveredGoodsId`)
  ON UPDATE CASCADE
);
```

```
CREATE TABLE IF NOT EXISTS `exchanging` (
  `ExchangingId` int NOT NULL AUTO_INCREMENT,
  `ExchangingDescription` text NOT NULL,
  `ExchangingUpdateTime` timestamp NOT NULL DEFAULT CURRENT_TIMESTAMP ON UPDATE CURRENT_TIMESTAMP,
  `CreatorUserId` int NOT NULL,
  `UpdaterUserId` int DEFAULT NULL,
  `UserExchangerId` int NOT NULL ,
  `TicketId` INT NULL,
  PRIMARY KEY (`ExchangingId`),
  UNIQUE KEY `ExchangingId` (`ExchangingId`),
  KEY `exchanging_ibfk_1` (`CreatorUserId`),
  KEY `exchanging_ibfk_2` (`UpdaterUserId`),
  CONSTRAINT `exchanging_ibfk_1` FOREIGN KEY (`CreatorUserId`) REFERENCES `user`(`UserId`) ON UPDATE CASCADE,
  CONSTRAINT `exchanging_ibfk_2` FOREIGN KEY (`UpdaterUserId`) REFERENCES `user`(`UserId`) ON UPDATE CASCADE,
  CONSTRAINT `exchanging_ibfk_3` FOREIGN KEY (`UserExchangerId`) REFERENCES `user`(`UserId`) ON UPDATE
  CONSTRAINT `exchanging_ibfk_4` FOREIGN KEY (`TicketId`) REFERENCES `ticket`(`TicketId`) ON UPDATE CASCADE
);
```

```
CREATE TABLE IF NOT EXISTS `deliveredgoodsexchanging` (
  `DeliveredGoodsId` int NOT NULL,
  `ExchangingId` int NOT NULL,
  `CreateTime` date NOT NULL,
  `DoneTime` date DEFAULT NULL,
  PRIMARY KEY (`DeliveredGoodsId`),`ExchangingId`),
  KEY `deliveredgoodsexchanging_ibfk_1` (`ExchangingId`),
  CONSTRAINT `deliveredgoodsexchanging_ibfk_1` FOREIGN KEY (`ExchangingId`) REFERENCES `exchanging`(`ExchangingId`)
  ON UPDATE CASCADE,
  CONSTRAINT `deliveredgoodsexchanging_ibfk_2` FOREIGN KEY (`DeliveredGoodsId`) REFERENCES `deliveredgoods`(`DeliveredGoodsId`)
  ON UPDATE CASCADE
);
```

```
CREATE TABLE IF NOT EXISTS `computersexchanging` (
  `ComputerPropertyNumber` int NOT NULL,
  `ExchangingId` int NOT NULL,
  `CreateTime` date NOT NULL,
  `DoneTime` date NOT NULL,
  PRIMARY KEY (`ComputerPropertyNumber`),`ExchangingId`),
  KEY `computersexchanging_ibfk_1` (`ExchangingId`),
  CONSTRAINT `computersexchanging_ibfk_1` FOREIGN KEY (`ExchangingId`) REFERENCES `exchanging`(`ExchangingId`)
  ON UPDATE CASCADE,
  CONSTRAINT `computersexchanging_ibfk_2` FOREIGN KEY (`ComputerPropertyNumber`) REFERENCES `computer`(`ComputerPropertyNumber`)
  ON UPDATE CASCADE
);
```

```
CREATE TABLE IF NOT EXISTS `abortion` (
    `AbortionId` int NOT NULL AUTO_INCREMENT,
    `AbortionCreateTime` timestamp NOT NULL DEFAULT CURRENT_TIMESTAMP,
    `AbortionUpdateTime` timestamp NOT NULL DEFAULT CURRENT_TIMESTAMP ON UPDATE CURRENT_TIMESTAMP,
    `AbortionDoneTime` date NOT NULL,
    `CreatorUserId` int NOT NULL,
    `OwnerUserId` int NOT NULL,
    `UpdaterUserId` int DEFAULT NULL,
    `DeliveredGoodsId` int DEFAULT NULL,
    `ComputerPropertyNumber` int DEFAULT NULL,
    `TicketId` int NOT NULL,
    PRIMARY KEY (`AbortionId`),
    UNIQUE KEY `AbortionId` (`AbortionId`),
    KEY `abortion_ibfk_1` (`CreatorUserId`),
    KEY `abortion_ibfk_2` (`UpdaterUserId`),
    KEY `abortion_ibfk_3` (`OwnerUserId`),
    KEY `abortion_ibfk_4` (`DeliveredGoodsId`),
    KEY `abortion_ibfk_5` (`ComputerPropertyNumber`),
    KEY `abortion_ibfk_6` (`TicketId`),
    CONSTRAINT `abortion_ibfk_1` FOREIGN KEY (`CreatorUserId`) REFERENCES `user` (`UserId`) ON UPDATE CASCADE,
    CONSTRAINT `abortion_ibfk_2` FOREIGN KEY (`UpdaterUserId`) REFERENCES `user` (`UserId`) ON UPDATE CASCADE,
    CONSTRAINT `abortion_ibfk_3` FOREIGN KEY (`OwnerUserId`) REFERENCES `user` (`UserId`) ON UPDATE CASCADE,
    CONSTRAINT `abortion_ibfk_4` FOREIGN KEY (`DeliveredGoodsId`) REFERENCES `deliveredgoods` (`DeliveredGoodsId`) ON UPDATE CASCADE,
    CONSTRAINT `abortion_ibfk_5` FOREIGN KEY (`ComputerPropertyNumber`) REFERENCES `computer` (`ComputerPropertyNumber`) ON UPDATE CASCADE,
    CONSTRAINT `abortion_ibfk_6` FOREIGN KEY (`TicketId`) REFERENCES `ticket` (`TicketId`) ON UPDATE CASCADE
);
```

```
CREATE TABLE IF NOT EXISTS `internalrepair` (
    `InternalRepairId` int NOT NULL AUTO_INCREMENT,
    `InternalRepairDescription` text NOT NULL,
    `InternalRepairChangingDescription` text NOT NULL,
    `InternalRepairCreateTime` timestamp NOT NULL DEFAULT CURRENT_TIMESTAMP,
    `InternalRepairUpdateTime` timestamp NOT NULL DEFAULT CURRENT_TIMESTAMP ON UPDATE CURRENT_TIMESTAMP,
    `InternalRepairDoneTime` date NOT NULL,
    `CreatorUserId` int NOT NULL,
    `OwnerUserId` int NOT NULL,
    `UpdaterUserId` int DEFAULT NULL,
    `DeliveredGoodsId` int NOT NULL,
    `TicketId` int NOT NULL,
    PRIMARY KEY (`InternalRepairId`),
    UNIQUE KEY `InternalRepairId` (`InternalRepairId`),
    KEY `internalrepair_ibfk_1` (`CreatorUserId`),
    KEY `internalrepair_ibfk_2` (`UpdaterUserId`),
    KEY `internalrepair_ibfk_3` (`DeliveredGoodsId`),
    KEY `internalrepair_ibfk_4` (`TicketId`),
    CONSTRAINT `internalrepair_ibfk_1` FOREIGN KEY (`CreatorUserId`) REFERENCES `user` (`UserId`) ON UPDATE CASCADE,
    CONSTRAINT `internalrepair_ibfk_2` FOREIGN KEY (`UpdaterUserId`) REFERENCES `user` (`UserId`) ON UPDATE CASCADE,
    CONSTRAINT `internalrepair_ibfk_3` FOREIGN KEY (`DeliveredGoodsId`) REFERENCES `deliveredgoods` (`DeliveredGoodsId`) ON UPDATE CASCADE,
    CONSTRAINT `internalrepair_ibfk_4` FOREIGN KEY (`TicketId`) REFERENCES `ticket` (`TicketId`) ON UPDATE CASCADE
);
```

```
CREATE TABLE IF NOT EXISTS `outbounddocument` (
    `OutboundDocumentSerial` int NOT NULL AUTO_INCREMENT,
    `OutboundDocumentDescription` text NOT NULL,
    `OutboundDocumentCreateTime` timestamp NOT NULL DEFAULT CURRENT_TIMESTAMP,
    `OutboundDocumentUpdateTime` timestamp NOT NULL DEFAULT CURRENT_TIMESTAMP ON UPDATE CURRENT_TIMESTAMP,
    `OutboundDocumentDoneTime` date NOT NULL,
    `DeliveredGoodsId` int NOT NULL,
    `CreatorUserId` int NOT NULL,
    `OwnerUserId` int NOT NULL,
    `UpdaterUserId` int DEFAULT NULL,
    `TicketId` int NOT NULL,
    PRIMARY KEY (`OutboundDocumentSerial`),
    UNIQUE KEY `OutboundDocumentSerial` (`OutboundDocumentSerial`),
    KEY `outbounddocument_ibfk_1` (`DeliveredGoodsId`),
    KEY `outbounddocument_ibfk_2` (`CreatorUserId`),
    KEY `outbounddocument_ibfk_3` (`UpdaterUserId`),
    KEY `outbounddocument_ibfk_4` (`OwnerUserId`),
    KEY `outbounddocument_ibfk_5` (`TicketId`),
    CONSTRAINT `outbounddocument_ibfk_1` FOREIGN KEY (`DeliveredGoodsId`) REFERENCES `deliveredgoods` (`DeliveredGoodsId`) ON UPDATE CASCADE,
    CONSTRAINT `outbounddocument_ibfk_2` FOREIGN KEY (`CreatorUserId`) REFERENCES `user` (`UserId`) ON UPDATE CASCADE,
    CONSTRAINT `outbounddocument_ibfk_3` FOREIGN KEY (`UpdaterUserId`) REFERENCES `user` (`UserId`) ON UPDATE CASCADE,
    CONSTRAINT `outbounddocument_ibfk_4` FOREIGN KEY (`OwnerUserId`) REFERENCES `user` (`UserId`) ON UPDATE CASCADE,
    CONSTRAINT `outbounddocument_ibfk_5` FOREIGN KEY (`TicketId`) REFERENCES `ticket` (`TicketId`) ON UPDATE CASCADE
);
```

```
CREATE TABLE IF NOT EXISTS `installation` (
    `InstallationId` int NOT NULL AUTO_INCREMENT,
    `InstallationDescription` text NOT NULL,
    `InstallationUpdateTime` timestamp NOT NULL DEFAULT CURRENT_TIMESTAMP ON UPDATE CURRENT_TIMESTAMP,
    `TicketId` int NOT NULL,
    `CreatorUserId` int NOT NULL,
    `UpdaterUserId` int DEFAULT NULL,
    PRIMARY KEY (`InstallationId`),
    UNIQUE KEY `InstallationId` (`InstallationId`),
    KEY `installation_ibfk_1` (`TicketId`),
    KEY `installation_ibfk_2` (`CreatorUserId`),
    KEY `installation_ibfk_3` (`UpdaterUserId`),
    CONSTRAINT `installation_ibfk_1` FOREIGN KEY (`TicketId`) REFERENCES `ticket` (`TicketId`) ON UPDATE CASCADE,
    CONSTRAINT `installation_ibfk_2` FOREIGN KEY (`CreatorUserId`) REFERENCES `user` (`UserId`) ON UPDATE CASCADE,
    CONSTRAINT `installation_ibfk_3` FOREIGN KEY (`UpdaterUserId`) REFERENCES `user` (`UserId`) ON UPDATE CASCADE
);
```

```
CREATE TABLE IF NOT EXISTS `softwares` (
    `SoftwareId` int NOT NULL AUTO_INCREMENT,
    `SoftwareName` varchar(255) CHARACTER SET utf8mb3 COLLATE utf8mb3_general_ci NOT NULL,
    `SoftwareCreateTime` timestamp NOT NULL DEFAULT CURRENT_TIMESTAMP,
    `SoftwareUpdateTime` timestamp NOT NULL DEFAULT CURRENT_TIMESTAMP ON UPDATE CURRENT_TIMESTAMP,
    `CreatorUserId` int NOT NULL,
    `UpdaterUserId` int DEFAULT NULL,
    PRIMARY KEY (`SoftwareId`),
    UNIQUE KEY `SoftwareId` (`SoftwareId`),
    KEY `softwares_ibfk_1` (`CreatorUserId`),
    KEY `softwares_ibfk_2` (`UpdaterUserId`),
    CONSTRAINT `softwares_ibfk_1` FOREIGN KEY (`CreatorUserId`) REFERENCES `user` (`UserId`) ON UPDATE CASCADE,
    CONSTRAINT `softwares_ibfk_2` FOREIGN KEY (`UpdaterUserId`) REFERENCES `user` (`UserId`) ON UPDATE CASCADE
);
```

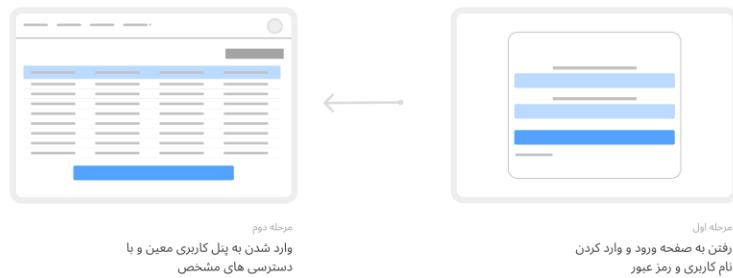
```
CREATE TABLE IF NOT EXISTS `softwaresininstallation` (
    `InstallationId` int NOT NULL,
    `SoftwareId` int NOT NULL,
    `UsageOfSoftwaresInInstallation` text NOT NULL,
    PRIMARY KEY (`InstallationId`, `SoftwareId`),
    KEY `softwaresininstallation_ibfk_1` (`InstallationId`),
    KEY `softwaresininstallation_ibfk_2` (`SoftwareId`),
    CONSTRAINT `softwaresininstallation_ibfk_1` FOREIGN KEY (`InstallationId`) REFERENCES `installation` (`InstallationId`) ON UPDATE CASCADE,
    CONSTRAINT `softwaresininstallation_ibfk_2` FOREIGN KEY (`SoftwareId`) REFERENCES `softwares` (`SoftwareId`) ON UPDATE CASCADE
);
```

```
CREATE TABLE IF NOT EXISTS `installationsoncomputer` (
    `InstallationId` int NOT NULL,
    `ComputerPropertyName` int NOT NULL,
    `InstallationCreateTime` timestamp NOT NULL DEFAULT CURRENT_TIMESTAMP,
    `InstallationDoneTime` date NOT NULL,
    PRIMARY KEY (`InstallationId`, `ComputerPropertyName`),
    KEY `installationsoncomputer_ibfk_1` (`InstallationId`),
    KEY `installationsoncomputer_ibfk_2` (`ComputerPropertyName`),
    CONSTRAINT `installationsoncomputer_ibfk_1` FOREIGN KEY (`InstallationId`) REFERENCES `installation` (`InstallationId`) ON UPDATE CASCADE,
    CONSTRAINT `installationsoncomputer_ibfk_2` FOREIGN KEY (`ComputerPropertyName`) REFERENCES `computer` (`ComputerPropertyName`) ON UPDATE CASCADE
);
```

```
CREATE TABLE IF NOT EXISTS `computersealling` (
    `ComputerSeallingId` int NOT NULL AUTO_INCREMENT,
    `ComputerSeallingNumber` int NOT NULL,
    `ComputerSeallingCreateTime` timestamp NOT NULL DEFAULT CURRENT_TIMESTAMP,
    `ComputerSeallingUpdateTime` timestamp NOT NULL DEFAULT CURRENT_TIMESTAMP ON UPDATE CURRENT_TIMESTAMP,
    `IsExpired` tinyint() DEFAULT '0',
    `ComputerPropertyNumber` int DEFAULT NULL,
    `UpdaterId` int DEFAULT NULL,
    `CreatorUserId` int NOT NULL,
    `UpdaterUserId` int DEFAULT NULL,
    `InternalRepairId` int DEFAULT NULL,
    PRIMARY KEY (`ComputerSeallingId`),
    UNIQUE KEY `ComputerSeallingNumber` (`ComputerSeallingNumber`),
    KEY `computersealling_ibfk_1` (`ComputerPropertyNumber`),
    KEY `computersealling_ibfk_2` (`UpdaterId`),
    KEY `computersealling_ibfk_3` (`CreatorUserId`),
    KEY `computersealling_ibfk_4` (`UpdaterUserId`),
    KEY `computersealling_ibfk_5` (`InternalRepairId`),
    CONSTRAINT `computersealling_ibfk_1` FOREIGN KEY (`ComputerPropertyNumber`) REFERENCES `computer`(`ComputerPropertyNumber`) ON UPDATE CASCADE,
    CONSTRAINT `computersealling_ibfk_2` FOREIGN KEY (`UpdaterId`) REFERENCES `updater`(`UpdaterId`) ON UPDATE CASCADE,
    CONSTRAINT `computersealling_ibfk_3` FOREIGN KEY (`CreatorUserId`) REFERENCES `user`(`UserId`) ON UPDATE CASCADE,
    CONSTRAINT `computersealling_ibfk_4` FOREIGN KEY (`UpdaterUserId`) REFERENCES `user`(`UserId`) ON UPDATE CASCADE,
    CONSTRAINT `computersealling_ibfk_5` FOREIGN KEY (`InternalRepairId`) REFERENCES `internalrepair`(`InternalRepairId`) ON UPDATE CASCADE
);
```

User Flow

:User Flow ورود به پنل کاربری



:User Flow مشاهده همه کاربران





:User Flow تغییر رمز توسط خود کاربر



:User Flow
جزئیات کاربران زیر دست و تغییرات آن



:User Flow
تغییر رمز کاربر زیر دست

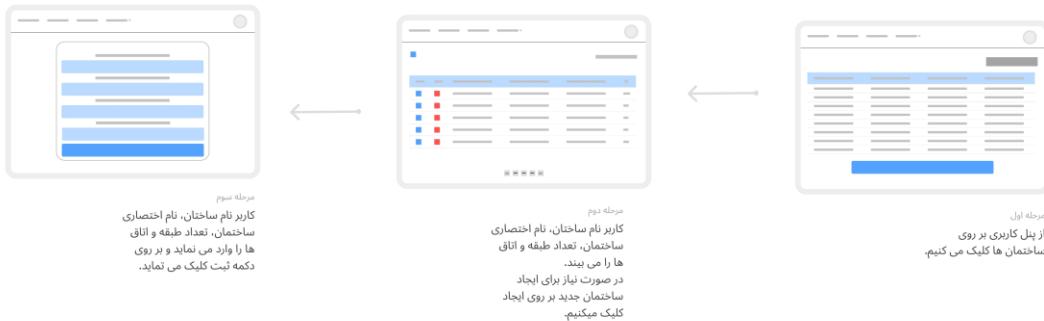


:User Flow

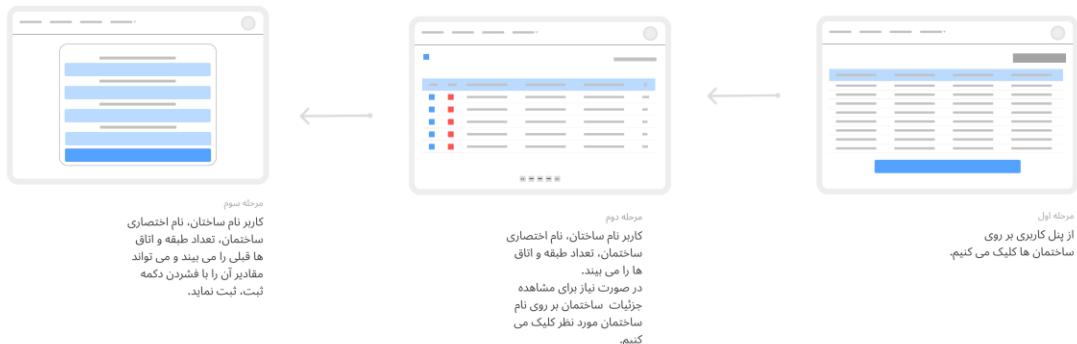
لیست ساختمان



:User Flow
ایجاد ساختمان



:User Flow
جزئیات ساختمان و تغییرات آن

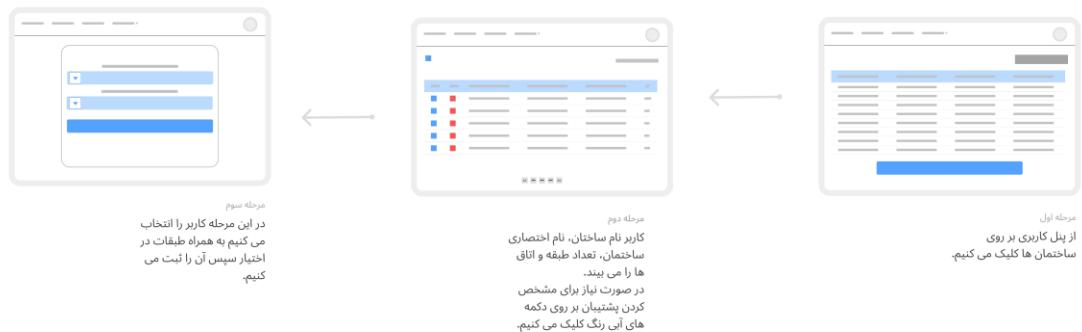


:User Flow

حذف ساختمان



:User Flow
انتخاب پشتیبان هر طبقه ساختمان



:User Flow

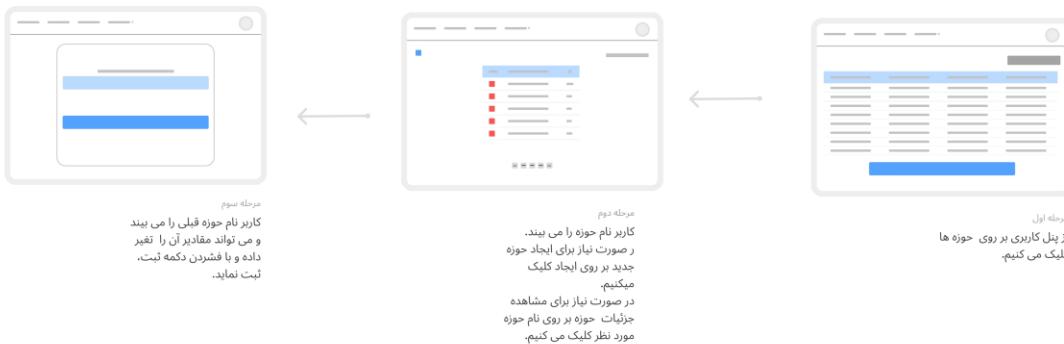
لیست حوزه



:User Flow
ایجاد حوزه



:User Flow
جزئیات حوزه و تغییرات آن



:User Flow

حذف حوزه



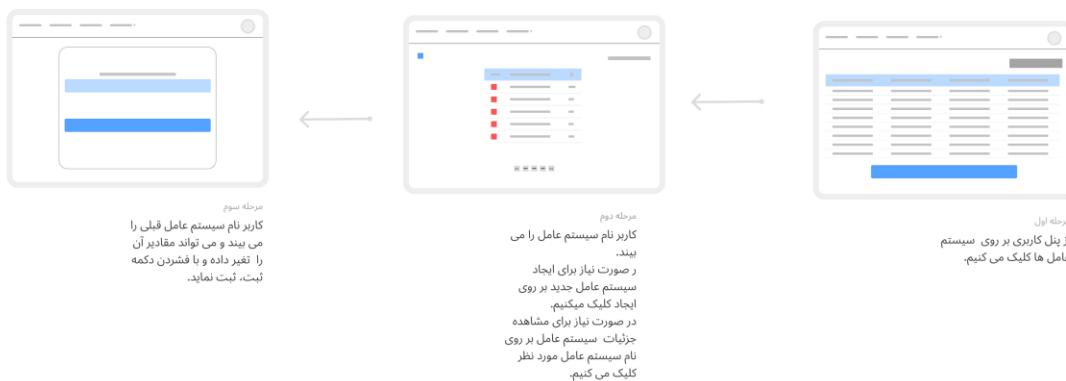
لیست سیستم عامل



:User Flow
ایجاد سیستم عامل

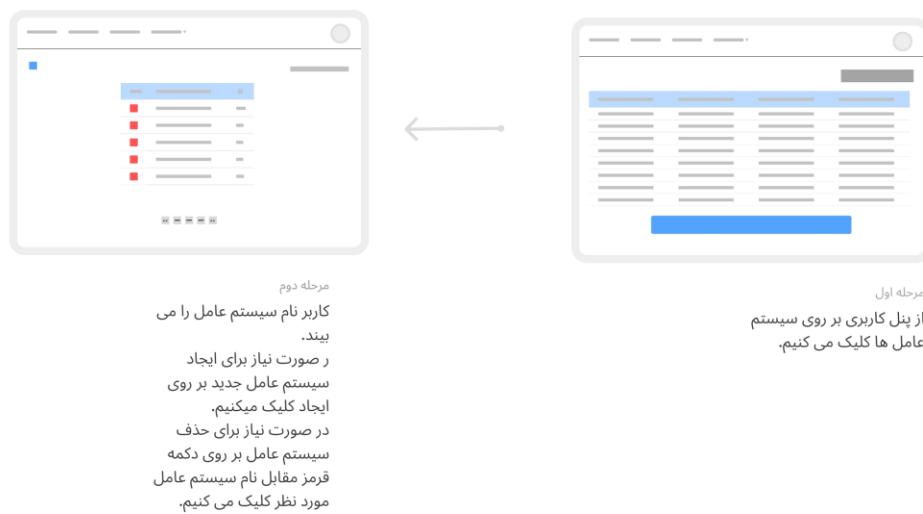


:User Flow
جزئیات سیستم عامل و تغییرات آن



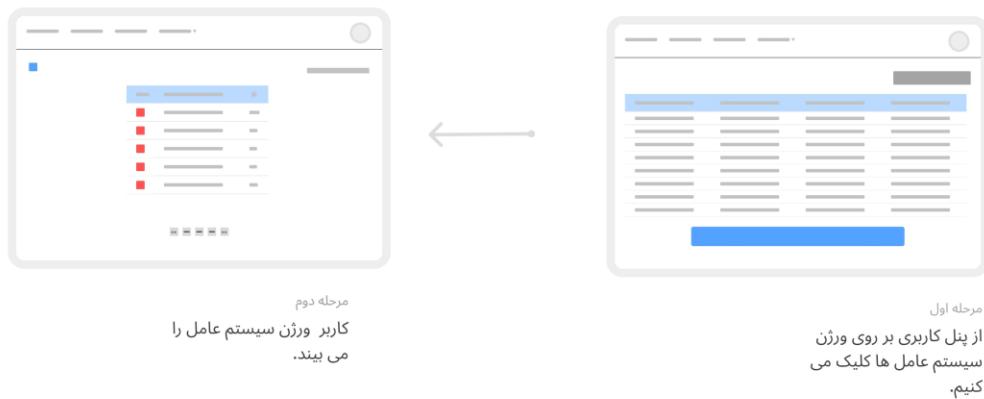
:User Flow

حذف سیستم عامل



:User Flow

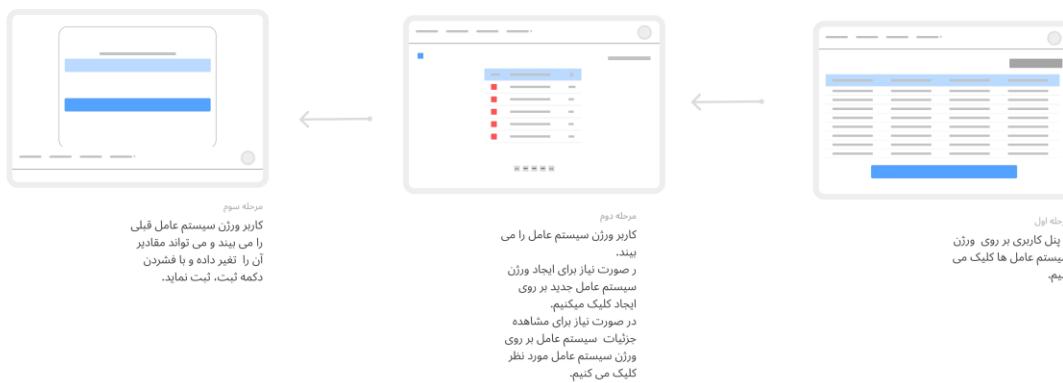
لیست ورژن سیستم عامل



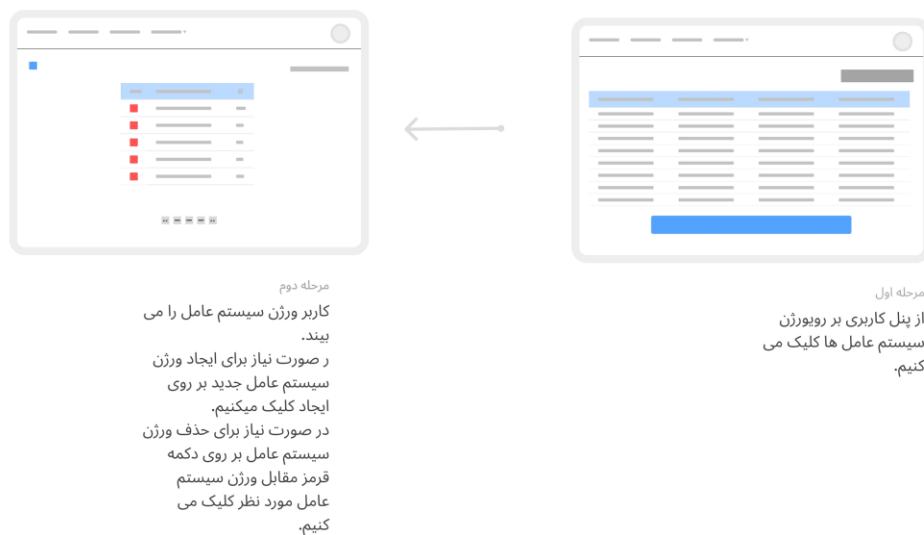
:User Flow
ایجاد ورژن سیستم عامل



:User Flow
جزئیات ورژن سیستم عامل و تغییرات آن

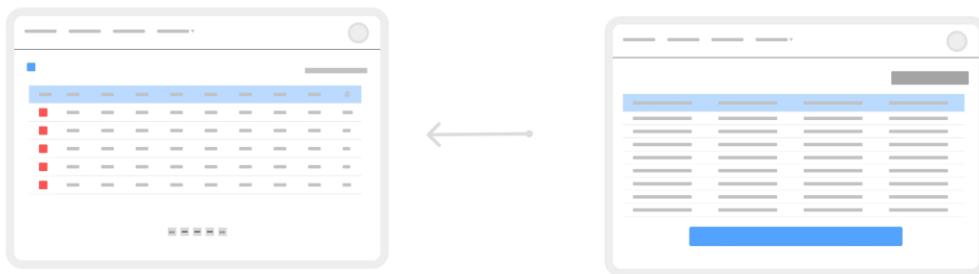


حذف ورژن سیستم عامل



:User Flow

لیست کامپیوتر ها



:User Flow
ایجاد کامپیوتر

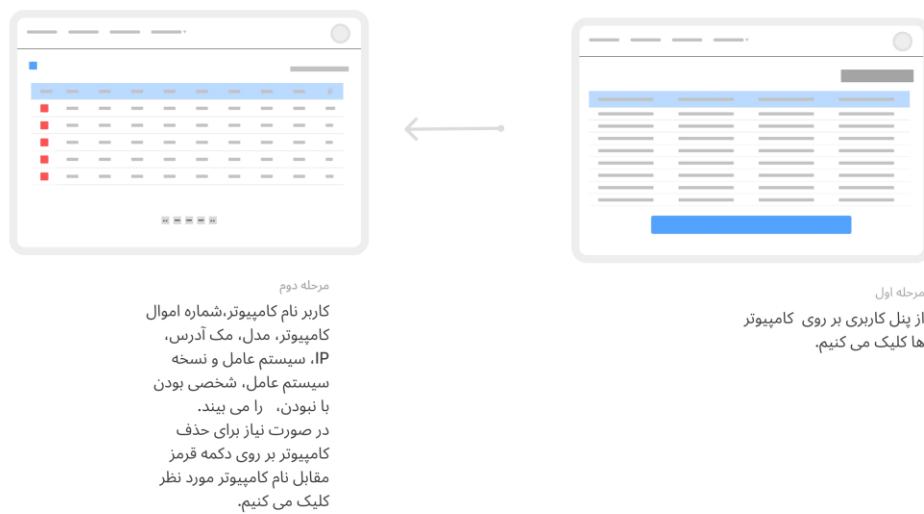


:User Flow
جزئیات کامپیوتر و تغییرات آن



:User Flow

حذف کامپیووتر



:User Flow

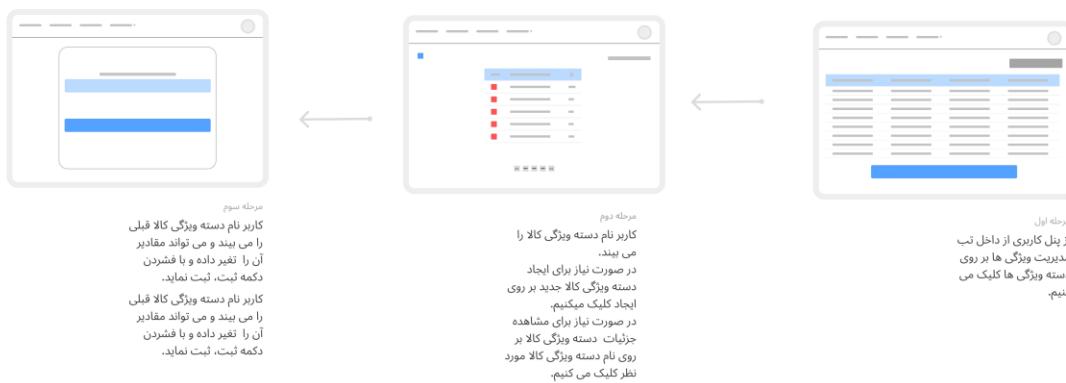
لیست دسته ویژگی



:User Flow
ایجاد دسته ویژگی

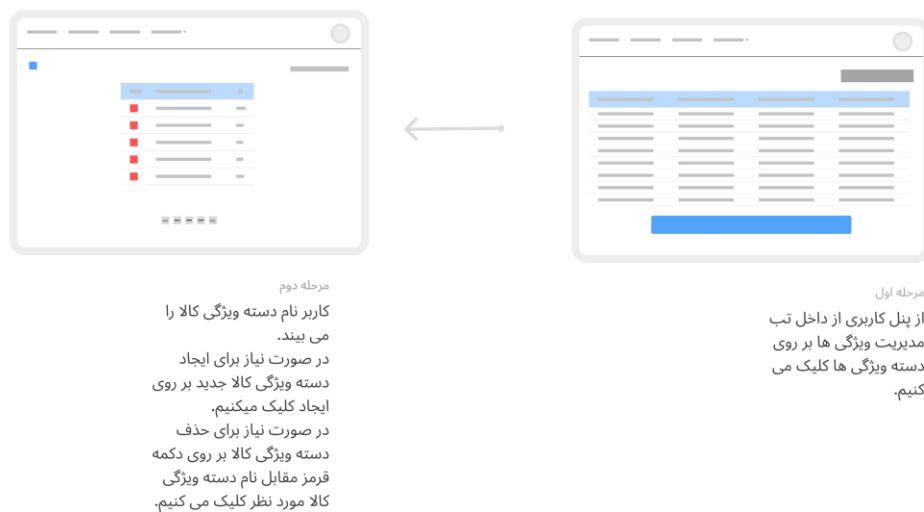


:User Flow
جزئیات دسته ویژگی و تغییرات آن



:User Flow

حذف دسته ویژگی



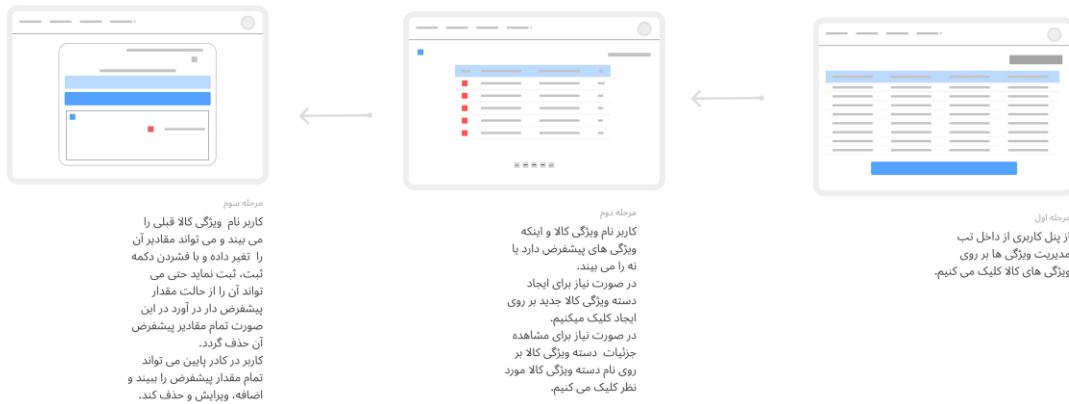
:User Flow
لیست ویژگی کالا



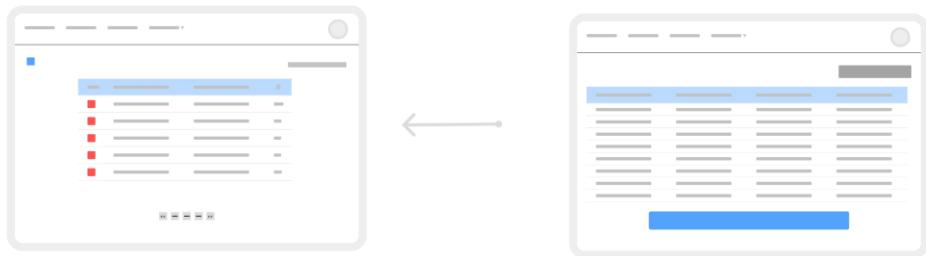
:User Flow
ایجاد ویژگی کالا



:User Flow
جزئیات ویژگی کالا و تغییرات آن



:User Flow
حذف ویژگی کالا



مرحله دوم
کاربر نام ویژگی کالا و اینکه ویژگی های پیشفرض دارد با نه را من بیند.
در صورت نیاز برای ایجاد دسته ویژگی کالا جدید بر روی ایجاد کلیک میکنیم.
در صورت نیاز برای مشاهده جزئیات دسته ویژگی کالا بر روی نام دسته ویژگی کالا مورد نظر کلیک من کنیم.
در صورت نیاز برای حذف دسته ویژگی کالا بر روی دکمه قرمز مقابل نام ویژگی کالا مورد نظر کلیک من کنیم.

مرحله اول
از پنل کاربری از داخل تب مدیریت ویژگی ها بر روی ویژگی های کالا کلیک من کنیم.

:User Flow

ایجاد مقادیر پیشفرض بعد از ایجاد ویژگی کالا



مرحله سوم
کاربر نام ویژگی کالا قبلي را من بیند و من تواند مقادير آن را تغیر داده و با فشردن دکمه ثبت، ثبت نماید حتى من تواند آن را از حالت مقدار پیشفرض دار در اور در این صورت تمام مقادير پیشفرض آن حذف گردد.
کاربر در کادر پايان من تواند تمام مقدار پیشفرض را بیند و اصاها، ویژگی و حذف کند.
حال برای اصاها کردن بعد از ایجاد ویژگی کالا بر روی دکمه ایجاد داخل کادر کلیک من کنیم.

مرحله اول
از پنل کاربری از داخل تب مدیریت ویژگی ها بر روی ویژگی های کالا کلیک من کنیم.

:User Flow
ویرایش مقادیر پیشفرض



:User Flow
ویرایش مقادیر پیشفرض

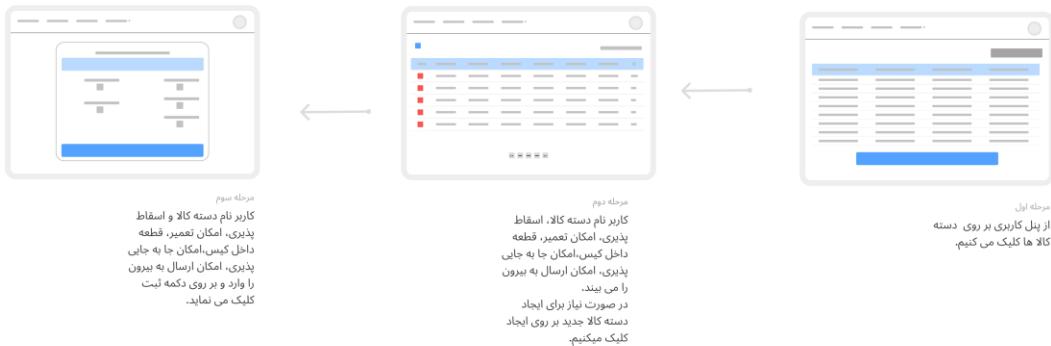


:User Flow

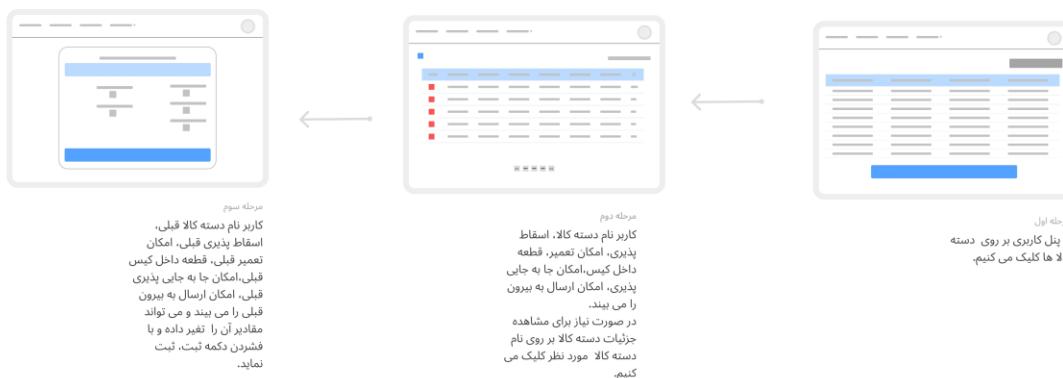
لیست دسته کالا



:User Flow
ایجاد دسته کالا

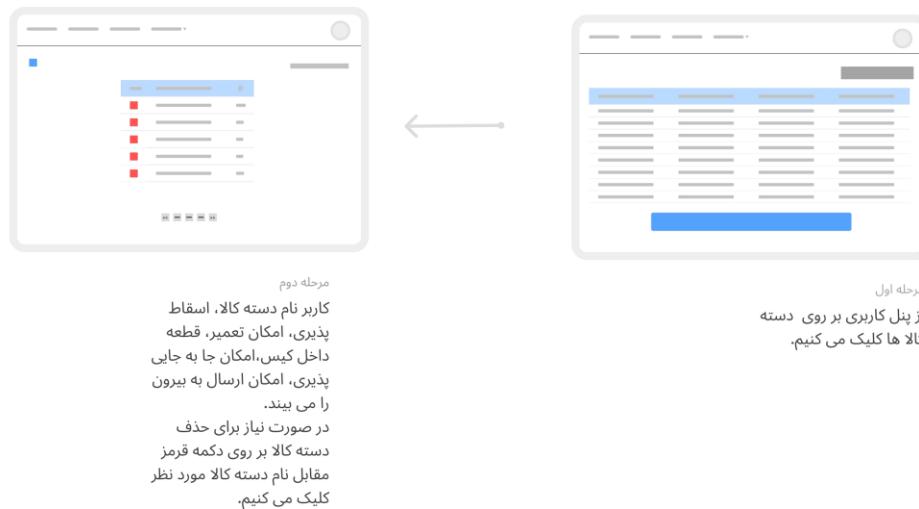


:User Flow
جزئیات دسته کالا و تغییرات آن

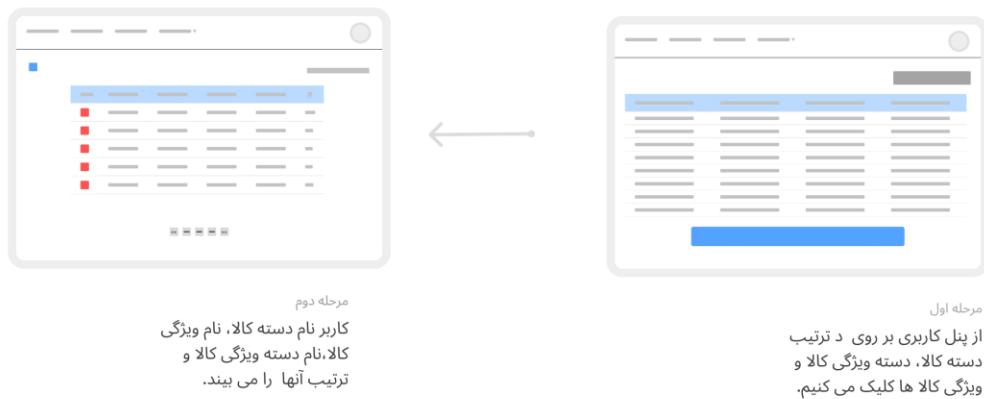


:User Flow

حذف دسته کالا

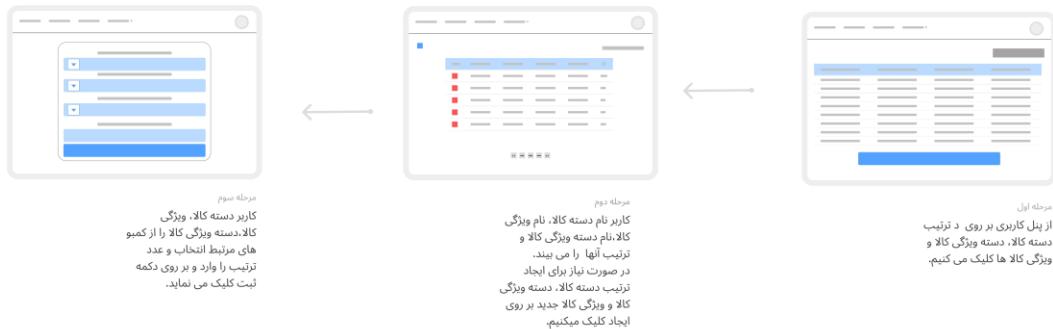


لیست ترتیب دسته کالا، دسته ویژگی کالا و ویژگی کالا



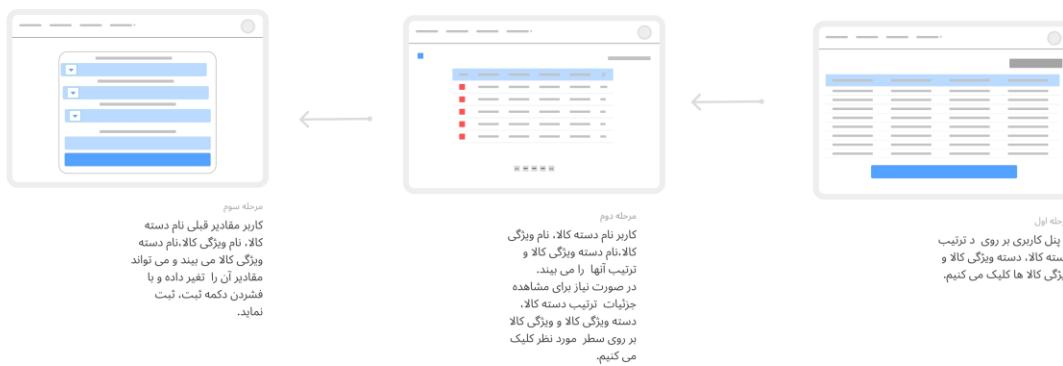
:User Flow

ایجاد ترتیب دسته کالا، دسته ویژگی کالا و ویژگی کالا



:User Flow

جزئیات ترتیب دسته کالا، دسته ویژگی کالا و ویژگی کالا و تغییرات آن



حذف ترتیب دسته کالا، دسته ویژگی کالا و ویژگی کالا و تغییرات آن

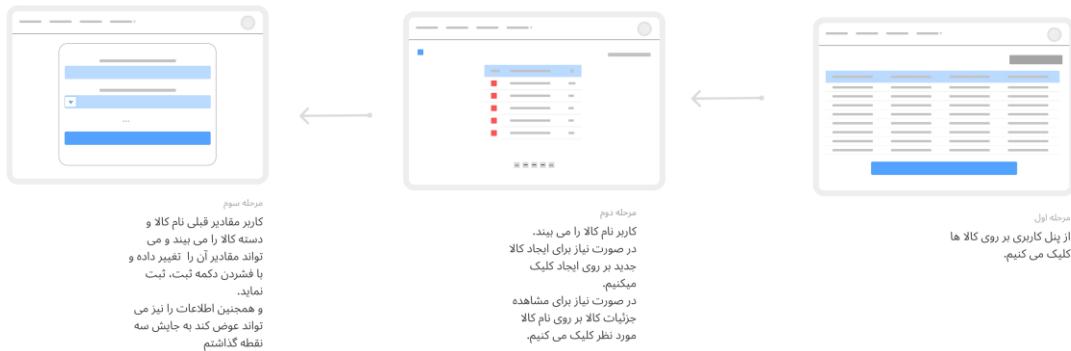


:User Flow

لیست کالاها

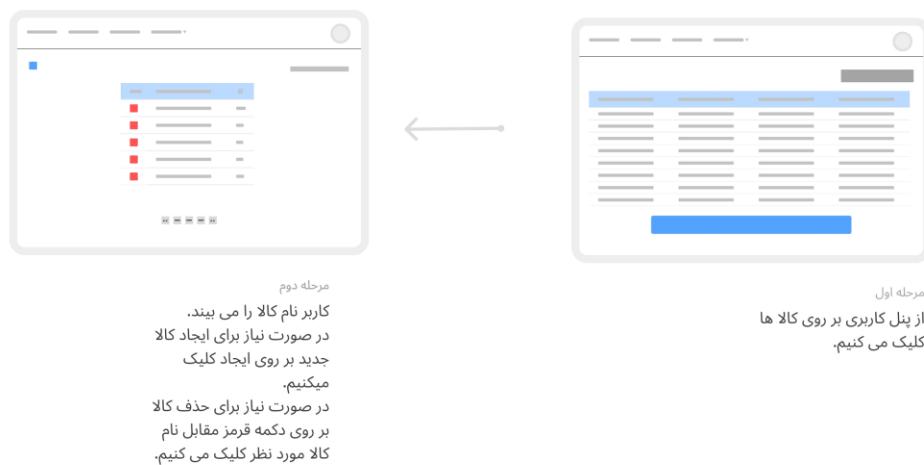


:User Flow
جزئیات کالا و تغییرات آن



:User Flow

حذف کالا



:User Flow

لیست کالاهای تحویلی

لیست کالاهای تحویلی



مرحله اول

از بین کاربری بر روی کالا های تحویلی کلیک می کنیم.

مرحله دوم

کاربر نام کالا و سریال کالای تحویلی را من بیند.

اجداد کالا تحویلی

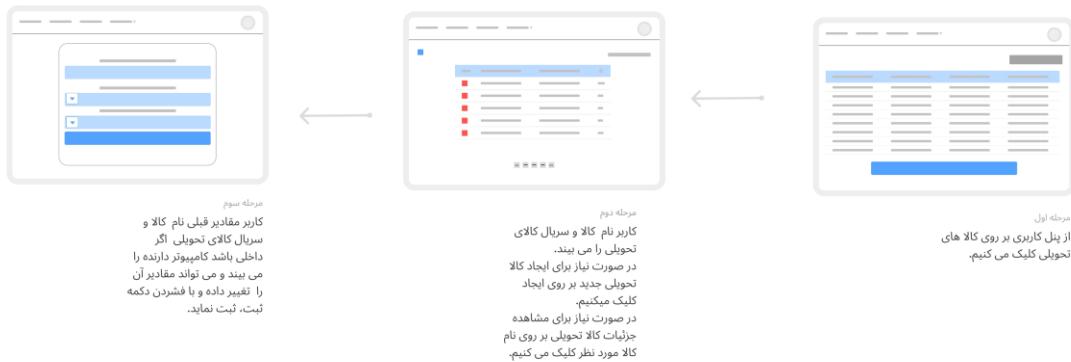


مرحله اول

کاربر نام و سریال کالای تحویلی را با وارد کردن آنها در فیلد نام و سریال کالای تحویلی کلیک می کند.

اجداد کالا تحویلی

:User Flow
جزئیات کالای تحویلی و تغییرات آن



:User Flow

حذف کالای تحویلی



:User Flow

لیست پلمپ



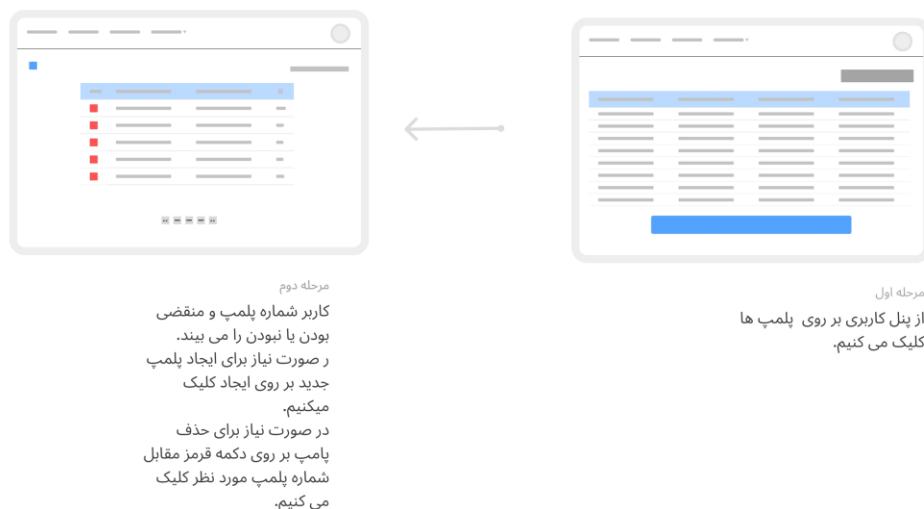
:User Flow
ایجاد پلیمپ



:User Flow
جزئیات پلیمپ و تغییرات آن



حذف پلمس



:User Flow

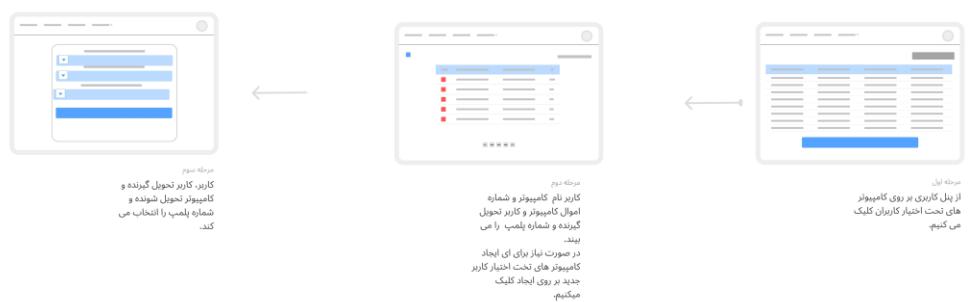
لیست کامپیوتر های تخت اختیار کاربر

لیست کامپیوتر های تخت اختیار کاربر را مشاهده کنید.



:User Flow

ایجاد کامپیوتر های تخت اختیار کاربر

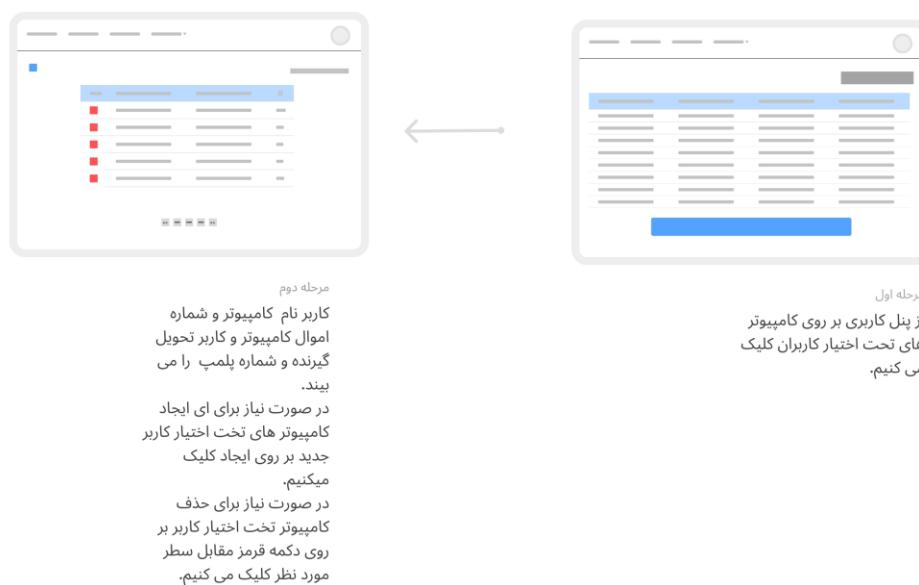


:User Flow
جزئیات کامپیوتر های تخت اختیار کاربر و تغییرات آن



:User Flow

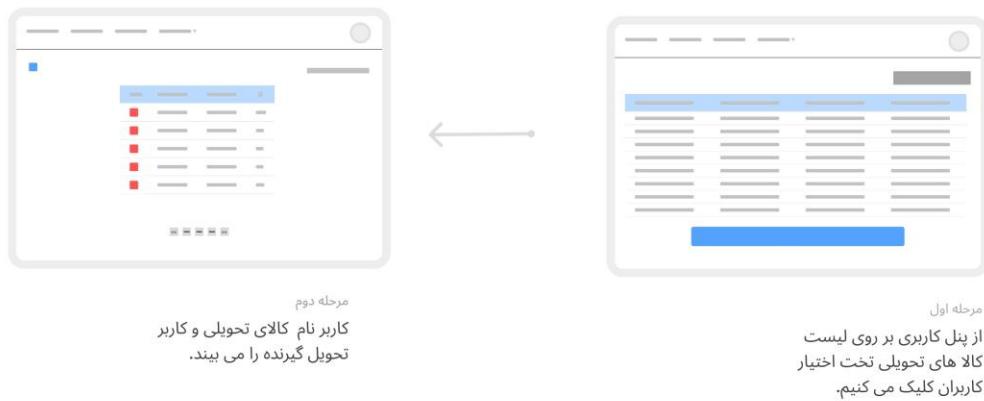
حذف کامپیوتر های تخت اختیار کاربر و تغییرات آن



:User Flow

لیست کالا های تحویلی تخت اختیار کاربر

چگونه در سمت بک اند ما باید اطلاعات کامپیوتر را بر کنیم؟

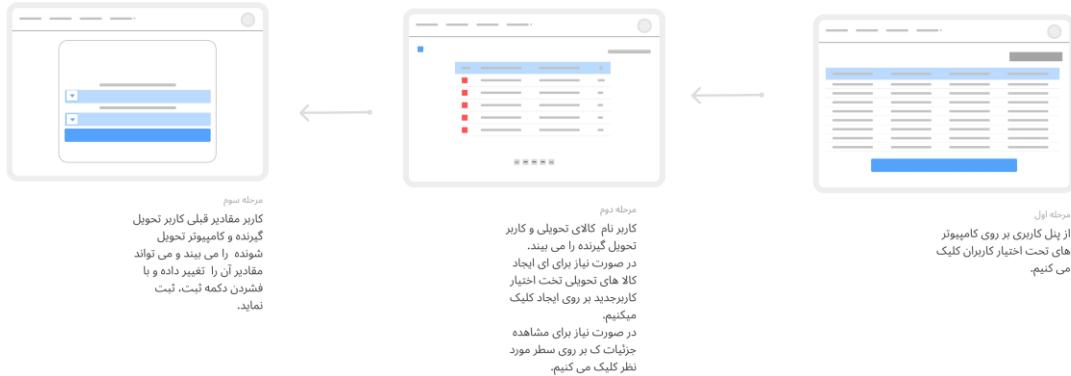


:User Flow

ايجاد کالا های تحویلی تخت اختیار کاربر

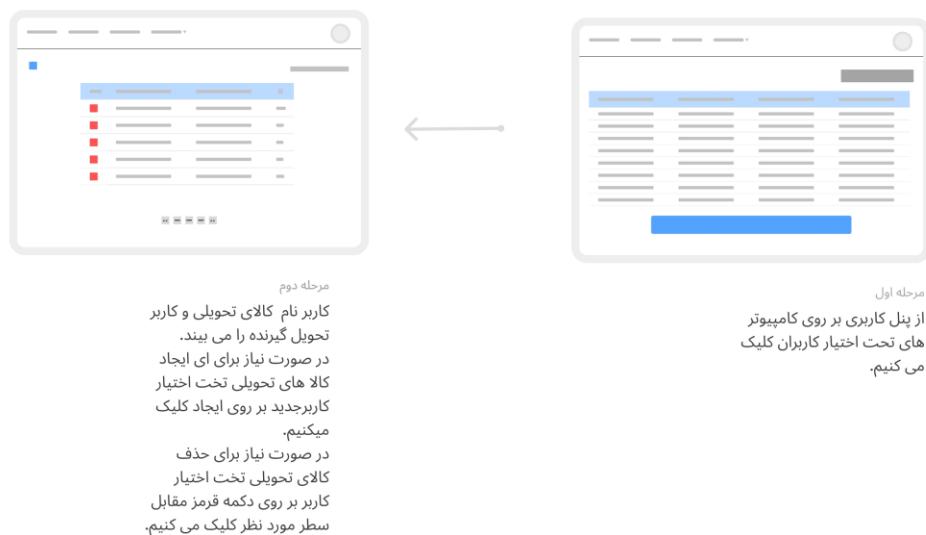


:User Flow
جزئیات کالا های تحویلی تخت اختیار کاربر



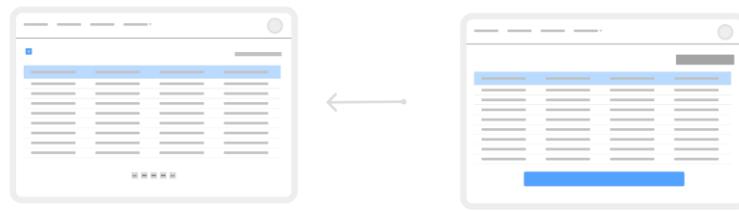
:User Flow

حذف کالا های تحویلی تخت اختیار کاربر



:User Flow

مشاهده همه دارایی های کاربر



مرحله دوم
 تمام دارایی ها قابل رویت اند
 و برای دیدن صفحات بعدی
 من توان بر روی صفحه مورد
 نظر کلیک کرد
 در صورتی که فرد در دو
 ساختمان مجزا کار کند با
 استفاده از کمبو باکس آن را
 منشخص می کنیم که دارایی
 های این ساختمانش داشت

مرحله اول
 از پبل کاربری بر روی دکمه
 همه وارد دارین کلیک می
 کنیم.

:User Flow

صفحه جزئیات دارایی



مرحله سوم
 از صفحه جزئیات محصول بر
 روی کمکه تیک کلیک کرده و
 تیک خود را لست می کنیم
 برای دیدن آثارشات به
 روزرسانی، عملیات تصمیم،
 اسنفاط، جا به جا، تغیر دادن،
 روز اضافی و تغییر دادن
 کلیک کرد بر روی عکس هایشها

مرحله دوم
 تمام دارایی ها قابل رویت اند
 و برای دیدن صفحات بعدی
 من توان بر روی صفحه مورد
 نظر کلیک کرد.
 بر روی دارایی مورد نظر کلیک
 کرده

مرحله اول
 از پبل کاربری بر روی دکمه
 همه وارد دارین کلیک می
 کنیم.

:User Flow

مشاهده دارایی های کاربران زیردست



User Flow

نحوه تیکت کاربر برای هر دارایی خود



:User Flow

مشاهده لیست تیکت ها



:User Flow

مشاهده جزئیات تیکت و باسخ به هر تیکت



تیکت تهییت برای کاربران زیردست



منشاءه جزئیات تیکت و پاسخ خود در هر تیکت



اعمال اسقاط



User Flow
ویرایش اسقاط



User Flow
حذف اسقاط



User Flow
عملی تغییر دادن





:User Flow



User Flow



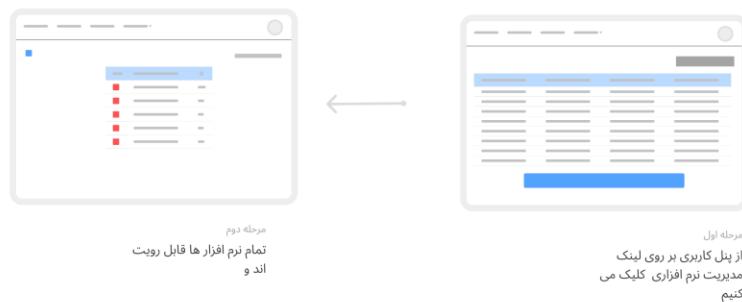
:User Flow



:User Flow
حذف ارسال سند به بیرون



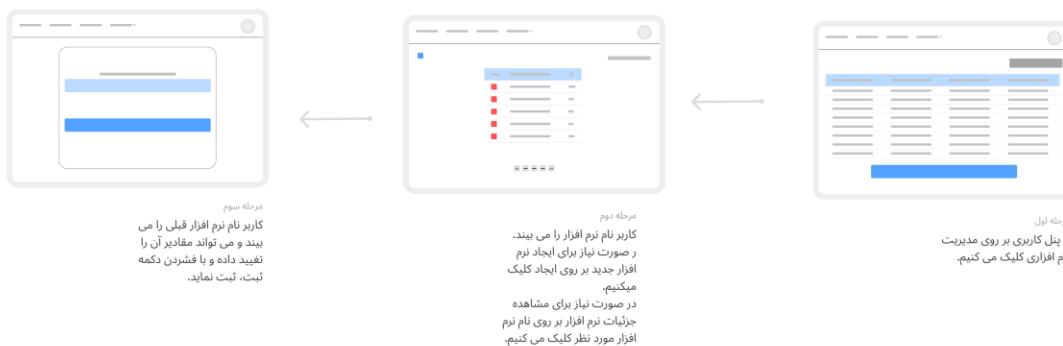
:User Flow
مشاهده همه نرم افزار ها



:User Flow
ایجاد نرم افزار

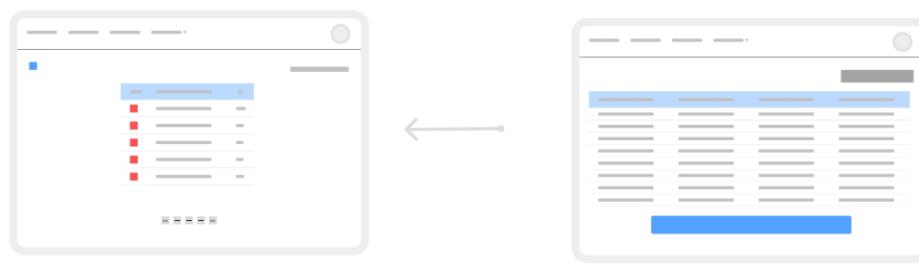


:User Flow
جزئیات نرم افزار و تغییرات آن



:User Flow

حذف نرم افزار



مکان ارسال پیغام ها به سمت شعبه فایده شعبه
پس از اینکه در اینجا نمودار را در خانه مطابق
فرموده باشیم

User Flow
اعمال نسبیات



-User Flow
ویرایش نسبیات



-User Flow
حذف نسبیات



-User Flow
اعمال بوزرسانی



:User Flow



:User Flow



:User Flow





:User Flow
حذف جا به جایی

