

سیستم مدیریت جای پارک  
**Park Management System**

اعضای گروه:

۹۳۱۳۲۳۱۱۶۷	سید حسین موسوی پور
۹۳۱۳۲۳۱۲۰۶	محسن حاجی محمدی
۹۳۱۳۲۳۱۱۵۱	پیمان آزاد

خرداد ۱۳۹۷

# فهرست مطالب

۱	معرفی سیستم.....
۱	سیستم مدیریت جای پارک چیست؟.....
۱	استفاده از سیستم مدیریت جای پارک چه مزایایی دارد؟.....
۲	چرا به یک سیستم مدیریت جای پارک نیاز داریم؟.....
۳	نیازمندی ها .....
۳	محدودیت های سیستم.....
۴	گزارشات امکان سنجی پروژه.....
۴	توصیف پروژه.....
۴	هدف.....
۴	ویژگی های کلی پروژه.....
۴	امکان سنجی پروژه.....
۵	منابع مورد نیاز پروژه.....
۵	منابع انسانی.....
۶	منابع محیطی.....
۷	معماری سیستم <b>PGS</b> .....
۷	زمان.....
۸	بودجه.....
۸	محدودیت های نرم افزار.....
۹	مدل های تحلیل سیستم.....
۹	مدل تحلیل <b>ER</b> .....
۱۰	توضیحات موجودیت های مدل تحلیل <b>ER</b> .....

۱۴	.....	<b>Data Flow Diagram</b>	نمودار های تحلیل
۲۳	.....	<b>STD Diagram</b>	مدل تحلیل
۲۴	.....		تبدیل مدل تحلیل به طراحی (ساختار سلسله مراتبی)
۲۵	.....		ماژول های سیستم مدیریت پارکینگ
۳۵	.....		توضیحات مربوط به هر یک از ماژول ها
۳۷	.....		روش های تست نرم افزار
۳۷	.....		تست جعبه سفید
۳۷	.....		رسم فلوگراف به همراه محاسبه وزن گراف و مسیر های مستقل
۴۴	.....		جداول تست جعبه سفید (تست کیس ها)
۴۹	.....		تست جعبه سیاه
۵۱	.....		نگهداری

### معرفی سیستم

فضای عمومی شهرها در جامعه امروزی ، بیش از هر زمانی نیاز به مدیریت دارد. تعداد بی شمار خودروها که به صورت روز افزون به تعداد آن ها افزوده می شود، مدیریت بدون کنترل و نظارت را بی معنا میسازد.

### سیستم مدیریت جای پارک چیست؟

به طور خلاصه سیستمی است که به شما کمک می کند در کمترین زمان ممکن مکان های قابل دسترس برای پارک را پیدا نموده، آن را برای بازه زمانی مشخصی از روز رزرو کرده و از آن استفاده نمایید.

### استفاده از سیستم مدیریت جای پارک چه مزایایی دارد؟

#### راننده ها

- پارک بدون دغدغه و دردسر
- کاهش زمان جستجو برای یافتن مکان خالی برای پارک کردن خودرو
- از بین رفتن استرس ناشی از ناکامی در پیدا کردن جای پارک
- حفظ زمان با ارزش به خصوص در فرودگاه ،مراکز خرید و ...
- کاهش مصرف سوخت
- اطلاع لحظه ای از موقعیت مکان های پارک یک منطقه

#### برای مالکان فضاهای پارک

- گردش سریع مکان های پارک در دسترس و افزایش درآمد
- کاهش نارضایتی مشتریان
- مدیریت جامع آمار
- مدیریت پربازده و کنترل آسان ماشین ها
- کاهش هزینه ها

## چرا به یک سیستم مدیریت جای پارک نیاز داریم؟

- پیدا کردن جای پارک در زمان شلوغی مشکل زا است ولی در این سیستم امکان بررسی مکان های خالی پارک در لحظه وجود دارد و مکان های دور افتاده تر و جاهایی که قابل رویت توسط کاربر نیستند را به او پیشنهاد می دهد.
- هیچ تضمینی در حالت عادی برای کابر وجود ندارد که بتواند در منطقه مورد نظر خود جای پارک خالی پیدا نماید، اما در سیستم مدیریت هوشمند جای پارک امکان رزرو و مشخص کردن جای پارک برای بازه زمانی مشخص برای کاربر وجود دارد.
- در سیستم قدیمی کاربر اجبارا باید کارت پارکی خریداری کرده و آن را هر دفعه در باجه های مخصوص خود شارژ نماید اما در این سیستم، رزرو و پرداخت برای جای پارک بدون نیاز به کارت پارک و تنها با استفاده از اینترنت (برای مثال تلفن همراه) امکان پذیر است.
- شناسایی ماشین های پارک شده غیرمجاز توسط مأمور سخت است اما در این سیستم از خود کاربران سیستم نیز علاوه بر مأمور برای شناسایی ماشین های متخلف استفاده میشود.
- امکان اطلاع از وضعیت جای پارک در یک زمان مشخص وجود ندارد اما در این سیستم کاربر میتواند وضعیت جای پارک را برای زمان های آینده نیز بررسی کرده و در صورت نیاز آن را برای زمان های آینده رزرو نماید.
- جای پارک در پارکینگ ها توسط مأمور پارکینگ مشخص می شود اما در این سیستم کاربر خود میتواند جای پارک خود را مشخص نماید.
- در سیستم قدیمی، قیمت پارک در همه بازه های زمانی و در موقعیت های مکانی در شهر ثابت است اما در این سیستم قیمت ها به صورت پویا تعیین می شود.
- سیستم قدیم فقط جای پارک را مشخص میکند و فاقد امکانات اضافی است. اما در این سیستم امکان عقد قرارداد با شرکت های خدماتی و ارائه خدمات آن ها به کاربران در طول زمان پارک می باشد (از جمله تنظیم، شستشو و...)
- امکان عقد قرارداد با سازمان ها و شرکت ها در جهت رزرو یک مکان در ساعات مشخص به صورت همه روزه وجود دارد.
- امکان استفاده از پارکینگ های شخصی برای ساعاتی که مورد استفاده قرار نمیگیرند، وجود دارد.

## نیازمندی ها

#	شرح نیازمندی
۱	رزرو جای پارک در بازه زمانی مشخص
۲	جستجوی مکان های پارک نزدیک بر اساس موقعیت کاربر
۳	بررسی وضعیت یک مکان پارک مشخص
۴	قیمت گذاری پویا بر اساس ساعت و موقعیت مکان رزرو
۵	آمارگیری از وضعیت های مکان های پارک در یک منطقه
۶	رزرو گروهی چند جای پارک برای سازمان ها و ارگان ها

## محدودیت های سیستم

- اگر برای دریافت جای پارک کاربر فقط محدود به سیستم مدیریت مدیریت جای پارک باشد:
- کاربران برای دریافت خدمات سیستم لزوما باید به اینترنت دسترسی داشته باشند.
- امکان دارد افرادی نتوانند با این سیستم ها کار کنند
- اگر سیستم مدیریت جای پارک به صورت همگام با سیستم های قدیمی کار کنند در مواردی باعث بروز ناسازگاری می شود.
- امکان پارک غیرمجاز در مکان های رزرو شده توسط کاربر توسط کاربران متخلف وجود دارد.

## گزارشات امکان سنجی پروژه

### توصیف پروژه

ایجاد نرم افزاری کاربردی در زمینه سیستم مدیریت پارکینگ

### هدف

طراحی سیستمی است که به شما کمک کند در کمترین زمان ممکن مکان های قابل دسترس برای پارک را پیدا نموده، آن را برای بازه زمانی مشخصی از روز رزرو کرده و از آن استفاده نمایید.

### ویژگی های کلی پروژه

مشاهده نزدیکترین مکان های پارک

قابلیت جستجوی جای پارک در تمام شهر و اطلاع از وضعیت رزرو

رزرو کردن برای مدت زمان مورد نظر

نمایش ظرفیت پارکینگ

### امکان سنجی پروژه

در این مطلب عوامل موثر بر پروژه به دقت مورد بررسی و کالبد شکافی قرار گرفته، نقاط قوت و ضعف آن بازنگری شده و در نهایت توجیه پذیری سرمایه گذاری، برآورد میگردد. این مطالعه در حکم گام نخست در فرآیند تصمیم گیری سرمایه گذاران یا مدیران سازمان محسوب میگردد.

### منابع مورد نیاز پروژه

۱. منابع انسانی (Resources Human)
۲. منابع محیطی (Environment Resources)
۳. منبع زمانی (Time)
۴. منبع مالی (Budget)

حال به تشریح منابع می پردازیم. در تشریح منابع چهار فاکتور باید بررسی شوند که عبارتند از:

۱. تشریح منبع
۲. در دسترس بودن و نحوه کسب منبع
۳. زمانی که منبع مورد نیاز میباشد و مدت زمانی که به آن نیاز است.
۴. منابع نرم افزاری با قابلیت استفاده مجدد

### منابع انسانی

۱. برنامه نویس وب و اندروید و iOS
- ۱،۱- از آنجایی که پلتفرم مد نظر پروژه، گوشی های می باشد، منبع انسانی اصلی مورد نیاز پروژه فردی متخصص در زمینه برنامه نویسی وب و موبایل است.
- ۱،۲- اگر در حال حاضر سازمان دارای برنامه نویس وب و اندروید و iOS می باشد میتوان به سراغ مراحل بعدی رفت. در غیر اینصورت با توجه ضرورت وجود برنامه نویس در روند توسعه پروژه، باید اقدام به جذب نیرویی کارآمد در این زمینه کرد. به عنوان مثال با درج آگهی در وب سایت هایی از قبیل [dehvand.ir](http://dehvand.ir) , [e-estekhdam.com](http://e-estekhdam.com) , [jobinja.ir](http://jobinja.ir) یا کانال های استخدام در تلگرام، نیروی مورد نیاز را جذب نمود.
- ۱،۳- پس از آنکه طراح طرح اولیه (wireframe) خود را برای پروژه ارائه داد، برنامه نویس میتواند شروع به کدنویسی پروژه نماید.
- ۱،۴- به طور کلی این محصول از قابلیت استفاده مجدد پشتیبانی می نماید.



۲. طراح

- ۲،۱- نیاز به یک متخصص ui/ux برای طراحی رابط کاربری و تجربه کاربری و همچنین لوگوی مناسب محصول وجود دارد.
- ۲،۲- قطعا وجود یک متخصص در این حوزه باعث ایجاد محصولی با کیفیت تر خواهد شد اما اگر سازمان فاقد چنین نیرویی است، برنامه نویس در صورت موافقت میتواند فعالیت مربوط به طراحی را نیز انجام دهد. در غیر اینصورت نیاز به استخدام یک متخصص در زمینه طراحی وجود دارد.
- ۲،۳- در روند توسعه نرم افزار، فاز طراحی قبل از فاز پیاده سازی صورت میگیرد. پس وجود طراح از ابتدای نهایی شدن پروژه برای توسعه، ضروری میباشد.

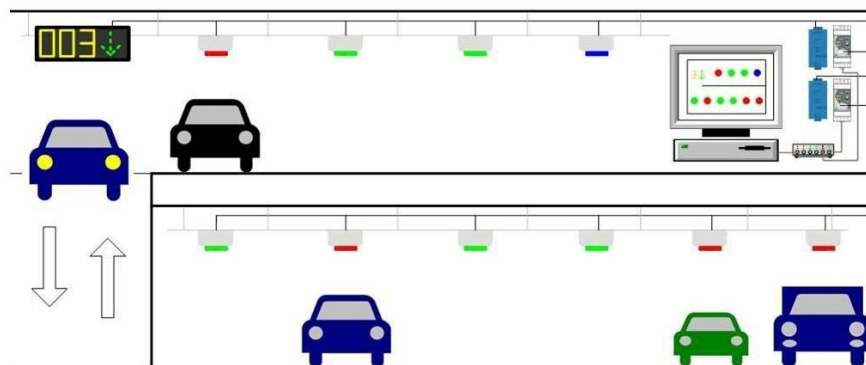
منابع محیطی

۱. بازار هدف محصول

- ۱،۱- پس از توسعه نرم افزار، از چه بسترهایی برای در اختیار قرار دادن محصول در دست کاربران نهایی میتوان استفاده نمود. که میتوان به عنوان نمونه از گوگل پلی، اپل استور، سیپچه، کافه بازار، ایران اپس و سایر مارکت های محبوب داخلی استفاده نمود.
- ۱،۲- این منبع پس از پایان توسعه نرم افزار مورد نیاز میباشد و تا زمانی که محصول قابلیت استفاده مجدد داشته باشد، نیاز به آن موجود می باشد.

۲. سیستم PGS

- این سیستم مکانهای پارک خالی را مشخص و راننده را در کمترین زمان ممکن به نزدیکترین مکان راهنمایی می کند. در هر مکان پارک یک سنسور ما فوق صوت جهت مشخص کردن وضعیت مکان و ارسال اطلاعات به لایه بالاتر موجود است، هنگامی که لایه بالاتر اطلاعات را دریافت کرد نشانه های مختلف را ارسال می کند.



## معماری سیستم PGS

### ۱. حسگر های محلی و چراغ های هدایتگر (Local Sensor & Indicator)

این مجموعه بالای هر کدام از محل پارک خودروها نصب میشود. این مجموعه توسط یک سنسور آلتراسونیک وضعیت جای پارک (وجود یا عدم وجود خودرو) شناسایی کرده و به صورت Stand Alone نسبت به تصمیم گیری رنگ چراغ اقدام می نماید. سپس نتیجه تصمیم گیری را با استفاده از زیر ساخت شبکه رادیویی (RF Link) ارسال میکند.

### ۲. کنترلر محلی (Zone Controller)

این قسمت وظیفه دریافت اطلاعات از هر کدام از حسگر های محلی، جمع آوری این اطلاعات و ارسال آنها به بخش کنترل و مدیریت را دارد. همچنین این بخش رابط ارسال فرامین بین بخش کنترل و مدیریت با حسگر های محلی را بر عهده دارد.

### ۳. سامانه پلاک خوان

همان گونه که می دانیم تصاویری که از پلاک خودرو ثبت می شوند، تنها قابلیت مشاهده را دارند و امکان جستجو و یا ذخیره سازی متن پلاک را ندارند، لذا جهت ثبت متن پلاک که بتوانیم آن را در پایگاه داده ذخیره کرده و روی آنها عملیات گزارش گیری، جستجو و ویرایش را انجام دهیم نیازمند یک نرم افزار هوشمند هستیم که تصاویر دریافتی از دوربین ثبت پلاک را پردازش کرده و متن پلاک را از روی آن استخراج نماید نرم افزار پلاک خوان در واقع یک نرم افزار هوشمند است که وظیفه آن شناسایی پلاک خودرو از روی تصویر دریافتی از دوربین های ثبت پلاک می باشد. در واقع ورودی این نرم افزار تصویر و خروجی آن متن پلاک و بریده پلاک می باشد.

## زمان

اگر مدت زمان اجرای پروژه بسیار طولانی باشد و نتواند اهداف مورد نظر را تأمین کند، انجام پروژه توجیه پذیر نیست و با شکست مواجه میشود. زمان لازم برای تکمیل پروژه نیاز به بررسی عوامل متعددی دارد زیرا اجرای هر پروژه فازهای مختلفی را شامل میشود و هر فاز گروه هایی از افراد توسعه دهنده را. اما در حالت کلی هر بخش زمان حدودی ای که نیاز دارد تا وظیفه خود را در روند توسعه محصول به درستی انجام دهد را میداند. پس با مشورت با اعضای تیم توسعه میتوان به زمانی منطقی رسید و آنرا با مشتری در میان گذاشت.

### بودجه

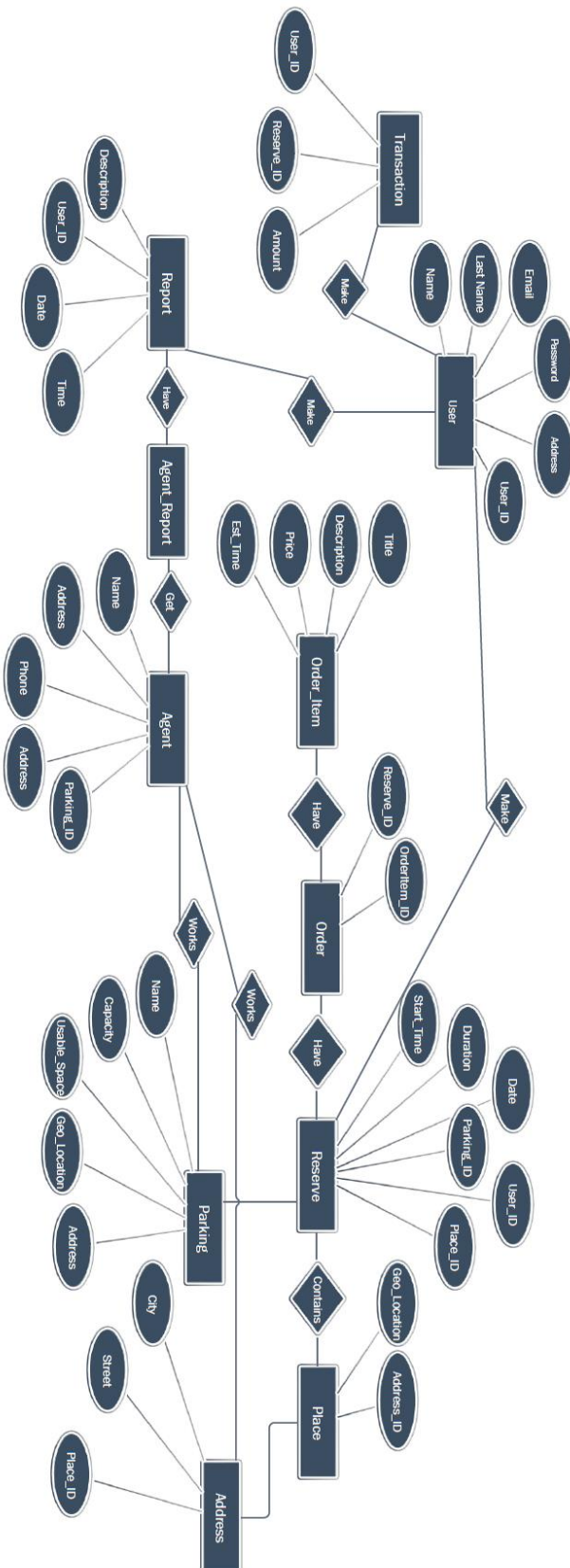
از آنجایی که به کار گیری افراد متخصص در روند توسعه نرم افزار نیازمند هزینه های خاص خود میباشد، با بررسی حدودی این هزینه ها و زمانی که قرار است برای این محصول صرف شود و سایر مسایل مالی، پروژه از این بابت مورد بررسی قرار میگیرد که آیا درآمد حاصله پس از اتمام پروژه نسبت به هزینه برای تولید آن به صرفه می باشد یا خیر. در صورتی که صرفه مالی وجود نداشته باشد انجام پروژه صورت نخواهد پذیرفت.

### محدودیت های نرم افزار

برای استفاده از نرم افزار حتما نیاز به دسترسی به تلفن هوشمند و یا کامپیوتر وجود دارد.  
برای استفاده از نرم افزار بهتر است دستگاه از تکنولوژی GPS بهره مند باشد.  
برای استفاده از نرم افزار باید دستگاه قابلیت اتصال به اینترنت را داشته باشد.  
برای استفاده از نرم افزار از سیستم PGS و سیستم پلاک خوان باید بخش های سخت افزاری پیاده سازی شوند.  
دریافت مجوزهای مورد نیاز برای نصب سخت افزارها در سطح شهر.

## مدل های تحلیل سیستم

### مدل تحلیل ER



## توضیحات موجودیت های مدل تحلیل ER

برنامه: سیستم مدیریت پارک موجودیت User نسخه: ۱,۰,۰ تاریخ: ۹۷/۲/۷				
شرح: کاربر شخصی است که از برنامه برای پیدا کردن جای پارک استفاده میکند.				
کلید	صفت	نوع	اجباری	توضیح
**	Use_Id	Numeric	Y	واحدی یکتا برای هر کاربر
	Name	Text	Y	نام کاربر
	Last Name	Text	Y	نام خانوادگی کاربر
	Password	Text	Y	رمز کاربر برای ورود به نرم افزار
	Address	Text	N	آدرس کاربر
	Email	Text	N	ایمیل کاربر

برنامه: سیستم مدیریت پارک موجودیت Transaction نسخه: ۱,۰,۰ تاریخ: ۹۷/۲/۷				
شرح: تراکنش عملی است که در آن کاربر برای یک رزرو مقداری هزینه پرداخت میکند.				
کلید	صفت	نوع	اجباری	توضیح
*	User_Id	Numeric	Y	واحدی یکتا برای هر کاربر و کلید خارجی
*	Reserve_Id	Numeric	Y	واحدی یکتا برای هر رزرو و کلید خارجی
	Amount	Numeric	Y	میزان هزینه پرداختی
**	Transaction_ID	Numeric	Y	واحدی یکتا برای هر تراکنش

برنامه: سیستم مدیریت پارک      موجودیت Report      نسخه: ۱,۰,۰      تاریخ: ۹۷/۲/۷				
شرح: گزارش کاربر در مورد محل پارک ، کیفیت برنامه، خدمات برنامه و ... است.				
کلید	صفت	نوع	اجباری	توضیح
*	User_Id	Numeric	Y	واحدی یکتا برای هر کاربر و کلید خارجی
	Description	Text	Y	توضیحات گزارش
	Date	Date	Y	روز گزارش
	Time	Time	Y	ساعت گزارش
**	Report_ID	Numeric	Y	واحدی یکتا برای هر گزارش

برنامه: سیستم مدیریت پارک      موجودیت Reserve      نسخه: ۱,۰,۰      تاریخ: ۹۷/۲/۷				
شرح: رزرو به انتخاب یک پارکینگ یا محل پارک برای مدتی مشخص است.				
کلید	صفت	نوع	اجباری	توضیح
**	Reserve_Id	Numeric	Y	واحدی یکتا برای هر رزرو
*	User_Id	Numeric	Y	واحدی یکتا برای هر کاربر و کلید خارجی
*	Place_Id	Numeric	N	واحدی یکتا برای هر مکان و کلید خارجی
*	Parking_ID	Numeric	N	واحدی یکتا برای پارکینگ و کلید خارجی
	Date	Date	Y	روزی که رزرو انجام شود.
	Duration	Time	Y	مدت زمان ک جایه پارک رزرو شده است.
	Start_Time	Time	Y	زمانی ک کاربر میخواهد پارک کند.

برنامه: سیستم مدیریت پارک موجودیت Order_Item نسخه: ۱,۰,۰ تاریخ: ۹۷/۲/۷				
شرح: آیتم ها به خدماتی گفته می شود که به صورت اضافی کاربران از آن ها می توانند استفاده کنند.				
کلید	صفت	نوع	اجباری	توضیح
**	Order_Item_ID	Numeric	Y	واحدی یکتا برای هر آیتم درخواستی
	Title	Text	Y	عنوان آیتم
	Description	Text	Y	توضیحات هر آیتم
	Price	Numeric	Y	قیمت هر آیتم
	Est_Time	Time	Y	

برنامه: سیستم مدیریت پارک موجودیت Agent نسخه: ۱,۰,۰ تاریخ: ۹۷/۲/۷				
شرح:				
کلید	صفت	نوع	اجباری	توضیح
**	Agent_ID	Numeric	Y	واحدی یکتا برای هر مامور
*	Report_ID	Numeric	Y	واحدی یکتا برای هر گزارش و کلید خارجی
*	Parking_ID	Numeric	Y	واحدی یکتا برای پارکینگ و کلید خارجی
	Name	Text	Y	نام مامور
	LastName	Text	Y	نام خانوادگی مامور
	Address	Text	Y	آدری هر مامور
	Phone	Text	Y	

برنامه: سیستم مدیریت پارک موجودیت Parking نسخه: ۱,۰,۰ تاریخ: ۹۷/۲/۷				
شرح: مشخصات و مکان پارکینگ را در اختیار کاربر قرار می دهد.				
کلید	صفت	نوع	اجباری	توضیح
**	Parking_Id	Numeric	Y	واحدی یکتا برای هر پارکینگ
	Name	Text	Y	اسم پارکینگ
	Capacity	Numeric	Y	گنجایش پارکینگ
	Address	Text	Y	آدرس پارکینگ
	Geo_Location	Location	Y	آدرس جغرافیایی
	Usable_Space	Numeric	Y	مقدار فضای قابل استفاده

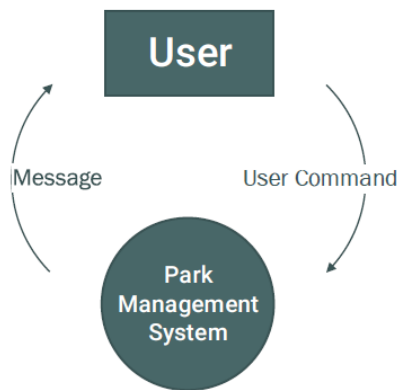
برنامه: سیستم مدیریت پارک place موجودیت نسخه: ۱,۰,۰ تاریخ: ۹۷/۲/۷				
شرح: مکان هایی که به عنوان جایه پارک می توان استفاده کرد مثل پارکینگ های خانگی در ساعاتی که استفاده ندارند یا کارواش هایی که در بعضی ساعات کار نمی کنند و ...				
کلید	صفت	نوع	اجباری	توضیح
**	Place_Id	Numeric	Y	واحدی یکتا برای هر مکان
*	Address	Numeric	Y	صفت مرکب است از آدرس محل پارک ها.
	Geo_Location	Location	Y	آدرس جغرافیایی



## نمودار های تحلیل Data Flow Diagram

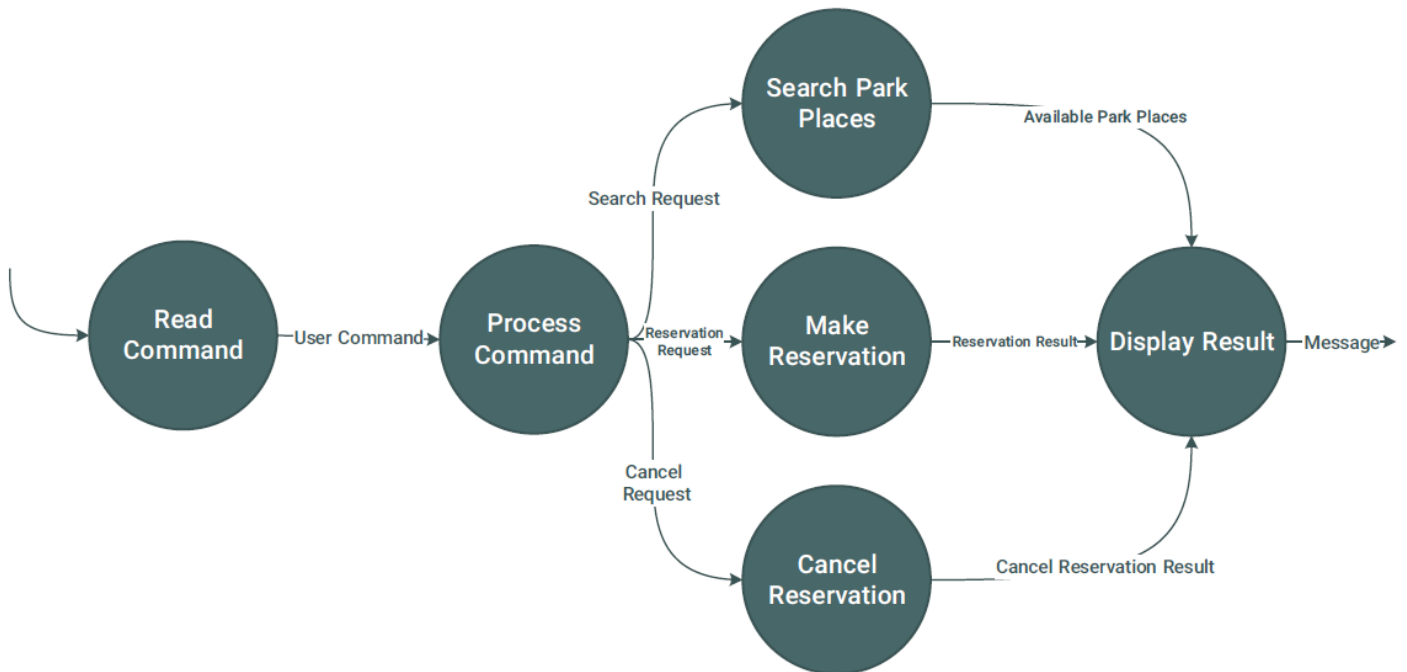
### سطح صفر Dataflow Diagram

نام تابع	Park Management System
توضیحات	تابع سیستم مدیریت جای پارک است که فرمان کاربر که شامل جستجوی جای پارک و رزرو محل پارک و یا لغو رزرو آن است را گرفته و نتیجه عملیات را به کاربر بر می گرداند.
ورودی ها	User Command
خروجی ها	Message



### سطح صفر از نمودار جریان داده

تابع گسترش داده شده Park Management System



سطح یک از نمودار جریان داده

نام تابع	Read Command
توضیحات	خواندن دستور کاربر
ورودی ها	
خروجی ها	User Command

نام تابع	Process Command
توضیحات	پردازش دستور داده شده توسط کاربر
ورودی ها	User Command
خروجی ها	Search Request, Reservation Request, Cancel Request

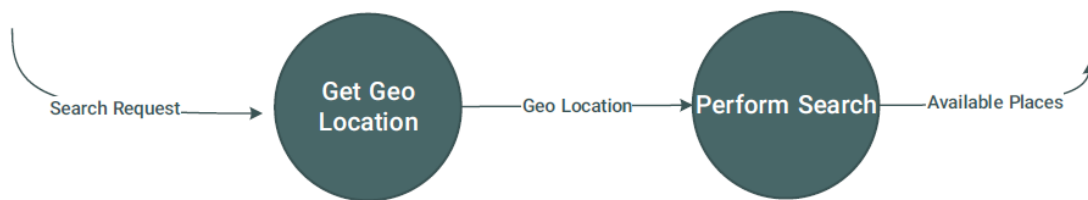
نام تابع	Search Park Places
توضیحات	جستجوی محل های پارک قابل رزرو اطراف کاربر
ورودی ها	Search Request
خروجی ها	Available Park Places

نام تابع	Make Reservation
توضیحات	انجام عملیات رزرو جای پارک
ورودی ها	Reservation Request
خروجی ها	Reservation Result

نام تابع	Cancel Reservation
توضیحات	انجام عملیات لغو رزرو جای پارک و مرجوع کردن هزینه توسط سیستم به کاربر
ورودی ها	Cancel Reservation Request
خروجی ها	Cancel Reservation Result

نام تابع	Display Result
توضیحات	نمایش نتایج عملیات های درخواست شده توسط کاربر
ورودی ها	/ Reservation Result / Available Park Places Cancel Reservation Result
خروجی ها	Message

تابع گسترش داده شده Search Park Places

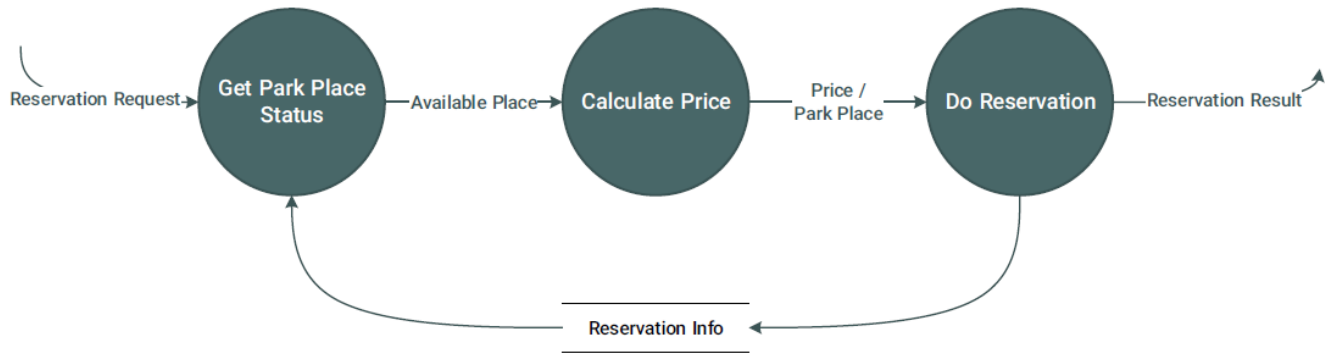


سطح دو از نمودار جریان داده

نام تابع	Get Geo Location
توضیحات	دریافت مختصات جغرافیایی کاربر
ورودی ها	Search Request
خروجی ها	Geo Location

نام تابع	Perform Search
توضیحات	یافتن محل های قابل رزرو پارک نزدیک
ورودی ها	Geo Location
خروجی ها	Available Places

تابع گسترش داده شده Make Reservation



سطح سه از نمودار جریان داده

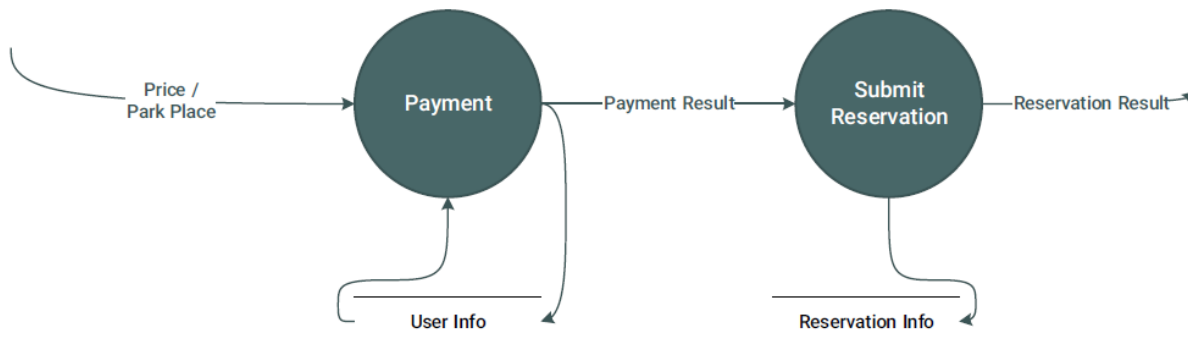
## سطح سه Dataflow Diagram

نام تابع	Get Park Place Status
توضیحات	بررسی قابل رزرو بودن جای پارک در زمان مورد نظر کاربر
ورودی ها	Reservation Request
خروجی ها	Available Place

نام تابع	Calculate Price
توضیحات	محاسبه هزینه رزرو مکان مورد نظر کاربر در زمان مشخص شده
ورودی ها	Available Place
خروجی ها	Park Place Price /

نام تابع	Do Reservation
توضیحات	انجام رزرو جای پارک و ثبت اطلاعات مربوط به رزرو کاربر در پایگاه داده
ورودی ها	Park Place Price /
خروجی ها	Reservation Result

تابع گسترش داده شده Do Reservation

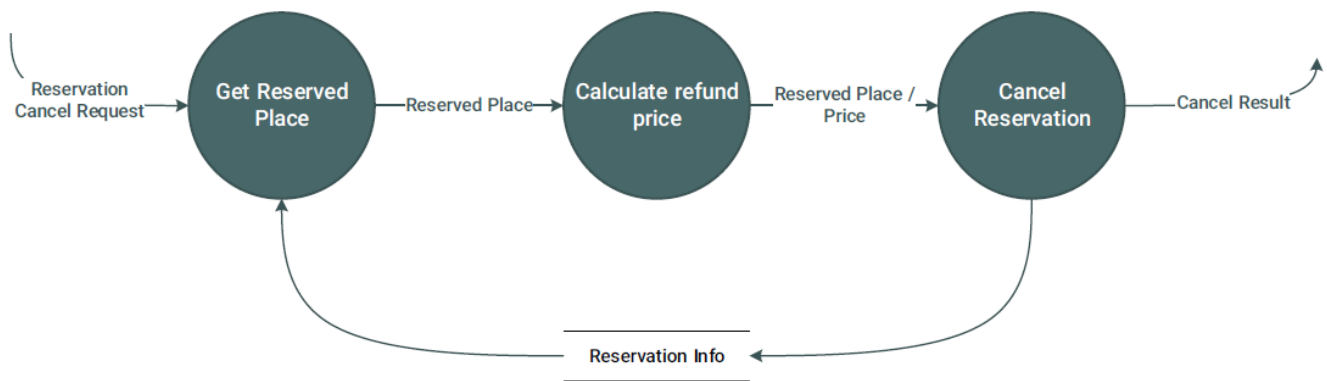


### سطح چهار از نمودار جریان داده

نام تابع	Payment
توضیحات	پرداخت مبلغ رزرو جای پارک توسط کاربر و ثبت اطلاعات پرداخت
ورودی ها	Price / Park Place
خروجی ها	Payment Result

نام تابع	Submit Reservation
توضیحات	انجام رزرو جای پارک و ثبت رزرو در پایگاه داده
ورودی ها	Payment Result
خروجی ها	Reservation Result

تابع گسترش داده شده Cancel Reservation



سطح پنج از نمودار جریان داده



## سطح پنج Dataflow Diagram

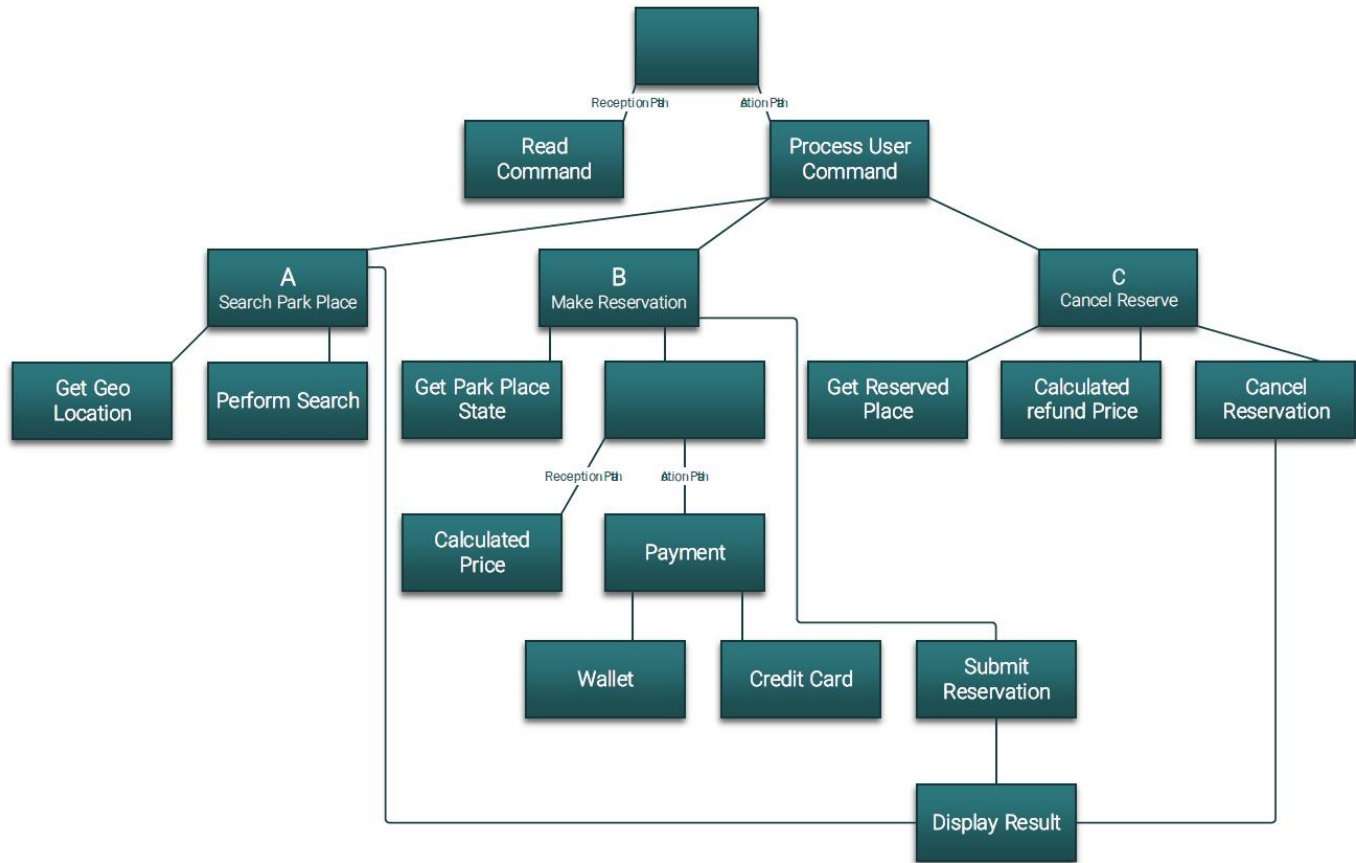
نام تابع	Get Reserved Place
توضیحات	بررسی اینکه آیا مکان توسط کاربر رزرو شده و آیا رزرو او قابل کنسل کردن می باشد یا خیر؟
ورودی ها	Reservation Cancel Request
خروجی ها	Reserved Place

نام تابع	Calculate refund price
توضیحات	محاسبه هزینه مرجوعی برای کنسل کردن رزرو کاربر
ورودی ها	Reserved Place
خروجی ها	Reserved Place / Price

نام تابع	Cancel Reservation
توضیحات	لغو عملیات رزرو و بازگشت وجه به کاربر و ثبت اطلاعات در پایگاه داده
ورودی ها	Reserved Place / Price
خروجی ها	Cancel Result



تبدیل مدل تحلیل به طراحی (ساختار سلسله مراتبی)



## ماژول های سیستم مدیریت پارکینگ

```
module Authorization
    procedure Login(email, pass)
        begin
            declare i = 0;
            declare count = 5;
            declare endBanTime;
            declare userInfo;

            if(email.isEmpty() OR pass.isEmpty()) then
                return errorMessage("Fields Cannot be empty");
            else if(i >= count) then
                begin
                    if(endBanTime.isEmpty()) then
                        endBanTime = time() + 5000;
                    else if(endBanTime < time()) then
                        i = 0;
                        break();
                    end if

                    return errorMessage("You can't login for a while");
                end
            else
                begin
                    userInfo = Database.getUser(email);

                    if(userInfo.isEmpty() OR userInfo.pass != pass) then
```

```

begin
    14
    i++;
    return errorMessage("Email or password is incorrect");
end
else
begin
    15
    Session.Set(email);
    return welcomeMessage("Hello" + userInfo.name + "!");
end
    16
end if
end
    17
end if
end procedure

procedure Register(email, pass, confirmPass)
begin
    1          2          3
    if(email.isEmpty() OR pass.isEmpty() OR confirmPass.isEmpty()) then
        4
        return errorMessage("Fields Cannot be empty");
    5
    else if(pass != confirmPass) then
        6
        return errorMessage("Passwords are not the same");
    7
    else if(Database.getUser(email).notEmpty()) then
        8
        return errorMessage("Email exist in database");

    else
    begin

```

9

```
Database.AddToUserTable(email, pass);
```

```
Session.Set(email);
```

```
return welcomeMessage("welcome to our service! Sign Up was successful");
```

```
end
```

```
10
```

```
end if
```

```
end
```

```
end module
```

```
module Reservation
```

```
procedure Search(location, time)
```

```
begin
```

```
1
```

```
declare searchList;
```

```
declare availablePlaces;
```

```
searchList = Database.getNearLocations(location);
```

```
2
```

```
for each(place in searchList)
```

```
3
```

```
if(Database.checkPlaceAvailableNow(place, time)) then
```

```
4
```

```
availablePlaces.Add(place);
```

```
5
```

```
end if
```

```
6
```

```
end for each
```

```
7
```

```
return availablePlaces;
```

```
end
```

```
end procedure
```

```

procedure GetServiceInfo(place, startTime, endTime)
begin
    1
    declare price;
    2
    if(Database.checkPlaceAvailable(place, startTime, endTime)) then
    3
        availablePlaces.Add(place);
        price = calculatePrice(place, startTime, endTime);
        return array(place, price);
    else
    4
        return false;
    5
    end if
end
end procedure

```

```

procedure MakeReservation(place, startTime, endTime)
begin
    1
    declare parkService;
    declare refId;
    2
    if(GetServiceInfo(place, startTime, endTime)) then
        begin
            3
            parkService = GetServiceInfo(place, startTime, endTime);
            refId = Payment.StartPay(parkService.price);
            4
            if(Payment.Verify(refId)) then
            5

```

```
        Database.AddReservation(place, startTime, endTime);
    else
        6
        return errorMessage("error in payment");
    7
    end if
end
else
    8
    return errorMessage("Park Place is not available in your requested time
range");

    9
end if

end
end procedure

procedure CancelReservation(email, place)
begin
    1
    declare time;

    2
    if(User.OrderDetails(email, place).startTime > time()) then
        begin
            3
            Payment.Refund(place, email);
            Database.UpdateRecord("Reserve Status = Refunded");
            return successMessage("Cancel Reservation was successful");
        end
    else
        4
        return errorMessage("You can't cancel this reservation");
    end
end
```



```
        5
    end if

end

end procedure

end module

module Payment

    procedure StartPay(price)
        begin
            1
            declare method = ReadUserSelectedPaymentMethod();
            declare refId;
            2
            switch(method)
            begin
            3
                case "wallet":
                    refId = payBywallet(price);
                end case
            4
                case "creditCard":
                    refId = payByCreditCard(price);
                end case
            5
            end
            6
            return refId;

        end
    end procedure

end module
```

```
procedure PayByWallet(price)
begin
    1
    declare userBalance;
    declare newBalance;
    declare refId;
    userBalance = User.GetCredit(session(email));

    2
    if(userBalance > price) then

        begin

            3
            newBalance = userBalance - price;

            Database.UpdateRecord("User Balance = newBalance")->where("User",
"=", session(user));

            $refId = Database.AddPayment(session(user), price);

            return refId;

        end

    else

        4
        return errorMessage("Balance is not enough");

    5
    end if

end
end procedure
```

```
procedure PayByCreditCard(price)
```

```
begin
```

```
    1
```

```
        declare refId;
```

```
        RedirectUserToBank();
```

```
        $refId = GetCallbackFromBank();
```

```
        return refId;
```

```
end
```

```
end procedure
```

```
procedure Refund(place, email)
```

```
begin
```

```
    1
```

```
        declare userBalance;
```

```
        declare newBalance;
```

```
        declare refundAmount;
```

```
        refundAmount = CalculateRefundAmount(email, place);
```

```
        userBalance = User.GetCredit(session(email));
```

```
        newBalance = userBalance + refundAmount;
```

```
        Database.UpdateRecord("UserBalance = newBalance")->where("user", "=", email);
```

```
end
```

```
end procedure
```

```
end module
```

```
module User
```

```
    procedure GetCredit(email)
```

```
    begin
```

```
        declare credit;
```

```
        Database.ReadRecord("Balance")->where("user", "=", email);
```

```
        return credit;
```

```
    end
```

```
end procedure
```

```
    procedure OrderDetails(email, place)
```

```
    begin
```

```
        declare info;
```

```
        info.startTime = Database.ReadRecord("startTime")->where("email", "=", email)->where("place", "=", place);
```

```
        info.endTime = Database.ReadRecord("endTime")->where("email", "=", email)->where("place", "=", place);
```

```
        info.price = Database.ReadRecord("price")->where("email", "=", email)->where("place", "=", place);
```

```
        return info;
```

```
    end
```

```
end procedure
```

```
    procedure EditProfile(newName, newLastName, newPhone, email)
```

```
    begin
```

```
        1
```

```
        declare result;
```

```
        result = Database.UpdateRecord("name = newName, lastName = newLastName, phone = newPhone")->where("user", "=", email);
```

```
        2
```

```
        if(result) then
            3
            return successMessage("Update was successful");
        else
            4
            return errorMessage("Error Occurred in updating profile, try again
later");
        5
        end if

    end
end procedure

procedure PostReport(email, report)
    begin
        1
        declare result;
        2
        result = Database.AddRecord(email, report);
        3
        if(result) then
            4
            return successMessage("Report submit was successful");
        else
            5
            return errorMessage("Error Occurred in submitting report, try
again later");
        end if
    end
end procedure

end module
```

## توضیحات مربوط به هر یک از ماژول ها

## ماژول Authorization

این ماژول شامل توابع احراز هویت است. عملیات هایی مانند ورود به سیستم و ثبت نام در سیستم توسط این ماژول انجام می شود.

تابع Login	این تابع ایمیل و کلمه عبور را از کاربر به عنوان ورودی گرفته و در صورت صحیح بودن اطلاعات وارد شده، اجازه ورود را به کاربر می دهد. در صورت خالی گذاشتن هر یک از پارامتر های ورودی توسط کاربر، سیستم پیغام خطا به کاربر می گرداند. همچنین در صورتی که تعداد دفعات ورود ناموفق توسط کاربر از تعداد مشخصی بیشتر شود، کاربر برای مدت مشخصی از سایت اخراج می شود.
تابع Register	این تابع ایمیل و کلمه عبور و تکرار کلمه عبور وارد شده توسط کاربر را به عنوان ورودی گرفته و در صورتی که کلمه عبور وارد شده با تکرار کلمه عبور تطابق داشته باشد و ایمیل کاربر از قبل در پایگاه داده وجود نداشته باشد، ثبت نام کاربر را انجام داده و رکورد مربوط به کاربر را در پایگاه داده وارد می کند. در غیر اینصورت به کاربر پیغام خطای مناسب را نمایش می دهد.

## ماژول Reservation

در این ماژول توابع مربوط به رزرواسیون قرار دارد که کاربر می تواند مکان های پارک نزدیک موقعیت مورد نظرش را جستجو کرده و اطلاعات مربوط به یک مکان پارک را دریافت نماید و رزرو را انجام داده و یا آن را لغو نماید.

تابع Search	این تابع موقعیت کاربر و زمان فعلی را به عنوان پارامتر های ورودی گرفته و به جستجوی محل های پارک نزدیک می پردازد. سپس محل های پارک نزدیک را که قابل رزرو هستند را در لیستی اضافه کرده و به کاربر بر می گرداند.
تابع GetServiceInfo	این تابع مکان پارک مورد نظر کاربر و زمان شروع و پایان رزرو مورد درخواست کاربر را به عنوان ورودی گرفته و در صورتی که مکان پارک طبق شرایط داده شده قابل رزرو باشد، قیمت رزرو برگردانده می شود در غیر اینصورت false برگردانده می شود.
تابع MakeReservation	این تابع مکان پارک مورد نظر کاربر و زمان شروع و پایان رزرو مورد درخواست کاربر را به عنوان ورودی گرفته و سپس از طریق احضار تابع GetServiceInfo اقدام به دریافت قیمت رزرو مکان مورد نظر می پردازد. در صورتی که توسط این تابع قیمت برگردانده شد تابع StartPay در ماژول Payment فراخوانی شده و در صورتی که پرداخت کاربر تایید شد رزرو انجام شده و رکورد در دیتابیس اضافه می شود. در غیر اینصورت پیغام خطای مناسب به کاربر نمایش داده می شود.

این تابع ایمیل کاربر و مکان پارک رزرو شده توسط او را به عنوان ورودی گرفته و در صورتی که زمان شروع رزرو از زمان فعلی بیشتر باشد تابع Refund در ماژول Payment فراخوانی شده و پس از بازگشت وجه به کاربر رکورد رزرو در دیتابیس بروز رسانی شده و وضعیت رزرو به Refunded تغییر می کند. در غیر اینصورت پیغام خطای مناسب به کاربر نمایش داده می شود.	تابع <b>CancelReservation</b>
--	----------------------------------

ماژول Payment	
توابع مربوط به پرداخت و بازگشت وجه در این ماژول قرار دارد که پرداخت می تواند از طریق کیف پول و یا کارت بانکی مشتری انجام شود.	
این تابع قیمت را از سیستم به عنوان پارامتر ورودی گرفته و تابع PayByWallet و یا PayByCreditCard متناسب با درخواست کاربر فراخوانی می کند.	تابع StartPay
این تابع مبلغ پرداخت را به عنوان ورودی از سیستم گرفته و در صورتی که موجودی کیف پول کاربر از این مبلغ بیشتر باشد، مبلغ از کیف پول کاربر کسر شده و شماره ارجاع پرداخت در دیتابیس برگردانده می شود در غیر اینصورت پیغام خطای مناسب به کاربر نمایش داده می شود.	تابع PayByWallet
این تابع مبلغ پرداخت را به عنوان ورودی از سیستم گرفته و کاربر را به دروازه بانک انتقال می دهد و پس از بازگشت از بانک شماره ارجاع را بر می گرداند.	تابع PayByCreditCard
این تابع مبلغ و ایمیل کاربر را به عنوان ورودی گرفته و مبلغ را به موجودی کیف پول کاربر اضافه می کند.	تابع Refund

ماژول User	
توابع مربوط به کاربر در این ماژول قرار دارد.	
این تابع ایمیل کاربر را به عنوان ورودی از سیستم گرفته و موجودی کاربر را بر می گرداند.	تابع GetCredit
این تابع ایمیل کاربر و مکان رزرو شده توسط او را از به عنوان ورودی از سیستم گرفته و جزئیات رزرو را بر می گرداند.	تابع OrderDetails
این تابع ایمیل کاربر به همراه نام و یا نام خانوادگی و شماره تماس را به عنوان ورودی گرفته و رکورد کاربر را بروزرسانی می کند.	تابع EditProfile
این تابع ایمیل و گزارش کاربر را به عنوان ورودی گرفته و آن ها را در سیستم ثبت می کند و در صورت ثبت موفقیت آمیز پیغام مناسب را به کاربر نمایش می دهد.	تابع PostReport

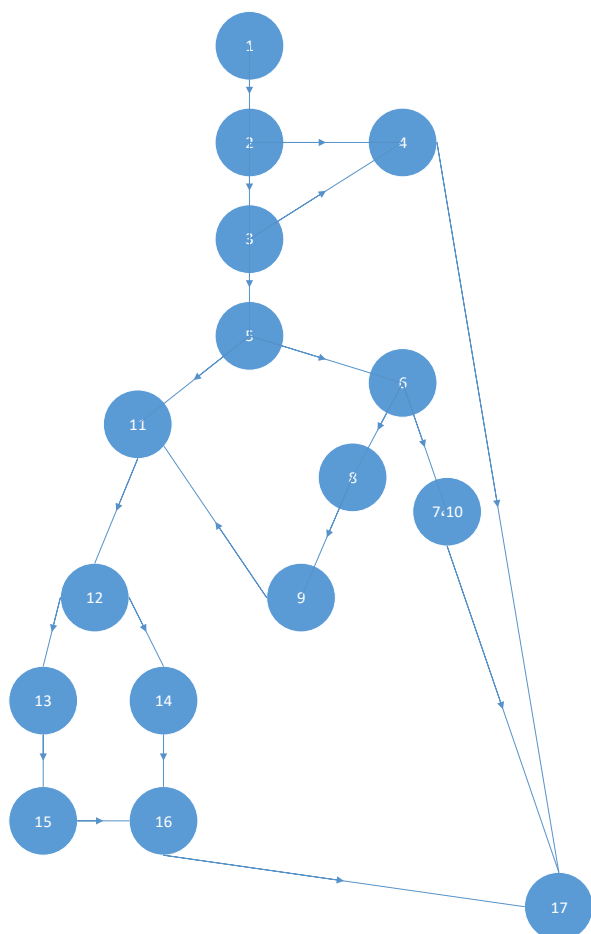
## روش های تست نرم افزار

## تست جعبه سفید

تست از همان ابتدای پروژه شروع می شود. بدین ترتیب که از فاز تحلیل، استراتژی تست تعیین می شود. بر اساس این استراتژی برنامه های تست (Test plan) اولیه طراحی شده و انواع تست مشخص می شود. سپس تست آغاز شده و تا هنگامی که برنامه در محیط مشتری استقرار یابد، ادامه پیدا می کند.

## رسم فلوگراف به همراه محاسبه وزن گراف و مسیر های مستقل

فلوگراف مربوط به تابع Login



$$V(G) = 20 - 16 + 2 = 6$$

1,2,4,17

1,2,3,4,17

1,2,3,5,6,7,10,17

...,1,2,3,5,6,8,9,11

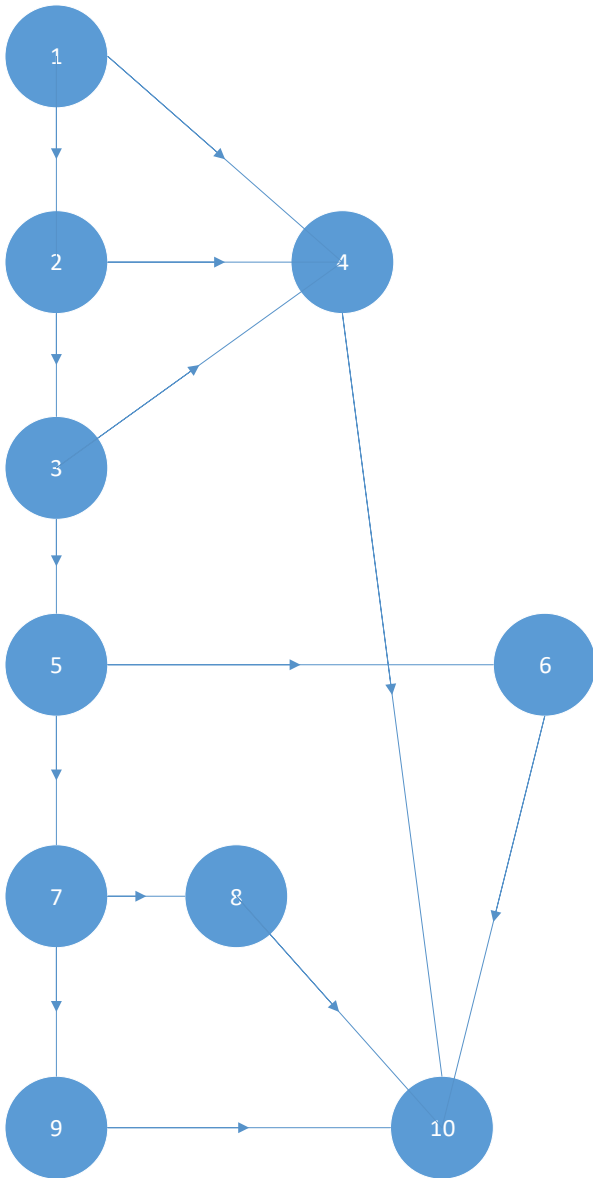
1,2,3,5,11,12,14,10,17

1,2,3,5,11,12,13,14,16,17

1,2,3,5,11,12,13,15,16,17



فلوگراف مربوط به تابع Register



$$V(G) = 14 - 10 + 2 = 6$$

1,4,10

1,2,4,10

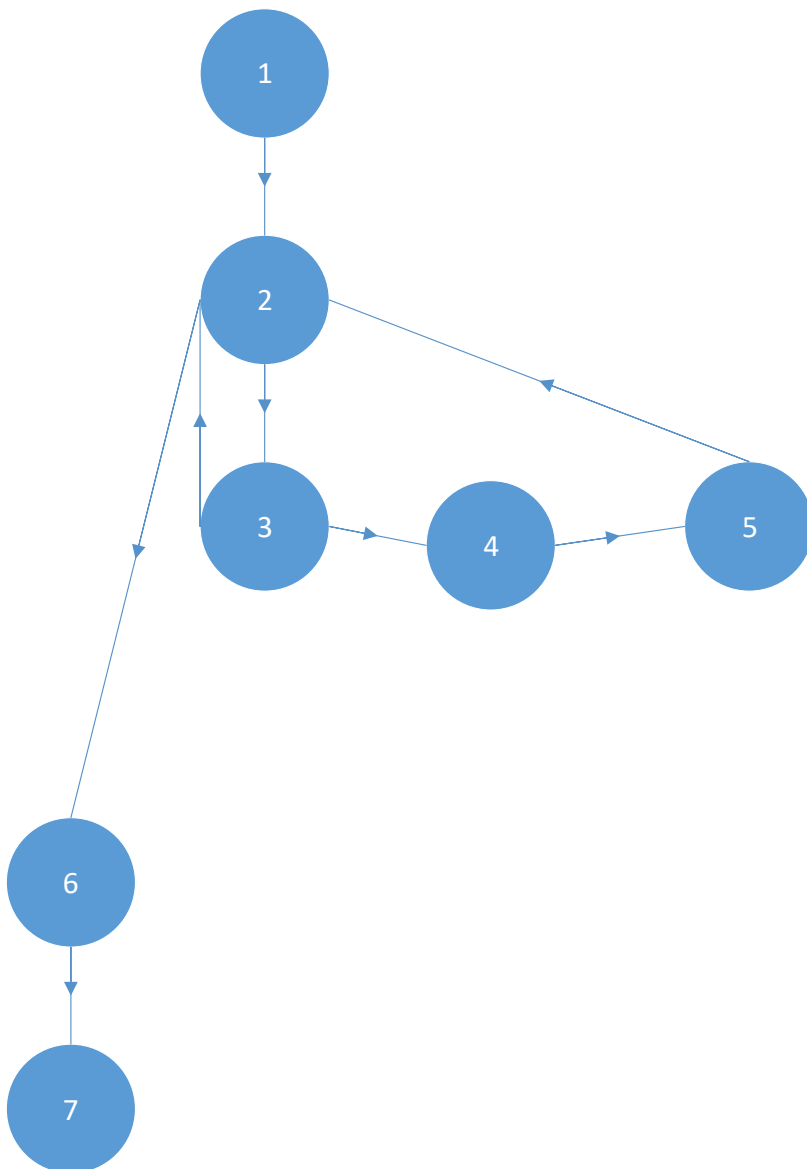
1,2,3,4,10

1,2,3,5,6,10

1,2,3,5,7,8,10

1,2,3,5,7,9,10

فلوگراف مربوط به تابع Reservation-Search



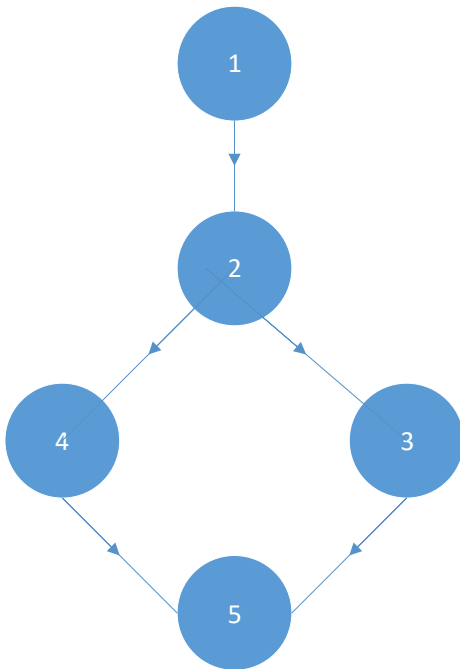
$$V(G)=8-7+2=3$$

۱,۲,۶,۷

...,۱,۲,۳,۲

...,۱,۲,۳,۴

فلوگراف مربوط به تابع `GetServiceInfo`

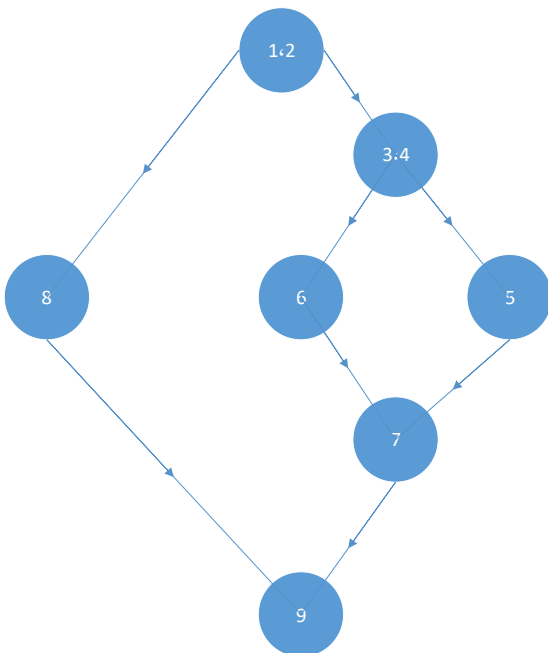


$$V(G)=5-5+2=2$$

1,2,3,5

1,2,4,5

فلوگراف مربوط به تابع `MakeReservation`



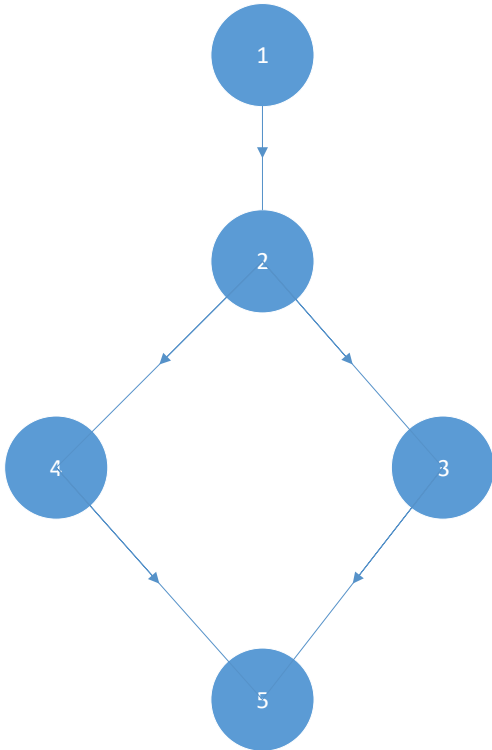
$$V(G)=8-7+2=3$$

1,2,3,4,5,7,9

1,2,3,4,6,7,9

1,2,8,9

فلوگراف مربوط به تابع CancelReservation

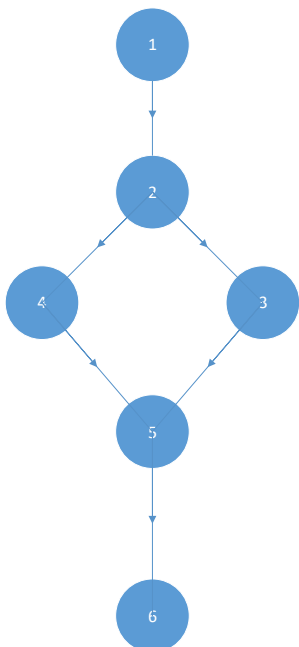


$$V(G)=2$$

۱,۲,۳,۵

۱,۲,۴,۵

فلوگراف مربوط به تابع StartPay

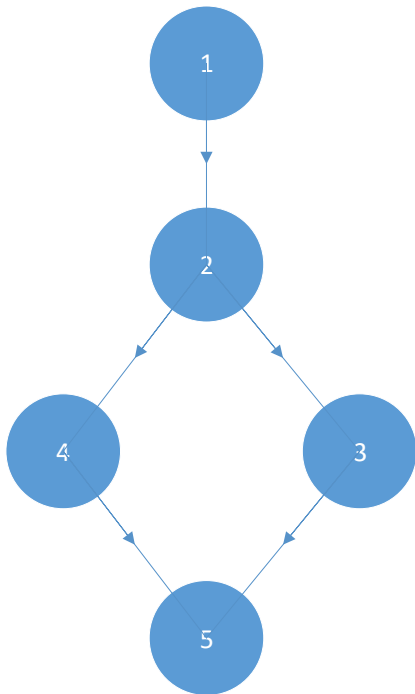


$$V(G)=2$$

1,2,3,5,6

1,2,4,5,6

فلوگراف مربوط به تابع PayByWallet

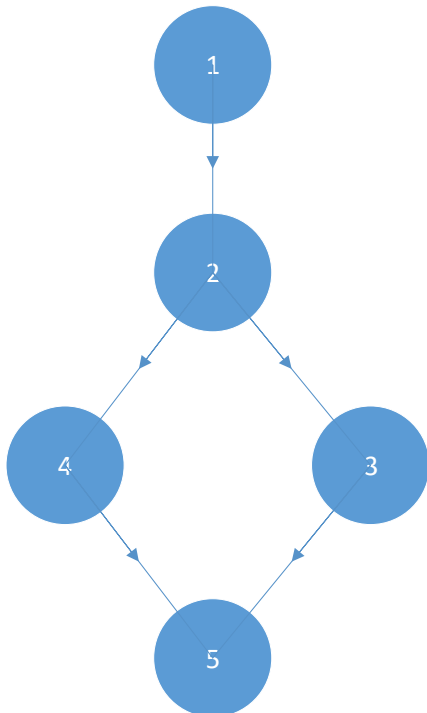


$V(G)=2$

1,2,3,5

1,2,4,5

فلوگراف مربوط به تابع EditProfile

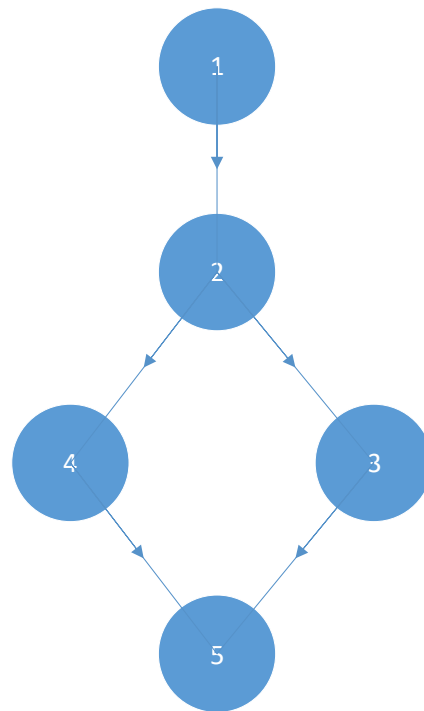


$V(G)=2$

1,2,3,5

1,2,4,5

فلوگراف مربوط به تابع PostReport



$$V(G) = 2$$

1,2,3,5

1,2,4,5

## جداول تست جعبه سفید (تست کیس ها)

## Login

TC#	توصیف مسیر	پیش نیاز	ورودی	خروجی مورد انتظار	خروجی واقعی	Pass Fail
۱	وضعیت ورود در صورت خالی گذاشتن فیلدهای ایمیل یا پسورد	۴	ورودی توسط کاربر وارد نشود	پیغام خطای ورود	خطای ورود به حساب کاربری	✓
۲	اگر تعداد وارد کردن نادرست ایمیل و کلمه عبور بیش از مقدار مجاز باشد	۶،۵	Email password	عدم اجازه دسترسی برای چک کردن نام کاربری و کلمه عبور	پیغام عدم اجازه دسترسی به سیستم برای مدت مشخص	✓
۳	در صورتی که مدت اخراج کاربر برای ورود به سیستم پایان یافته باشد	۵،۶،۸	Email Password نادرست	پیغام خطای ورود	خطای ورود به حساب کاربری	✓
۴	در صورتی که مدت اخراج کاربر برای ورود به سیستم پایان یافته باشد	۵،۶،۸	Email Password صحیح	ورود موفق به سیستم	پیغام خوش آمد گویی به کاربر	✓
۵	در صورتی که کاربر ایمیل و کلمه عبور نادرست در فرم ورود وارد کند و قبلا اخراج نشده باشد	۱۲،۱۳	Email Password نادرست	پیغام خطای ورود	خطای ورود به حساب کاربری	✓
۶	در صورتی که کاربر ایمیل و کلمه عبور صحیح در فرم ورود وارد کند و قبلا اخراج نشده باشد	۱۵	Email Password صحیح	ورود موفق به سیستم	پیغام خوش آمد گویی به کاربر	✓

## Register

TC#	توصیف مسیر	پیش نیاز	ورودی	خروجی مورد انتظار	خروجی واقعی	Pass Fail
۱	یکی از ورودی های فرم ثبت نام خالی باشد	۱،۲،۳	ورودی خالی برای یکی از پارامتر های ثبت نام	خطا در ثبت نام	پیغام خطای ثبت - ورودی ایمیل و یا کلمه عبور نمی تواند خالی باشد	✓
۲	عدم تطابق پسورد های وارد شده	۵	Pass != ConfirmPass	خطا در ثبت نام	پیغام خطا - پسورد های وارد شده تطابق ندارند	✓
۳	ایمیل کاربر در پایگاه داده از قبل موجود باشد	۷	Existed mail	خطا در ثبت نام	پیغام خطا - امکان ثبت نام برای این ایمیل وجود ندارد	✓
۴	اگر تمام اطلاعات ورودی صحیح باشد	۹	Email Pass ConfirmPass	ثبت نام موفق	پیغام ثبت نام موفقیت آمیز و تبریک به کاربر	✓



## Search

TC#	توصیف مسیر	پیش نیاز	ورودی	خروجی مورد انتظار	خروجی واقعی	Pass Fail
۱	اگر اطراف موقعیت مشخص شده توسط کاربر جای پارک وجود داشته باشد	۴	Place Time	لیست مکان های پارک	Available Park Place List	✓
۲	اگر اطراف موقعیت مشخص شده توسط کاربر جای پارک وجود نداشته باشد	۳	Place Time	عدم نمایش لیست	Empty Available Park Place List	✓

## Get Service Info

TC#	توصیف مسیر	پیش نیاز	ورودی	خروجی مورد انتظار	خروجی واقعی	Pass Fail
۱	مکان پارک مشخص شده توسط کاربر در دسترس باشد	۳	Place Start Time End Time	سیستم قیمت رزرو جای پارک را برگرداند	Place Price	✓
۲	مکان پارک مشخص شده توسط کاربر در دسترس نباشد	۴	Place Start Time End Time	عدم ارائه اطلاعات سرویس	false	✓

## Make Reservation

TC#	توصیف مسیر	پیش نیاز	ورودی	خروجی مورد انتظار	خروجی واقعی	Pass Fail
۱	اگر سرویس انتخاب شده در دسترس بود و پرداخت کاربر موفق باشد	۲،۴	Place Start Time End Time	رزرو موفق	نمایش پیغام رزرو موفق	✓
۲	اگر سرویس انتخاب شده در دسترس بود و پرداخت کاربر ناموفق باشد	۲،۴،۶	Place Start Time End Time	خطای پرداخت	نمایش خطای پرداخت به کاربر	✓
۳	اگر سرویس انتخاب شده در دسترس نباشد	۲،۸	Place Start Time End Time	خطای رزرو	نمایش خطای رزرو به کاربر	✓

## Cancel Reservation

TC#	توصیف مسیر	پیش نیاز	ورودی	خروجی مورد انتظار	خروجی واقعی	Pass Fail
۱	اگر زمان شروع سرویس رزرو آغاز نشده باشد	۲،۳	Email place	لغو رزرو جای پارک	نمایش پیغام لغو رزرو موفق به کاربر و بازگشت وجه	✓
۲	اگر از زمان شروع سرویس رزرو مدتی سپری شده باشد	۲،۴	Email place	خطا	نمایش خطای عدم امکان کنسل رزرو	✓

## Start Pay

TC#	توصیف مسیر	پیش نیاز	ورودی	خروجی مورد انتظار	خروجی واقعی	Pass Fail
۱	اگر کاربر پرداخت توسط کیف پول را انتخاب کند	۳	Price	پرداخت توسط کیف پول	فراخوانی تابع پرداخت توسط کیف پول	✓
۲	اگر کاربر پرداخت توسط کارت بانکی را انتخاب کند	۴	price	پرداخت توسط کارت بانکی	فراخوانی تابع پرداخت توسط کارت بانکی	✓

## Pay By Wallet

TC#	توصیف مسیر	پیش نیاز	ورودی	خروجی مورد انتظار	خروجی واقعی	Pass Fail
۱	اگر موجودی کاربر برای پرداخت کافی باشد	۲،۳	Price	Ref ID	نمایش پیغام پرداخت موفق و کسر مبلغ رزرو از حساب کاربری	✓
۲	اگر موجودی کاربر برای پرداخت کافی نباشد	۲،۴	price	خطای پرداخت	نمایش پیغام خطای پرداخت	✓

### تست جعبه سیاه

روشی در تست نرم افزار است که در آن فرض میشود اطلاعاتی در مورد جزئیات داخلی عملکرد نرم افزار وجود ندارد و تمرکز تستها بر روی خروجیهای مختلف در برابر ورودیهای متفاوت است. به این رویکرد تست functional نیز می گویند.

### قسمت بندی متعادل

تقسیم دامنه ورودی ها و نوشتن تست کیس برای هر دامنه

بررسی اتصال به اینترنت

تایید درخواست برای دسترسی به موقعیت مکانی کاربر

جستجو مکان مورد نظر

پرداخت هزینه

ورود کاربر

ثبت نام کاربر

رزرو کردن

ویرایش پروفایل

## تحلیل مقادیر مرزی

انتخاب مقادیری نزدیک به مرز

ردیف	ورودی	انتظار	نتیجه	وضعیت
۱	ورود به نرم افزار بدون اتصال به اینترنت	نمایش پیغام خطا مبنی بر عدم اتصال به اینترنت	به اینترنت متصل نمی باشید	✓
۲	رد کردن درخواست دسترسی به GPS	نمایش پیغامی مبنی بر عدم موفقیت در مکان یابی	دسترسی به موقعیت مکانی نیاز است	✓
۳	ورود نام مکان فرضی	عدم ارائه مکان در نتایج جستجو	جستجو نتیجه ای در برنداشت	✓
۴	مقدار هزینه پارک برابر با موجودی حساب کاربر	موفقیت در پرداخت هزینه	موفقیت در پرداخت هزینه	✓
۵	وارد کردن اشتباه نام کاربری یا رمز ورود	نمایش پیغام خطا	Email or password is incorrect	✓
۶	وارد نکردن نام کاربری یا رمز ورود	نمایش پیغام خطا	Fields Cannot be empty	✓
۷	وارد نکردن یکی از فیلدها در زمان ثبت نام	نمایش پیغام خطا	Fields Cannot be empty	✓
۸	وارد نکردن رمز یکسان در موقع ثبت نام	نمایش پیغام خطا	Passwords are not the same	✓
۹	وارد کردن ایمیل مشابه در ثبت نام	نمایش پیغام خطا	Email exist in database	✓
۱۰	رزرو در زمانی که مکان مشغول است.	نمایش پیغام خطا	Park Place is not available in your requested time range	✓
۱۱	کنسل کردن رزرو پس از شروع زمان آغاز رزرو	نمایش پیغام خطا	You can't cancel this reservation	✓
۱۲	خالی گذاشتن فیلدی در قسمت ویرایش	نمایش پیغام خطا	Error Occurred in updating profile, try again later	✓

## نگهداری

هدف از مرحله تعمیر و نگهداری این است که سیستم را در سطح قابل قبول نگه دارد.

تعمیر و نگهداری زمانی شروع می شود که سیستم منتشر شود. اگر در طول یک سال، هزینه نگهداری سیستم به طور قابل توجهی بیش از هزینه توسعه آن نباشد، غیر معمول نیست، بنابراین هدف اصلی کنترل هزینه های تعمیر و نگهداری است.

کلید کنترل هزینه های تعمیر و نگهداری، طراحی سیستم هایی است که تغییر می کنند.

به طور خلاصه رویه نگهداری نرم افزار بر مبنای زیر تأکید می کند: کنترل روی رویه های سیستم، کنترل روی تغییرات سیستم، جلوگیری از کاهش کیفیت و کارایی سیستم، ایجاد تغییرات برای افزایش کارایی سیستم.

در نرم افزار مدیریت پارکینگ، به منظور ایجاد بستری لازم و مناسب برای استفاده کاملتر از راهکارها و امکانات نرم افزاری برای کاربران می توان اقدامات ذیل را انجام نمود:

- قرار دادن قسمت پشتیبانی در برنامه که در بردارنده سوال های متدوال و راهنمای کامل و ارتباط با قسمت پشتیبانی آنلاین است.
- ارسال پیشنهادات و انتقادات کاربر در مورد خدمات و کارایی برنامه.
- گزارش خطاهای برنامه توسط کاربر برای رسیدگی سریع توسط تیم نگهداری و رفع خطا.

تیم نگهداری می تواند با استفاده از ابزارهایی مانند نرم افزار مدیریت پایگاه داده، سیستم های توسعه نرم افزار، ژنراتورهای برنامه، زبان نسل چهارم، تکنیک های ساخت یافته و تکنیک های شی گرا به طور قابل توجهی قابلیت اطمینان نرم افزار را افزایش دهند. از جمله وظایف مربوط به تیم نگهداری، حذف اشکالات نرم افزاری باقی مانده، بهبود یکپارچگی و قابلیت اطمینان برنامه ها، ساده سازی و تکرار روال اعتبار سنجی داده ها، اصلاح پردازش و گزارش نادرست و به حداقل رساندن خرابی است.

تغییر در UI برنامه (از جمله آیکن ها - تصاویر - انیمیشن ها - فونت و ...) جهت ایجاد تنوع و جلوگیری از تکراری شدن برنامه برای کاربر پس از مدتی استفاده از آن.

افزودن بخش های تکمیلی به نرم افزار از طریق بروزرسانی در گذر زمان در راستای توسعه قابلیت ها و همچنین رضایت کاربران نرم افزار.

قدردانی از کاربرانی که از محصول راضی هستند از طریق بازخورد مثبت